

GNSS-SDR

0.0.20

Generated by Doxygen 1.13.2



<b>1 GNSS-SDR</b>	<b>1</b>
1.1 Contents	1
1.2 Overview	2
1.3 Building GNSS-SDR	3
1.3.1 Debug and Release builds	5
1.3.2 Updating GNSS-SDR	5
1.4 Using GNSS-SDR	5
1.5 Control plane	6
1.5.1 Configuration	6
1.5.2 GNSS block factory	7
1.6 Signal Processing plane	7
1.6.1 Signal Source	8
1.6.2 Signal Conditioner	8
1.6.3 Channel	9
1.6.3.1 Acquisition	9
1.6.3.2 Tracking	10
1.6.3.3 Decoding of the navigation message	11
1.6.4 Observables	11
1.6.5 Computation of Position, Velocity and Time	11
1.7 About the software license	12
1.8 Publications and Credits	13
1.9 Ok, now what?	14
<b>2 Reference Documents</b>	<b>15</b>
2.1 Interface Control Documents	15
2.1.1 GPS	15
2.1.2 GLONASS	15
2.1.3 Galileo	16
2.1.4 BeiDou	16
2.1.5 Satellite Based Augmentation Systems (SBAS)	17
2.2 Other Standards	17
2.2.1 RINEX	17
2.2.2 NMEA	18
2.2.3 KML	18
2.2.4 C++ Standards	18
2.2.5 Positioning protocols in wireless communication networks	18
<b>3 Signal model</b>	<b>21</b>
3.1 GNSS signal model	21
3.1.1 Global Positioning System (GPS) signal in space	22
3.1.2 GLONASS signal in space	23
3.1.3 Galileo signal in space	24
3.1.4 Reference	27

<b>4 Todo List</b>	<b>29</b>
<b>5 Topic Index</b>	<b>31</b>
5.1 Topics	31
<b>6 Hierarchical Index</b>	<b>33</b>
6.1 Class Hierarchy	33
<b>7 Class Index</b>	<b>41</b>
7.1 Class List	41
<b>8 File Index</b>	<b>57</b>
8.1 File List	57
<b>9 Topic Documentation</b>	<b>75</b>
9.1 Acquisition	75
9.1.1 Detailed Description	75
9.1.2 acquisition_adapters	76
9.1.2.1 Detailed Description	77
9.1.2.2 Typedef Documentation	77
9.1.3 acquisition_gr_blocks	77
9.1.3.1 Detailed Description	79
9.1.3.2 Typedef Documentation	79
9.1.4 acquisition_libs	80
9.1.4.1 Detailed Description	81
9.2 Channel	81
9.2.1 Detailed Description	81
9.2.2 channel_adapters	81
9.2.2.1 Detailed Description	82
9.2.3 channel_libs	82
9.2.3.1 Detailed Description	82
9.2.3.2 Typedef Documentation	83
9.3 Signal Conditioner	83
9.3.1 Detailed Description	83
9.3.2 conditioner_adapters	83
9.3.2.1 Detailed Description	84
9.4 Data Type Adapters	84
9.4.1 Detailed Description	84
9.4.2 data_type_adapters	84
9.4.2.1 Detailed Description	85
9.4.3 data_type_gr_blocks	85
9.4.3.1 Detailed Description	86
9.4.3.2 Typedef Documentation	86
9.5 Input Filter	87



9.5.1 Detailed Description . . . . .	87
9.5.2 input_filter_adapters . . . . .	87
9.5.2.1 Detailed Description . . . . .	88
9.5.3 input_filter_gr_blocks . . . . .	88
9.5.3.1 Detailed Description . . . . .	89
9.5.3.2 Typedef Documentation . . . . .	89
9.5.3.3 Variable Documentation . . . . .	89
9.6 Algorithms Common Library . . . . .	90
9.6.1 Detailed Description . . . . .	90
9.6.2 algorithms_libs . . . . .	90
9.6.2.1 Detailed Description . . . . .	95
9.6.2.2 Typedef Documentation . . . . .	95
9.6.2.3 Function Documentation . . . . .	97
9.6.3 gnss_sdr_flags . . . . .	113
9.6.3.1 Detailed Description . . . . .	114
9.6.3.2 Function Documentation . . . . .	114
9.6.3.3 Variable Documentation . . . . .	118
9.7 PVT . . . . .	118
9.7.1 Detailed Description . . . . .	118
9.7.2 algorithms_libs_rtklib . . . . .	118
9.7.2.1 Detailed Description . . . . .	130
9.7.2.2 Macro Definition Documentation . . . . .	130
9.7.2.3 Typedef Documentation . . . . .	139
9.7.2.4 Function Documentation . . . . .	139
9.7.2.5 Variable Documentation . . . . .	139
9.7.3 pvt_adapters . . . . .	169
9.7.3.1 Detailed Description . . . . .	169
9.7.4 pvt_gr_blocks . . . . .	170
9.7.4.1 Detailed Description . . . . .	170
9.7.4.2 Typedef Documentation . . . . .	170
9.7.5 pvt_libs . . . . .	170
9.7.5.1 Detailed Description . . . . .	172
9.7.5.2 Typedef Documentation . . . . .	172
9.7.5.3 Function Documentation . . . . .	172
9.8 Observables . . . . .	173
9.8.1 Detailed Description . . . . .	173
9.8.2 obs_adapters . . . . .	173
9.8.2.1 Detailed Description . . . . .	174
9.8.3 obs_gr_blocks . . . . .	174
9.8.3.1 Detailed Description . . . . .	174
9.8.3.2 Typedef Documentation . . . . .	174
9.8.4 observables_libs . . . . .	175

9.8.4.1 Detailed Description	175
9.9 Resampler	175
9.9.1 Detailed Description	175
9.9.2 resampler_adapters	176
9.9.2.1 Detailed Description	176
9.9.3 resampler_gr_blocks	176
9.9.3.1 Detailed Description	177
9.9.3.2 Typedef Documentation	177
9.10 Signal Source	178
9.10.1 Detailed Description	178
9.10.2 signal_source_adapters	178
9.10.2.1 Detailed Description	180
9.10.3 signal_source_gr_blocks	180
9.10.3.1 Detailed Description	181
9.10.3.2 Typedef Documentation	181
9.10.4 signal_source_libs	183
9.10.4.1 Detailed Description	184
9.10.4.2 Macro Definition Documentation	184
9.10.4.3 Enumeration Type Documentation	185
9.10.4.4 Function Documentation	185
9.11 Telemetry Decoder	185
9.11.1 Detailed Description	186
9.11.2 telemetry_decoder_adapters	186
9.11.2.1 Detailed Description	187
9.11.3 telemetry_decoder_gr_blocks	187
9.11.3.1 Detailed Description	188
9.11.3.2 Typedef Documentation	188
9.11.4 telemetry_decoder_libswiftnav	189
9.11.4.1 Detailed Description	190
9.11.4.2 Macro Definition Documentation	190
9.11.5 telemetry_decoder_libs	192
9.11.5.1 Detailed Description	192
9.12 Tracking	193
9.12.1 Detailed Description	193
9.12.2 tracking_adapters	193
9.12.2.1 Detailed Description	194
9.12.3 tracking_gr_blocks	195
9.12.3.1 Detailed Description	196
9.12.3.2 Typedef Documentation	196
9.12.4 tracking_libs	198
9.12.4.1 Detailed Description	200
9.12.4.2 Macro Definition Documentation	200

9.12.4.3 Typedef Documentation . . . . .	200
9.12.4.4 Function Documentation . . . . .	201
9.13 Core GNSS Receiver . . . . .	205
9.13.1 Detailed Description . . . . .	205
9.13.2 GNSS block interfaces . . . . .	206
9.13.2.1 Detailed Description . . . . .	206
9.13.2.2 Typedef Documentation . . . . .	206
9.13.2.3 Function Documentation . . . . .	207
9.13.3 core_libs . . . . .	207
9.13.3.1 Detailed Description . . . . .	209
9.13.3.2 Macro Definition Documentation . . . . .	209
9.13.3.3 Typedef Documentation . . . . .	209
9.13.3.4 Function Documentation . . . . .	210
9.13.4 core_monitor . . . . .	210
9.13.4.1 Detailed Description . . . . .	211
9.13.4.2 Typedef Documentation . . . . .	211
9.13.5 core_receiver . . . . .	211
9.13.5.1 Detailed Description . . . . .	211
9.13.6 core_system_parameters . . . . .	212
9.13.6.1 Detailed Description . . . . .	246
9.13.6.2 Macro Definition Documentation . . . . .	246
9.13.6.3 Function Documentation . . . . .	246
9.13.6.4 Variable Documentation . . . . .	247
<b>10 Class Documentation</b>	<b>323</b>
10.1 Acq_Conf Class Reference . . . . .	324
10.1.1 Detailed Description . . . . .	325
10.1.2 Member Data Documentation . . . . .	325
10.1.2.1 bit_transition_flag . . . . .	325
10.1.2.2 blocking . . . . .	325
10.1.2.3 blocking_on_standby . . . . .	325
10.1.2.4 chips_per_second . . . . .	326
10.1.2.5 doppler_max . . . . .	326
10.1.2.6 doppler_min . . . . .	326
10.1.2.7 doppler_step . . . . .	326
10.1.2.8 doppler_step2 . . . . .	326
10.1.2.9 dump . . . . .	326
10.1.2.10 dump_channel . . . . .	326
10.1.2.11 dump_filename . . . . .	326
10.1.2.12 enable_monitor_output . . . . .	326
10.1.2.13 fs_in . . . . .	326
10.1.2.14 it_size . . . . .	326

10.1.2.15 item_type . . . . .	326
10.1.2.16 make_2_steps . . . . .	327
10.1.2.17 max_dwells . . . . .	327
10.1.2.18 ms_per_code . . . . .	327
10.1.2.19 num_doppler_bins_step2 . . . . .	327
10.1.2.20 pfa . . . . .	327
10.1.2.21 pfa2 . . . . .	327
10.1.2.22 resampled_fs . . . . .	327
10.1.2.23 resampler_latency_samples . . . . .	327
10.1.2.24 resampler_ratio . . . . .	327
10.1.2.25 sampled_ms . . . . .	327
10.1.2.26 samples_per_chip . . . . .	327
10.1.2.27 samples_per_code . . . . .	327
10.1.2.28 samples_per_ms . . . . .	328
10.1.2.29 use_automatic_resampler . . . . .	328
10.1.2.30 use_CFAR_algorithm_flag . . . . .	328
10.2 Acq_Conf_Fpga Class Reference . . . . .	329
10.2.1 Detailed Description . . . . .	330
10.2.2 Member Data Documentation . . . . .	330
10.2.2.1 all_fft_codes . . . . .	330
10.2.2.2 code_length . . . . .	330
10.2.2.3 code_length_chips . . . . .	330
10.2.2.4 code_rate_cps . . . . .	330
10.2.2.5 device_name . . . . .	330
10.2.2.6 doppler_max . . . . .	330
10.2.2.7 doppler_step . . . . .	331
10.2.2.8 doppler_step2 . . . . .	331
10.2.2.9 downsampling_factor . . . . .	331
10.2.2.10 downsampling_filter_delay . . . . .	331
10.2.2.11 downsampling_filter_num . . . . .	331
10.2.2.12 enable_zero_padding . . . . .	331
10.2.2.13 excludelimit . . . . .	331
10.2.2.14 fft_size . . . . .	331
10.2.2.15 fs_in . . . . .	331
10.2.2.16 make_2_steps . . . . .	331
10.2.2.17 max_num_acqs . . . . .	331
10.2.2.18 num_doppler_bins_step2 . . . . .	331
10.2.2.19 repeat_satellite . . . . .	332
10.2.2.20 resampled_fs . . . . .	332
10.2.2.21 total_block_exp . . . . .	332
10.3 AcquisitionInterface Class Reference . . . . .	332
10.3.1 Detailed Description . . . . .	334

10.3.2 Member Function Documentation	334
10.3.2.1 init()	334
10.3.2.2 mag()	334
10.3.2.3 reset()	335
10.3.2.4 set_channel()	335
10.3.2.5 set_channel_fsm()	335
10.3.2.6 set_doppler_center()	335
10.3.2.7 set_doppler_max()	335
10.3.2.8 set_doppler_step()	335
10.3.2.9 set_gnss_synchro()	336
10.3.2.10 set_local_code()	336
10.3.2.11 set_resampler_latency()	336
10.3.2.12 set_state()	336
10.3.2.13 set_threshold()	336
10.3.2.14 stop_acquisition()	337
10.4 ad936x_iio_custom Class Reference	337
10.4.1 Detailed Description	338
10.4.2 Member Data Documentation	338
10.4.2.1 n_channels	338
10.5 ad936x_iio_samples Class Reference	338
10.5.1 Detailed Description	339
10.5.2 Member Data Documentation	339
10.5.2.1 buffer	339
10.5.2.2 n_bytes	339
10.5.2.3 n_channels	339
10.5.2.4 n_interleaved_iq_samples	339
10.5.2.5 step_bytes	339
10.6 ad936x_iio_source Class Reference	339
10.6.1 Detailed Description	341
10.6.2 Member Function Documentation	341
10.6.2.1 start()	341
10.6.2.2 stop()	341
10.7 Ad936xCUSTOMSignalSource Class Reference	341
10.7.1 Detailed Description	344
10.7.2 Member Function Documentation	344
10.7.2.1 connect()	344
10.7.2.2 disconnect()	344
10.7.2.3 get_left_block()	344
10.7.2.4 get_right_block() [1/2]	344
10.7.2.5 get_right_block() [2/2]	345
10.7.2.6 item_size()	345
10.8 Adrv9361z7035SignalSourceFPGA Class Reference	346

10.8.1 Detailed Description	348
10.8.2 Member Function Documentation	348
10.8.2.1 connect()	348
10.8.2.2 disconnect()	348
10.8.2.3 get_left_block()	348
10.8.2.4 get_right_block()	348
10.8.2.5 item_size()	348
10.9 Agnss_Ref_Location Class Reference	349
10.9.1 Detailed Description	349
10.9.2 Constructor & Destructor Documentation	349
10.9.2.1 Agnss_Ref_Location()	349
10.9.3 Member Function Documentation	349
10.9.3.1 serialize()	349
10.9.4 Member Data Documentation	350
10.9.4.1 lat	350
10.9.4.2 lon	350
10.9.4.3 uncertainty	350
10.9.4.4 valid	350
10.10 Agnss_Ref_Time Class Reference	350
10.10.1 Detailed Description	351
10.10.2 Constructor & Destructor Documentation	351
10.10.2.1 Agnss_Ref_Time()	351
10.10.3 Member Function Documentation	351
10.10.3.1 serialize()	351
10.10.4 Member Data Documentation	351
10.10.4.1 microseconds	351
10.10.4.2 seconds	351
10.10.4.3 tow	351
10.10.4.4 valid	351
10.10.4.5 week	352
10.11 Allocator Struct Reference	352
10.11.1 Detailed Description	352
10.11.2 Constructor & Destructor Documentation	352
10.11.2.1 Allocator()	352
10.11.3 Member Function Documentation	352
10.11.3.1 operator()()	352
10.11.4 Member Data Documentation	353
10.11.4.1 buffer_	353
10.11.4.2 countwords_	353
10.12 alm_t Struct Reference	353
10.12.1 Detailed Description	354
10.12.2 Member Data Documentation	354

10.12.2.1 A	354
10.12.2.2 e	354
10.12.2.3 f0	354
10.12.2.4 f1	354
10.12.2.5 i0	354
10.12.2.6 M0	354
10.12.2.7 omg	354
10.12.2.8 OMG0	355
10.12.2.9 OMGd	355
10.12.2.10 sat	355
10.12.2.11 svconf	355
10.12.2.12 svh	355
10.12.2.13 toa	355
10.12.2.14 toas	355
10.12.2.15 week	355
10.13 ambc_t Struct Reference	356
10.13.1 Detailed Description	356
10.13.2 Member Data Documentation	356
10.13.2.1 epoch	356
10.13.2.2 fixcnt	356
10.13.2.3 flags	356
10.13.2.4 LC	357
10.13.2.5 LCv	357
10.13.2.6 n	357
10.14 An_Packet_Printer Class Reference	357
10.14.1 Detailed Description	357
10.14.2 Constructor & Destructor Documentation	358
10.14.2.1 An_Packet_Printer()	358
10.14.2.2 ~An_Packet_Printer()	358
10.14.3 Member Function Documentation	358
10.14.3.1 close_serial()	358
10.14.3.2 print_packet()	358
10.15 an_packet_t Struct Reference	358
10.15.1 Detailed Description	358
10.15.2 Member Data Documentation	359
10.15.2.1 data	359
10.15.2.2 header	359
10.16 ArraySignalConditioner Class Reference	359
10.16.1 Detailed Description	362
10.16.2 Constructor & Destructor Documentation	362
10.16.2.1 ArraySignalConditioner()	362
10.16.2.2 ~ArraySignalConditioner()	362

10.16.3 Member Function Documentation	362
10.16.3.1 connect()	362
10.16.3.2 data_type_adapter()	362
10.16.3.3 disconnect()	362
10.16.3.4 get_left_block()	363
10.16.3.5 get_right_block()	363
10.16.3.6 implementation()	363
10.16.3.7 input_filter()	363
10.16.3.8 item_size()	363
10.16.3.9 resampler()	363
10.16.3.10 role()	363
10.17 Bayesian_estimator Class Reference	363
10.17.1 Detailed Description	364
10.18 beamformer Class Reference	364
10.18.1 Detailed Description	365
10.19 BeamformerFilter Class Reference	366
10.19.1 Detailed Description	368
10.19.2 Member Function Documentation	368
10.19.2.1 connect()	368
10.19.2.2 disconnect()	368
10.19.2.3 get_left_block()	368
10.19.2.4 get_right_block()	368
10.19.2.5 implementation()	368
10.19.2.6 item_size()	368
10.19.2.7 role()	368
10.20 beidou_b1i_telemetry_decoder_gs Class Reference	369
10.20.1 Detailed Description	370
10.20.2 Constructor & Destructor Documentation	371
10.20.2.1 ~beidou_b1i_telemetry_decoder_gs()	371
10.20.3 Member Function Documentation	371
10.20.3.1 general_work()	371
10.20.3.2 set_channel()	371
10.20.3.3 set_satellite()	371
10.21 beidou_b3i_telemetry_decoder_gs Class Reference	371
10.21.1 Detailed Description	373
10.21.2 Constructor & Destructor Documentation	373
10.21.2.1 ~beidou_b3i_telemetry_decoder_gs()	373
10.21.3 Member Function Documentation	373
10.21.3.1 general_work()	373
10.21.3.2 set_channel()	373
10.21.3.3 set_satellite()	373
10.22 Beidou_Dnav_Almanac Class Reference	374



10.22.1 Detailed Description	376
10.22.2 Constructor & Destructor Documentation	376
10.22.2.1 Beidou_Dnav_Almanac()	376
10.22.3 Member Function Documentation	377
10.22.3.1 serialize()	377
10.22.4 Member Data Documentation	377
10.22.4.1 SV_health	377
10.23 Beidou_Dnav_Ephemeris Class Reference	377
10.23.1 Detailed Description	382
10.23.2 Constructor & Destructor Documentation	382
10.23.2.1 Beidou_Dnav_Ephemeris()	382
10.23.3 Member Function Documentation	382
10.23.3.1 serialize()	382
10.23.4 Member Data Documentation	382
10.23.4.1 alert_flag	382
10.23.4.2 antispoofing_flag	383
10.23.4.3 AODC	383
10.23.4.4 AODE	383
10.23.4.5 AODO	383
10.23.4.6 fit_interval_flag	383
10.23.4.7 integrity_status_flag	383
10.23.4.8 nav_type	383
10.23.4.9 satelliteBlock	384
10.23.4.10 sig_type	384
10.23.4.11 spare1	384
10.23.4.12 spare2	384
10.23.4.13 SV_accuracy	384
10.23.4.14 SV_health	384
10.23.4.15 TGD1	384
10.23.4.16 TGD2	384
10.24 Beidou_Dnav_Iono Class Reference	384
10.24.1 Detailed Description	387
10.24.2 Constructor & Destructor Documentation	387
10.24.2.1 Beidou_Dnav_Iono()	387
10.25 Beidou_Dnav_Navigation_Message Class Reference	387
10.25.1 Detailed Description	389
10.25.2 Constructor & Destructor Documentation	389
10.25.2.1 Beidou_Dnav_Navigation_Message()	389
10.25.3 Member Function Documentation	389
10.25.3.1 d1_subframe_decoder()	389
10.25.3.2 d2_subframe_decoder()	389
10.25.3.3 get_ephemeris()	389

10.25.3.4	<a href="#">get_flag_CRC_test()</a>	389
10.25.3.5	<a href="#">get_flag_new_SOW_available()</a>	389
10.25.3.6	<a href="#">get_iono()</a>	389
10.25.3.7	<a href="#">get_SOW()</a>	389
10.25.3.8	<a href="#">get_utc_model()</a>	390
10.25.3.9	<a href="#">have_new_almanac()</a>	390
10.25.3.10	<a href="#">have_new_ephemeris()</a>	390
10.25.3.11	<a href="#">have_new_iono()</a>	390
10.25.3.12	<a href="#">have_new_utc_model()</a>	390
10.25.3.13	<a href="#">set_flag_new_SOW_available()</a>	390
10.25.3.14	<a href="#">set_satellite_PRN()</a>	390
10.25.3.15	<a href="#">set_signal_type()</a>	390
10.25.3.16	<a href="#">utc_time()</a>	390
10.26	<a href="#">Beidou_Dnav_Utc_Model Class Reference</a>	390
10.26.1	<a href="#">Detailed Description</a>	392
10.26.2	<a href="#">Member Function Documentation</a>	392
10.26.2.1	<a href="#">serialize()</a>	392
10.26.3	<a href="#">Member Data Documentation</a>	392
10.26.3.1	<a href="#">A0_GAL</a>	392
10.26.3.2	<a href="#">A0_GLO</a>	392
10.26.3.3	<a href="#">A0_GPS</a>	392
10.26.3.4	<a href="#">A0_UTC</a>	392
10.26.3.5	<a href="#">A1_GAL</a>	392
10.26.3.6	<a href="#">A1_GLO</a>	393
10.26.3.7	<a href="#">A1_GPS</a>	393
10.26.3.8	<a href="#">A1_UTC</a>	393
10.26.3.9	<a href="#">DeltaT_LS</a>	393
10.26.3.10	<a href="#">DeltaT_LSF</a>	393
10.26.3.11	<a href="#">DN</a>	393
10.26.3.12	<a href="#">valid</a>	393
10.26.3.13	<a href="#">WN_LSF</a>	393
10.27	<a href="#">BeidouB1iDIPIITracking Class Reference</a>	393
10.27.1	<a href="#">Detailed Description</a>	396
10.27.2	<a href="#">Member Function Documentation</a>	396
10.27.2.1	<a href="#">connect()</a>	396
10.27.2.2	<a href="#">disconnect()</a>	396
10.27.2.3	<a href="#">get_left_block()</a>	396
10.27.2.4	<a href="#">get_right_block()</a>	396
10.27.2.5	<a href="#">implementation()</a>	396
10.27.2.6	<a href="#">item_size()</a>	396
10.27.2.7	<a href="#">role()</a>	397
10.27.2.8	<a href="#">set_channel()</a>	397

10.27.2.9 set_gnss_synchro()	397
10.27.2.10 start_tracking()	397
10.27.2.11 stop_tracking()	397
10.28 BeidouB1iPcpsAcquisition Class Reference	397
10.28.1 Detailed Description	400
10.28.2 Member Function Documentation	401
10.28.2.1 connect()	401
10.28.2.2 disconnect()	401
10.28.2.3 get_left_block()	401
10.28.2.4 get_right_block()	401
10.28.2.5 implementation()	401
10.28.2.6 init()	401
10.28.2.7 item_size()	401
10.28.2.8 mag()	401
10.28.2.9 reset()	401
10.28.2.10 role()	401
10.28.2.11 set_channel()	402
10.28.2.12 set_channel_fsm()	402
10.28.2.13 set_doppler_max()	402
10.28.2.14 set_doppler_step()	402
10.28.2.15 set_gnss_synchro()	402
10.28.2.16 set_local_code()	402
10.28.2.17 set_resampler_latency()	402
10.28.2.18 set_state()	402
10.28.2.19 set_threshold()	402
10.28.2.20 stop_acquisition()	403
10.29 BeidouB1iTelemetryDecoder Class Reference	403
10.29.1 Detailed Description	406
10.29.2 Member Function Documentation	406
10.29.2.1 connect()	406
10.29.2.2 disconnect()	406
10.29.2.3 get_left_block()	406
10.29.2.4 get_right_block()	406
10.29.2.5 implementation()	406
10.29.2.6 item_size()	406
10.29.2.7 reset()	407
10.29.2.8 role()	407
10.29.2.9 set_channel()	407
10.29.2.10 set_satellite()	407
10.30 BeidouB3iDIPIITracking Class Reference	407
10.30.1 Detailed Description	410
10.30.2 Member Function Documentation	410

10.30.2.1 connect()	410
10.30.2.2 disconnect()	410
10.30.2.3 get_left_block()	410
10.30.2.4 get_right_block()	410
10.30.2.5 implementation()	410
10.30.2.6 item_size()	410
10.30.2.7 role()	411
10.30.2.8 set_channel()	411
10.30.2.9 set_gnss_synchro()	411
10.30.2.10 start_tracking()	411
10.30.2.11 stop_tracking()	411
10.31 BeidouB3iPcpsAcquisition Class Reference	411
10.31.1 Detailed Description	414
10.31.2 Member Function Documentation	414
10.31.2.1 connect()	414
10.31.2.2 disconnect()	415
10.31.2.3 get_left_block()	415
10.31.2.4 get_right_block()	415
10.31.2.5 implementation()	415
10.31.2.6 init()	415
10.31.2.7 item_size()	415
10.31.2.8 mag()	415
10.31.2.9 reset()	415
10.31.2.10 role()	415
10.31.2.11 set_channel()	415
10.31.2.12 set_channel_fsm()	416
10.31.2.13 set_doppler_max()	416
10.31.2.14 set_doppler_step()	416
10.31.2.15 set_gnss_synchro()	416
10.31.2.16 set_local_code()	416
10.31.2.17 set_resampler_latency()	416
10.31.2.18 set_state()	416
10.31.2.19 set_threshold()	416
10.31.2.20 stop_acquisition()	417
10.32 BeidouB3iTelemetryDecoder Class Reference	417
10.32.1 Detailed Description	420
10.32.2 Member Function Documentation	420
10.32.2.1 connect()	420
10.32.2.2 disconnect()	420
10.32.2.3 get_left_block()	420
10.32.2.4 get_right_block()	420
10.32.2.5 implementation()	420

10.32.2.6 item_size()	420
10.32.2.7 reset()	421
10.32.2.8 role()	421
10.32.2.9 set_channel()	421
10.32.2.10 set_satellite()	421
10.33 byte_x2_to_complex_byte Class Reference	421
10.33.1 Detailed Description	422
10.34 ByteToShort Class Reference	422
10.34.1 Detailed Description	425
10.34.2 Member Function Documentation	425
10.34.2.1 connect()	425
10.34.2.2 disconnect()	425
10.34.2.3 get_left_block()	425
10.34.2.4 get_right_block()	425
10.34.2.5 implementation()	425
10.34.2.6 item_size()	425
10.34.2.7 role()	425
10.35 Channel Class Reference	426
10.35.1 Detailed Description	428
10.35.2 Constructor & Destructor Documentation	428
10.35.2.1 Channel()	428
10.35.2.2 ~Channel()	429
10.35.3 Member Function Documentation	429
10.35.3.1 acquisition()	429
10.35.3.2 assist_acquisition_doppler()	429
10.35.3.3 connect()	429
10.35.3.4 disconnect()	429
10.35.3.5 get_left_block()	429
10.35.3.6 get_left_block_acq()	429
10.35.3.7 get_left_block_trk()	429
10.35.3.8 get_right_block()	429
10.35.3.9 get_right_block_acq()	429
10.35.3.10 get_right_block_trk()	430
10.35.3.11 get_signal()	430
10.35.3.12 implementation()	430
10.35.3.13 item_size()	430
10.35.3.14 role()	430
10.35.3.15 set_signal()	430
10.35.3.16 start_acquisition()	430
10.35.3.17 stop_channel()	430
10.35.3.18 telemetry()	430
10.35.3.19 tracking()	430

10.36 Channel_Event Class Reference . . . . .	431
10.36.1 Detailed Description . . . . .	431
10.36.2 Member Data Documentation . . . . .	431
10.36.2.1 channel_id . . . . .	431
10.36.2.2 event_type . . . . .	431
10.37 channel_msg_receiver_cc Class Reference . . . . .	431
10.37.1 Detailed Description . . . . .	432
10.37.2 Constructor & Destructor Documentation . . . . .	433
10.37.2.1 ~channel_msg_receiver_cc() . . . . .	433
10.38 channel_status_msg_receiver Class Reference . . . . .	433
10.38.1 Detailed Description . . . . .	434
10.38.2 Constructor & Destructor Documentation . . . . .	434
10.38.2.1 ~channel_status_msg_receiver() . . . . .	434
10.38.3 Member Function Documentation . . . . .	434
10.38.3.1 get_current_status_map() . . . . .	434
10.38.3.2 get_current_status_pvt() . . . . .	435
10.39 ChannelFsm Class Reference . . . . .	435
10.39.1 Detailed Description . . . . .	436
10.40 ChannelInterface Class Reference . . . . .	436
10.40.1 Detailed Description . . . . .	439
10.40.2 Member Function Documentation . . . . .	439
10.40.2.1 get_left_block() . . . . .	439
10.40.2.2 get_left_block_acq() . . . . .	439
10.40.2.3 get_left_block_trk() . . . . .	439
10.40.2.4 get_right_block() . . . . .	439
10.40.2.5 get_right_block_acq() . . . . .	439
10.40.2.6 get_right_block_trk() . . . . .	439
10.40.2.7 set_signal() . . . . .	440
10.40.2.8 start_acquisition() . . . . .	440
10.40.2.9 stop_channel() . . . . .	440
10.41 cl_fft_plan Struct Reference . . . . .	441
10.41.1 Detailed Description . . . . .	442
10.41.2 Member Data Documentation . . . . .	442
10.41.2.1 context . . . . .	442
10.41.2.2 dim . . . . .	442
10.41.2.3 format . . . . .	442
10.41.2.4 kernel_info . . . . .	442
10.41.2.5 kernel_string . . . . .	442
10.41.2.6 last_batch_size . . . . .	442
10.41.2.7 max_localmem_fft_size . . . . .	442
10.41.2.8 max_radix . . . . .	442
10.41.2.9 max_work_item_per_workgroup . . . . .	442

10.41.2.10 min_mem_coalesce_width . . . . .	443
10.41.2.11 n . . . . .	443
10.41.2.12 num_kernels . . . . .	443
10.41.2.13 num_local_mem_banks . . . . .	443
10.41.2.14 program . . . . .	443
10.41.2.15 temp_buffer_needed . . . . .	443
10.41.2.16 tempmemobj . . . . .	443
10.41.2.17 tempmemobj_imag . . . . .	443
10.41.2.18 tempmemobj_real . . . . .	443
10.41.2.19 twist_kernel . . . . .	443
10.42 cIFFT_Complex Struct Reference . . . . .	444
10.42.1 Detailed Description . . . . .	444
10.42.2 Member Data Documentation . . . . .	444
10.42.2.1 imag . . . . .	444
10.42.2.2 real . . . . .	444
10.43 cIFFT_Dim3 Struct Reference . . . . .	445
10.43.1 Detailed Description . . . . .	445
10.43.2 Member Data Documentation . . . . .	445
10.43.2.1 x . . . . .	445
10.43.2.2 y . . . . .	445
10.43.2.3 z . . . . .	445
10.44 cIFFT_SplitComplex Struct Reference . . . . .	446
10.44.1 Detailed Description . . . . .	446
10.44.2 Member Data Documentation . . . . .	446
10.44.2.1 imag . . . . .	446
10.44.2.2 real . . . . .	446
10.45 cnav_msg_decoder_t Struct Reference . . . . .	446
10.45.1 Detailed Description . . . . .	447
10.45.2 Member Data Documentation . . . . .	448
10.45.2.1 part1 . . . . .	448
10.45.2.2 part2 . . . . .	448
10.46 cnav_msg_t Struct Reference . . . . .	448
10.46.1 Detailed Description . . . . .	448
10.46.2 Member Data Documentation . . . . .	449
10.46.2.1 alert . . . . .	449
10.46.2.2 msg_id . . . . .	449
10.46.2.3 prn . . . . .	449
10.46.2.4 raw_msg . . . . .	449
10.46.2.5 tow . . . . .	449
10.47 cnav_v27_part_t Struct Reference . . . . .	449
10.47.1 Detailed Description . . . . .	451
10.47.2 Member Data Documentation . . . . .	451

10.47.2.1	<a href="#">crc_ok</a>	451
10.47.2.2	<a href="#">dec</a>	451
10.47.2.3	<a href="#">decisions</a>	451
10.47.2.4	<a href="#">decoded</a>	451
10.47.2.5	<a href="#">init</a>	451
10.47.2.6	<a href="#">invert</a>	451
10.47.2.7	<a href="#">message_lock</a>	451
10.47.2.8	<a href="#">n_crc_fail</a>	452
10.47.2.9	<a href="#">n_decoded</a>	452
10.47.2.10	<a href="#">n_symbols</a>	452
10.47.2.11	<a href="#">preamble_seen</a>	452
10.47.2.12	<a href="#">symbols</a>	452
10.48	<a href="#">Command_Event Class Reference</a>	452
10.48.1	<a href="#">Detailed Description</a>	453
10.48.2	<a href="#">Member Data Documentation</a>	453
10.48.2.1	<a href="#">command_id</a>	453
10.48.2.2	<a href="#">event_type</a>	453
10.49	<a href="#">complex_byte_to_float_x2 Class Reference</a>	453
10.49.1	<a href="#">Detailed Description</a>	454
10.50	<a href="#">complex_float_to_complex_byte Class Reference</a>	454
10.50.1	<a href="#">Detailed Description</a>	455
10.51	<a href="#">Concurrent_Map&lt; Data &gt; Class Template Reference</a>	456
10.51.1	<a href="#">Detailed Description</a>	456
10.51.2	<a href="#">Member Function Documentation</a>	456
10.51.2.1	<a href="#">get_map_copy()</a> [1/2]	456
10.51.2.2	<a href="#">get_map_copy()</a> [2/2]	456
10.51.2.3	<a href="#">read()</a>	456
10.51.2.4	<a href="#">size()</a>	457
10.51.2.5	<a href="#">write()</a>	457
10.52	<a href="#">Concurrent_Queue&lt; Data &gt; Class Template Reference</a>	457
10.52.1	<a href="#">Detailed Description</a>	458
10.52.2	<a href="#">Member Function Documentation</a>	458
10.52.2.1	<a href="#">clear()</a>	458
10.52.2.2	<a href="#">empty()</a>	458
10.52.2.3	<a href="#">push()</a> [1/2]	459
10.52.2.4	<a href="#">push()</a> [2/2]	459
10.52.2.5	<a href="#">size()</a>	459
10.52.2.6	<a href="#">timed_wait_and_pop()</a>	459
10.52.2.7	<a href="#">try_pop()</a>	459
10.52.2.8	<a href="#">wait_and_pop()</a>	459
10.53	<a href="#">ConfigurationInterface Class Reference</a>	459
10.53.1	<a href="#">Detailed Description</a>	461



10.54 conjugate_cc Class Reference . . . . .	461
10.54.1 Detailed Description . . . . .	462
10.55 conjugate_ic Class Reference . . . . .	463
10.55.1 Detailed Description . . . . .	463
10.56 conjugate_sc Class Reference . . . . .	464
10.56.1 Detailed Description . . . . .	465
10.57 ControlThread Class Reference . . . . .	465
10.57.1 Detailed Description . . . . .	466
10.57.2 Constructor & Destructor Documentation . . . . .	466
10.57.2.1 ControlThread() [1/2] . . . . .	466
10.57.2.2 ControlThread() [2/2] . . . . .	466
10.57.2.3 ~ControlThread() . . . . .	466
10.57.3 Member Function Documentation . . . . .	466
10.57.3.1 applied_actions() . . . . .	466
10.57.3.2 flowgraph() . . . . .	466
10.57.3.3 processed_control_messages() . . . . .	466
10.57.3.4 run() . . . . .	467
10.57.3.5 set_control_queue() . . . . .	467
10.57.4 Member Data Documentation . . . . .	467
10.57.4.1 me . . . . .	467
10.58 Cpu_Multicorrelator Class Reference . . . . .	467
10.58.1 Detailed Description . . . . .	468
10.59 Cpu_Multicorrelator_16sc Class Reference . . . . .	468
10.59.1 Detailed Description . . . . .	469
10.60 Cpu_Multicorrelator_Real_Codes Class Reference . . . . .	469
10.60.1 Detailed Description . . . . .	470
10.61 cshort_to_float_x2 Class Reference . . . . .	470
10.61.1 Detailed Description . . . . .	471
10.62 cshort_to_gr_complex Class Reference . . . . .	472
10.62.1 Detailed Description . . . . .	472
10.63 CshortToGrComplex Class Reference . . . . .	473
10.63.1 Detailed Description . . . . .	475
10.63.2 Member Function Documentation . . . . .	475
10.63.2.1 connect() . . . . .	475
10.63.2.2 disconnect() . . . . .	475
10.63.2.3 get_left_block() . . . . .	475
10.63.2.4 get_right_block() . . . . .	475
10.63.2.5 implementation() . . . . .	475
10.63.2.6 item_size() . . . . .	475
10.63.2.7 role() . . . . .	475
10.64 CubatureFilter Class Reference . . . . .	476
10.64.1 Detailed Description . . . . .	476

10.65 <a href="#">cuda_multicorrelator Class Reference</a>	476
10.65.1 Detailed Description	477
10.66 <a href="#">CustomUDPSignalSource Class Reference</a>	477
10.66.1 Detailed Description	480
10.66.2 Member Function Documentation	480
10.66.2.1 <a href="#">connect()</a>	480
10.66.2.2 <a href="#">disconnect()</a>	480
10.66.2.3 <a href="#">get_left_block()</a>	480
10.66.2.4 <a href="#">get_right_block()</a> [1/2]	480
10.66.2.5 <a href="#">get_right_block()</a> [2/2]	481
10.66.2.6 <a href="#">item_size()</a>	481
10.67 <a href="#">Deleter Struct Reference</a>	481
10.67.1 Detailed Description	481
10.67.2 Constructor & Destructor Documentation	481
10.67.2.1 <a href="#">Deleter()</a>	481
10.67.3 Member Function Documentation	482
10.67.3.1 <a href="#">operator()()</a>	482
10.67.4 Member Data Documentation	482
10.67.4.1 <a href="#">buffer_</a>	482
10.68 <a href="#">dgps_t Struct Reference</a>	482
10.68.1 Detailed Description	482
10.68.2 Member Data Documentation	483
10.68.2.1 <a href="#">iod</a>	483
10.68.2.2 <a href="#">prc</a>	483
10.68.2.3 <a href="#">rrc</a>	483
10.68.2.4 <a href="#">t0</a>	483
10.68.2.5 <a href="#">udre</a>	483
10.69 <a href="#">direct_resampler_conditioner_cb Class Reference</a>	483
10.69.1 Detailed Description	484
10.69.2 Member Function Documentation	484
10.69.2.1 <a href="#">sample_freq_in()</a>	484
10.69.2.2 <a href="#">sample_freq_out()</a>	484
10.70 <a href="#">direct_resampler_conditioner_cc Class Reference</a>	485
10.70.1 Detailed Description	486
10.70.2 Member Function Documentation	486
10.70.2.1 <a href="#">sample_freq_in()</a>	486
10.70.2.2 <a href="#">sample_freq_out()</a>	486
10.71 <a href="#">direct_resampler_conditioner_cs Class Reference</a>	486
10.71.1 Detailed Description	488
10.71.2 Member Function Documentation	488
10.71.2.1 <a href="#">sample_freq_in()</a>	488
10.71.2.2 <a href="#">sample_freq_out()</a>	488

10.72 DirectResamplerConditioner Class Reference	488
10.72.1 Detailed Description	491
10.72.2 Member Function Documentation	491
10.72.2.1 connect()	491
10.72.2.2 disconnect()	491
10.72.2.3 get_left_block()	491
10.72.2.4 get_right_block()	491
10.72.2.5 implementation()	491
10.72.2.6 item_size()	491
10.72.2.7 role()	491
10.73 Dll_Pll_Conf Class Reference	492
10.73.1 Detailed Description	493
10.73.2 Member Data Documentation	493
10.73.2.1 bit_synchronization_time_limit_s	493
10.73.2.2 carrier_aiding	493
10.73.2.3 carrier_lock_test_smoother_alpha	494
10.73.2.4 carrier_lock_test_smoother_samples	494
10.73.2.5 carrier_lock_th	494
10.73.2.6 cn0_min	494
10.73.2.7 cn0_samples	494
10.73.2.8 cn0_smoother_alpha	494
10.73.2.9 cn0_smoother_samples	494
10.73.2.10 dll_bw_hz	494
10.73.2.11 dll_bw_narrow_hz	494
10.73.2.12 dll_filter_order	494
10.73.2.13 dll_pull_in_bw_hz	494
10.73.2.14 dump	494
10.73.2.15 dump_filename	495
10.73.2.16 dump_mat	495
10.73.2.17 early_late_space_chips	495
10.73.2.18 early_late_space_narrow_chips	495
10.73.2.19 enable_doppler_correction	495
10.73.2.20 enable_fll_pull_in	495
10.73.2.21 enable_fll_steady_state	495
10.73.2.22 extend_correlation_symbols	495
10.73.2.23 fll_bw_hz	495
10.73.2.24 fll_filter_order	495
10.73.2.25 fs_in	495
10.73.2.26 high_dyn	495
10.73.2.27 item_type	496
10.73.2.28 max_carrier_lock_fail	496
10.73.2.29 max_code_lock_fail	496

10.73.2.30 pll_bw_hz . . . . .	496
10.73.2.31 pll_bw_narrow_hz . . . . .	496
10.73.2.32 pll_filter_order . . . . .	496
10.73.2.33 pll_pull_in_bw_hz . . . . .	496
10.73.2.34 pull_in_time_s . . . . .	496
10.73.2.35 signal . . . . .	496
10.73.2.36 slope . . . . .	496
10.73.2.37 smoother_length . . . . .	496
10.73.2.38 spc . . . . .	496
10.73.2.39 system . . . . .	497
10.73.2.40 track_pilot . . . . .	497
10.73.2.41 vector_length . . . . .	497
10.73.2.42 very_early_late_space_chips . . . . .	497
10.73.2.43 very_early_late_space_narrow_chips . . . . .	497
10.73.2.44 y_intercept . . . . .	497
10.74 Dll_Pll_Conf_Fpga Class Reference . . . . .	498
10.74.1 Detailed Description . . . . .	499
10.74.2 Member Data Documentation . . . . .	499
10.74.2.1 bit_synchronization_time_limit_s . . . . .	499
10.74.2.2 ca_codes . . . . .	500
10.74.2.3 carrier_aiding . . . . .	500
10.74.2.4 carrier_lock_test_smoother_alpha . . . . .	500
10.74.2.5 carrier_lock_test_smoother_samples . . . . .	500
10.74.2.6 carrier_lock_th . . . . .	500
10.74.2.7 cn0_min . . . . .	500
10.74.2.8 cn0_samples . . . . .	500
10.74.2.9 cn0_smoother_alpha . . . . .	500
10.74.2.10 cn0_smoother_samples . . . . .	500
10.74.2.11 code_length_chips . . . . .	500
10.74.2.12 code_samples_per_chip . . . . .	500
10.74.2.13 data_codes . . . . .	500
10.74.2.14 device_name . . . . .	501
10.74.2.15 dll_bw_hz . . . . .	501
10.74.2.16 dll_bw_narrow_hz . . . . .	501
10.74.2.17 dll_filter_order . . . . .	501
10.74.2.18 dll_pull_in_bw_hz . . . . .	501
10.74.2.19 dump . . . . .	501
10.74.2.20 dump_filename . . . . .	501
10.74.2.21 dump_mat . . . . .	501
10.74.2.22 early_late_space_chips . . . . .	501
10.74.2.23 early_late_space_narrow_chips . . . . .	501
10.74.2.24 enable_doppler_correction . . . . .	501

10.74.2.25 enable_fill_pull_in . . . . .	501
10.74.2.26 enable_fill_steady_state . . . . .	502
10.74.2.27 extend_correlation_symbols . . . . .	502
10.74.2.28 extend_fpga_integration_periods . . . . .	502
10.74.2.29 extended_correlation_in_fpga . . . . .	502
10.74.2.30 fill_bw_hz . . . . .	502
10.74.2.31 fill_filter_order . . . . .	502
10.74.2.32 fpga_integration_period . . . . .	502
10.74.2.33 fs_in . . . . .	502
10.74.2.34 high_dyn . . . . .	502
10.74.2.35 max_carrier_lock_fail . . . . .	502
10.74.2.36 max_code_lock_fail . . . . .	502
10.74.2.37 pll_bw_hz . . . . .	502
10.74.2.38 pll_bw_narrow_hz . . . . .	503
10.74.2.39 pll_filter_order . . . . .	503
10.74.2.40 pll_pull_in_bw_hz . . . . .	503
10.74.2.41 pull_in_time_s . . . . .	503
10.74.2.42 signal . . . . .	503
10.74.2.43 slope . . . . .	503
10.74.2.44 smoother_length . . . . .	503
10.74.2.45 spc . . . . .	503
10.74.2.46 system . . . . .	503
10.74.2.47 track_pilot . . . . .	503
10.74.2.48 vector_length . . . . .	503
10.74.2.49 very_early_late_space_chips . . . . .	503
10.74.2.50 very_early_late_space_narrow_chips . . . . .	504
10.74.2.51 y_intercept . . . . .	504
10.75 dll_pll_veml_tracking Class Reference . . . . .	504
10.75.1 Detailed Description . . . . .	505
10.76 dll_pll_veml_tracking_fpga Class Reference . . . . .	505
10.76.1 Detailed Description . . . . .	508
10.76.2 Constructor & Destructor Documentation . . . . .	508
10.76.2.1 ~dll_pll_veml_tracking_fpga() . . . . .	508
10.76.3 Member Function Documentation . . . . .	508
10.76.3.1 general_work() . . . . .	508
10.76.3.2 reset() . . . . .	508
10.76.3.3 set_channel() . . . . .	508
10.76.3.4 set_gnss_synchro() . . . . .	508
10.76.3.5 start_tracking() . . . . .	508
10.76.3.6 stop_tracking() . . . . .	508
10.77 DMASignalSourceFPGA Class Reference . . . . .	509
10.77.1 Detailed Description . . . . .	511

10.77.2 Member Function Documentation	511
10.77.2.1 connect()	511
10.77.2.2 disconnect()	511
10.77.2.3 get_left_block()	511
10.77.2.4 get_right_block()	511
10.77.2.5 item_size()	511
10.77.2.6 start()	512
10.78 DSM_dsm_header Class Reference	512
10.78.1 Detailed Description	512
10.78.2 Member Data Documentation	512
10.78.2.1 dsm_block_id	512
10.78.2.2 dsm_id	512
10.79 DSM_KROOT_message Class Reference	513
10.79.1 Detailed Description	514
10.79.2 Member Data Documentation	514
10.79.2.1 alpha	514
10.79.2.2 cidkr	514
10.79.2.3 ds	514
10.79.2.4 hf	514
10.79.2.5 kroot	514
10.79.2.6 ks	514
10.79.2.7 maclt	514
10.79.2.8 mf	514
10.79.2.9 nb_dk	515
10.79.2.10 p_dk	515
10.79.2.11 pkid	515
10.79.2.12 reserved	515
10.79.2.13 reserved1	515
10.79.2.14 towh_k	515
10.79.2.15 ts	515
10.79.2.16 verified	515
10.79.2.17 wn_k	515
10.80 DSM_nma_header Class Reference	516
10.80.1 Detailed Description	516
10.80.2 Member Data Documentation	516
10.80.2.1 cid	516
10.80.2.2 cpks	516
10.80.2.3 nmas	516
10.80.2.4 reserved	516
10.81 DSM_PKR_message Class Reference	517
10.81.1 Detailed Description	518
10.81.2 Member Data Documentation	518

10.81.2.1 itn . . . . .	518
10.81.2.2 mid . . . . .	518
10.81.2.3 nb_dp . . . . .	518
10.81.2.4 npk . . . . .	518
10.81.2.5 npkt . . . . .	518
10.81.2.6 npktid . . . . .	518
10.81.2.7 p_dp . . . . .	518
10.82 eph_t Struct Reference . . . . .	519
10.82.1 Detailed Description . . . . .	520
10.82.2 Member Data Documentation . . . . .	520
10.82.2.1 A . . . . .	520
10.82.2.2 Adot . . . . .	520
10.82.2.3 apply_has_corrections . . . . .	520
10.82.2.4 cic . . . . .	520
10.82.2.5 cis . . . . .	520
10.82.2.6 code . . . . .	521
10.82.2.7 crc . . . . .	521
10.82.2.8 crs . . . . .	521
10.82.2.9 cuc . . . . .	521
10.82.2.10 cus . . . . .	521
10.82.2.11 deln . . . . .	521
10.82.2.12 e . . . . .	521
10.82.2.13 f0 . . . . .	521
10.82.2.14 f1 . . . . .	521
10.82.2.15 f2 . . . . .	521
10.82.2.16 fit . . . . .	521
10.82.2.17 flag . . . . .	521
10.82.2.18 has_clock_correction_m . . . . .	522
10.82.2.19 has_orbit_cross_track_correction_m . . . . .	522
10.82.2.20 has_orbit_in_track_correction_m . . . . .	522
10.82.2.21 has_orbit_radial_correction_m . . . . .	522
10.82.2.22 i0 . . . . .	522
10.82.2.23 idot . . . . .	522
10.82.2.24 iodic . . . . .	522
10.82.2.25 iode . . . . .	522
10.82.2.26 isc . . . . .	522
10.82.2.27 M0 . . . . .	522
10.82.2.28 ndot . . . . .	522
10.82.2.29 omg . . . . .	522
10.82.2.30 OMG0 . . . . .	523
10.82.2.31 OMGd . . . . .	523
10.82.2.32 sat . . . . .	523

10.82.2.33 sva . . . . .	523
10.82.2.34 svh . . . . .	523
10.82.2.35 tgd . . . . .	523
10.82.2.36 toc . . . . .	523
10.82.2.37 toe . . . . .	523
10.82.2.38 toes . . . . .	523
10.82.2.39 ttr . . . . .	523
10.82.2.40 week . . . . .	523
10.83 erp_t Struct Reference . . . . .	524
10.83.1 Detailed Description . . . . .	524
10.83.2 Member Data Documentation . . . . .	524
10.83.2.1 data . . . . .	524
10.83.2.2 n . . . . .	524
10.83.2.3 nmax . . . . .	524
10.84 erpd_t Struct Reference . . . . .	525
10.84.1 Detailed Description . . . . .	525
10.84.2 Member Data Documentation . . . . .	525
10.84.2.1 lod . . . . .	525
10.84.2.2 mjd . . . . .	525
10.84.2.3 ut1_utc . . . . .	525
10.84.2.4 xp . . . . .	526
10.84.2.5 xpr . . . . .	526
10.84.2.6 yp . . . . .	526
10.84.2.7 ypr . . . . .	526
10.85 Exponential_Smoother Class Reference . . . . .	526
10.85.1 Detailed Description . . . . .	527
10.85.2 Constructor & Destructor Documentation . . . . .	527
10.85.2.1 Exponential_Smoother() [1/2] . . . . .	527
10.85.2.2 ~Exponential_Smoother() . . . . .	527
10.85.2.3 Exponential_Smoother() [2/2] . . . . .	527
10.85.3 Member Function Documentation . . . . .	528
10.85.3.1 operator=() . . . . .	528
10.85.3.2 set_alpha() . . . . .	528
10.85.3.3 set_samples_for_initialization() . . . . .	528
10.86 exterr_t Struct Reference . . . . .	528
10.86.1 Detailed Description . . . . .	529
10.86.2 Member Data Documentation . . . . .	529
10.86.2.1 cerr . . . . .	529
10.86.2.2 ena . . . . .	529
10.86.2.3 gloicb . . . . .	529
10.86.2.4 gpsglob . . . . .	529
10.86.2.5 perr . . . . .	529



10.87 fcbd_t Struct Reference . . . . .	530
10.87.1 Detailed Description . . . . .	530
10.87.2 Member Data Documentation . . . . .	530
10.87.2.1 bias . . . . .	530
10.87.2.2 std . . . . .	530
10.87.2.3 te . . . . .	530
10.87.2.4 ts . . . . .	531
10.88 FifoReader Class Reference . . . . .	531
10.88.1 Detailed Description . . . . .	532
10.88.2 Member Typedef Documentation . . . . .	532
10.88.2.1 sptr . . . . .	532
10.88.3 Member Function Documentation . . . . .	532
10.88.3.1 start() . . . . .	532
10.89 FifoSignalSource Class Reference . . . . .	533
10.89.1 Detailed Description . . . . .	535
10.89.2 Member Function Documentation . . . . .	535
10.89.2.1 connect() . . . . .	535
10.89.2.2 disconnect() . . . . .	535
10.89.2.3 get_left_block() . . . . .	536
10.89.2.4 get_right_block() . . . . .	536
10.89.2.5 item_size() . . . . .	536
10.90 file_t Struct Reference . . . . .	537
10.90.1 Detailed Description . . . . .	538
10.90.2 Member Data Documentation . . . . .	538
10.90.2.1 fp . . . . .	538
10.90.2.2 fp_tag . . . . .	538
10.90.2.3 fp_tag_tmp . . . . .	538
10.90.2.4 fp_tmp . . . . .	538
10.90.2.5 fpos . . . . .	538
10.90.2.6 lock . . . . .	538
10.90.2.7 mode . . . . .	538
10.90.2.8 offset . . . . .	538
10.90.2.9 openpath . . . . .	539
10.90.2.10 path . . . . .	539
10.90.2.11 repmode . . . . .	539
10.90.2.12 speed . . . . .	539
10.90.2.13 start . . . . .	539
10.90.2.14 swapintv . . . . .	539
10.90.2.15 tick . . . . .	539
10.90.2.16 tick_f . . . . .	539
10.90.2.17 time . . . . .	539
10.90.2.18 timetag . . . . .	539

10.90.2.19 wtime . . . . .	539
10.91 FileConfiguration Class Reference . . . . .	540
10.91.1 Detailed Description . . . . .	542
10.91.2 Member Function Documentation . . . . .	542
10.91.2.1 property() [1/10] . . . . .	542
10.91.2.2 property() [2/10] . . . . .	542
10.91.2.3 property() [3/10] . . . . .	542
10.91.2.4 property() [4/10] . . . . .	542
10.91.2.5 property() [5/10] . . . . .	542
10.91.2.6 property() [6/10] . . . . .	542
10.91.2.7 property() [7/10] . . . . .	543
10.91.2.8 property() [8/10] . . . . .	543
10.91.2.9 property() [9/10] . . . . .	543
10.91.2.10 property() [10/10] . . . . .	543
10.91.2.11 set_property() . . . . .	543
10.92 FileSignalSource Class Reference . . . . .	543
10.92.1 Detailed Description . . . . .	547
10.93 FileSourceBase Class Reference . . . . .	547
10.93.1 Detailed Description . . . . .	551
10.93.2 Constructor & Destructor Documentation . . . . .	551
10.93.2.1 FileSourceBase() . . . . .	551
10.93.3 Member Function Documentation . . . . .	551
10.93.3.1 computeSamplesInFile() . . . . .	551
10.93.3.2 connect() . . . . .	552
10.93.3.3 disconnect() . . . . .	552
10.93.3.4 filename() . . . . .	552
10.93.3.5 get_left_block() . . . . .	552
10.93.3.6 get_right_block() . . . . .	552
10.93.3.7 init() . . . . .	552
10.93.3.8 item_size() . . . . .	552
10.93.3.9 item_type() . . . . .	552
10.93.3.10 itemTypeToSize() . . . . .	552
10.93.3.11 packetsPerSample() . . . . .	552
10.93.3.12 repeat() . . . . .	552
10.93.3.13 samples() . . . . .	553
10.93.3.14 samplesToSkip() . . . . .	553
10.93.3.15 sampling_frequency() . . . . .	553
10.93.3.16 source() . . . . .	553
10.93.3.17 source_item_size() . . . . .	553
10.94 FileTimestampSignalSource Class Reference . . . . .	553
10.94.1 Detailed Description . . . . .	557
10.94.2 Member Function Documentation . . . . .	557

10.94.2.1 create_file_source_hook()	557
10.94.2.2 pre_connect_hook()	557
10.94.2.3 pre_disconnect_hook()	557
10.94.2.4 source()	557
10.95 FirFilter Class Reference	558
10.95.1 Detailed Description	560
10.95.2 Constructor & Destructor Documentation	560
10.95.2.1 FirFilter()	560
10.95.2.2 ~FirFilter()	560
10.95.3 Member Function Documentation	560
10.95.3.1 connect()	560
10.95.3.2 disconnect()	560
10.95.3.3 get_left_block()	560
10.95.3.4 get_right_block()	560
10.95.3.5 implementation()	561
10.95.3.6 item_size()	561
10.95.3.7 role()	561
10.96 FlexibandSignalSource Class Reference	561
10.96.1 Detailed Description	564
10.96.2 Member Function Documentation	564
10.96.2.1 connect()	564
10.96.2.2 disconnect()	564
10.96.2.3 get_left_block()	564
10.96.2.4 get_right_block() [1/2]	564
10.96.2.5 get_right_block() [2/2]	565
10.96.2.6 item_size()	565
10.97 Fmcomms2SignalSource Class Reference	566
10.97.1 Detailed Description	568
10.97.2 Member Function Documentation	568
10.97.2.1 connect()	568
10.97.2.2 disconnect()	568
10.97.2.3 get_left_block()	568
10.97.2.4 get_right_block()	568
10.97.2.5 item_size()	568
10.98 Fmcomms5SignalSourceFPGA Class Reference	569
10.98.1 Detailed Description	571
10.98.2 Member Function Documentation	571
10.98.2.1 connect()	571
10.98.2.2 disconnect()	571
10.98.2.3 get_left_block()	571
10.98.2.4 get_right_block()	571
10.98.2.5 item_size()	571

10.99 FourBitCpxFileSignalSource Class Reference	572
10.99.1 Detailed Description	575
10.99.2 Member Function Documentation	575
10.99.2.1 create_file_source_hook()	575
10.99.2.2 itemTypeToSize()	575
10.99.2.3 packetsPerSample()	575
10.99.2.4 pre_connect_hook()	575
10.99.2.5 pre_disconnect_hook()	576
10.99.2.6 source()	576
10.100 Fpga_Acquisition Class Reference	576
10.100.1 Detailed Description	577
10.100.2 Constructor & Destructor Documentation	577
10.100.2.1 Fpga_Acquisition()	577
10.100.2.2 ~Fpga_Acquisition()	577
10.100.3 Member Function Documentation	577
10.100.3.1 close_device()	577
10.100.3.2 configure_acquisition()	578
10.100.3.3 init()	578
10.100.3.4 open_device()	578
10.100.3.5 read_acquisition_results()	578
10.100.3.6 reset_acquisition()	578
10.100.3.7 run_acquisition()	578
10.100.3.8 set_block_exp()	578
10.100.3.9 set_doppler_max()	578
10.100.3.10 set_doppler_step()	579
10.100.3.11 set_doppler_sweep()	579
10.100.3.12 set_local_code()	579
10.100.3.13 stop_acquisition()	579
10.100.3.14 write_local_code()	579
10.101 Fpga_buffer_monitor Class Reference	579
10.101.1 Detailed Description	580
10.101.2 Constructor & Destructor Documentation	580
10.101.2.1 Fpga_buffer_monitor()	580
10.101.2.2 ~Fpga_buffer_monitor()	580
10.101.3 Member Function Documentation	580
10.101.3.1 check_buffer_overflow_and_monitor_buffer_status()	580
10.102 Fpga_DMA Class Reference	581
10.102.1 Detailed Description	581
10.102.2 Constructor & Destructor Documentation	581
10.102.2.1 Fpga_DMA()	581
10.102.2.2 ~Fpga_DMA()	581
10.102.3 Member Function Documentation	582

10.102.3.1 DMA_close()	582
10.102.3.2 DMA_open()	582
10.102.3.3 DMA_write()	582
10.102.3.4 get_buffer_address()	582
10.103 Fpga_dynamic_bit_selection Class Reference	582
10.103.1 Detailed Description	583
10.103.2 Constructor & Destructor Documentation	583
10.103.2.1 Fpga_dynamic_bit_selection()	583
10.103.2.2 ~Fpga_dynamic_bit_selection()	583
10.103.3 Member Function Documentation	583
10.103.3.1 bit_selection()	583
10.104 Fpga_Multicorrelator_8sc Class Reference	583
10.104.1 Detailed Description	585
10.104.2 Constructor & Destructor Documentation	585
10.104.2.1 Fpga_Multicorrelator_8sc()	585
10.104.2.2 ~Fpga_Multicorrelator_8sc()	585
10.104.3 Member Function Documentation	585
10.104.3.1 Carrier_wipeoff_multicorrelator_resampler()	585
10.104.3.2 disable_secondary_codes()	586
10.104.3.3 enable_secondary_codes()	586
10.104.3.4 free()	586
10.104.3.5 initialize_secondary_code()	586
10.104.3.6 lock_channel()	586
10.104.3.7 open_channel()	586
10.104.3.8 read_sample_counter()	586
10.104.3.9 set_initial_sample()	586
10.104.3.10 set_local_code_and_taps()	586
10.104.3.11 set_output_vectors()	587
10.104.3.12 set_secondary_code_lengths()	587
10.104.3.13 unlock_channel()	587
10.104.3.14 update_local_code()	587
10.104.3.15 update_prn_code_length()	587
10.105 Fpga_spidev Class Reference	588
10.105.1 Detailed Description	588
10.105.2 Constructor & Destructor Documentation	588
10.105.2.1 Fpga_spidev()	588
10.105.2.2 ~Fpga_spidev()	588
10.105.3 Member Function Documentation	589
10.105.3.1 read_reg32()	589
10.105.3.2 SPI_close()	589
10.105.3.3 SPI_open()	589
10.105.3.4 write_reg32()	589

10.106 Fpga_Switch Class Reference	589
10.106.1 Detailed Description	590
10.106.2 Constructor & Destructor Documentation	590
10.106.2.1 Fpga_Switch()	590
10.106.2.2 ~Fpga_Switch()	590
10.106.3 Member Function Documentation	590
10.106.3.1 set_switch_position()	590
10.107 FreqXlatingFirFilter Class Reference	590
10.107.1 Detailed Description	593
10.107.2 Member Function Documentation	593
10.107.2.1 connect()	593
10.107.2.2 disconnect()	593
10.107.2.3 get_left_block()	593
10.107.2.4 get_right_block()	593
10.107.2.5 implementation()	593
10.107.2.6 item_size()	593
10.107.2.7 role()	593
10.108 ftp_t Struct Reference	594
10.108.1 Detailed Description	594
10.108.2 Member Data Documentation	595
10.108.2.1 addr	595
10.108.2.2 error	595
10.108.2.3 file	595
10.108.2.4 local	595
10.108.2.5 passwd	595
10.108.2.6 proto	595
10.108.2.7 state	595
10.108.2.8 thread	595
10.108.2.9 tnext	595
10.108.2.10 topts	595
10.108.2.11 user	595
10.109 Galileo_Almanac Class Reference	596
10.109.1 Detailed Description	598
10.109.2 Constructor & Destructor Documentation	598
10.109.2.1 Galileo_Almanac()	598
10.109.3 Member Function Documentation	599
10.109.3.1 serialize()	599
10.109.4 Member Data Documentation	599
10.109.4.1 E1B_HS	599
10.109.4.2 E5a_HS	599
10.109.4.3 E5b_HS	599
10.109.4.4 IODa	599

10.110 Galileo_Almanac_Helper Class Reference . . . . .	599
10.110.1 Detailed Description . . . . .	601
10.110.2 Constructor & Destructor Documentation . . . . .	601
10.110.2.1 Galileo_Almanac_Helper() . . . . .	601
10.110.3 Member Data Documentation . . . . .	601
10.110.3.1 af0_10 . . . . .	601
10.110.3.2 af0_8 . . . . .	601
10.110.3.3 af0_9 . . . . .	601
10.110.3.4 af1_10 . . . . .	602
10.110.3.5 af1_8 . . . . .	602
10.110.3.6 af1_9 . . . . .	602
10.110.3.7 DELTA_A_7 . . . . .	602
10.110.3.8 DELTA_A_8 . . . . .	602
10.110.3.9 DELTA_A_9 . . . . .	602
10.110.3.10 delta_i_7 . . . . .	602
10.110.3.11 delta_i_8 . . . . .	602
10.110.3.12 delta_i_9 . . . . .	602
10.110.3.13 E1B_HS_10 . . . . .	602
10.110.3.14 E1B_HS_8 . . . . .	602
10.110.3.15 E1B_HS_9 . . . . .	602
10.110.3.16 E5a_HS_10 . . . . .	603
10.110.3.17 E5a_HS_8 . . . . .	603
10.110.3.18 E5a_HS_9 . . . . .	603
10.110.3.19 E5b_HS_10 . . . . .	603
10.110.3.20 E5b_HS_8 . . . . .	603
10.110.3.21 E5b_HS_9 . . . . .	603
10.110.3.22 e_7 . . . . .	603
10.110.3.23 e_8 . . . . .	603
10.110.3.24 e_9 . . . . .	603
10.110.3.25 IOD_a_10 . . . . .	603
10.110.3.26 IOD_a_7 . . . . .	603
10.110.3.27 IOD_a_8 . . . . .	603
10.110.3.28 IOD_a_9 . . . . .	604
10.110.3.29 M0_10 . . . . .	604
10.110.3.30 M0_7 . . . . .	604
10.110.3.31 M0_9 . . . . .	604
10.110.3.32 Omega0_10 . . . . .	604
10.110.3.33 Omega0_7 . . . . .	604
10.110.3.34 Omega0_8 . . . . .	604
10.110.3.35 omega_7 . . . . .	604
10.110.3.36 omega_8 . . . . .	604
10.110.3.37 omega_9 . . . . .	604

10.110.3.38 Omega_dot_10 . . . . .	604
10.110.3.39 Omega_dot_7 . . . . .	604
10.110.3.40 Omega_dot_8 . . . . .	605
10.110.3.41 SVID1_7 . . . . .	605
10.110.3.42 SVID2_8 . . . . .	605
10.110.3.43 SVID3_9 . . . . .	605
10.110.3.44 t0a_7 . . . . .	605
10.110.3.45 t0a_9 . . . . .	605
10.110.3.46 WN_a_7 . . . . .	605
10.110.3.47 WN_a_9 . . . . .	605
10.111 Galileo_Cnav_Message Class Reference . . . . .	605
10.111.1 Detailed Description . . . . .	606
10.111.2 Member Function Documentation . . . . .	606
10.111.2.1 get_flag_CRC_test() . . . . .	606
10.111.2.2 get_HAS_encoded_page() . . . . .	606
10.111.2.3 have_new_HAS_page() . . . . .	606
10.111.2.4 is_HAS_in_test_mode() . . . . .	607
10.111.2.5 is_HAS_page_dummy() . . . . .	607
10.111.2.6 set_time_stamp() . . . . .	607
10.111.2.7 set_tow() . . . . .	607
10.112 Galileo_E1_Tcp_Connector_Tracking_cc Class Reference . . . . .	607
10.112.1 Detailed Description . . . . .	608
10.113 galileo_e5a_noncoherentIQ_acquisition_caf_cc Class Reference . . . . .	608
10.113.1 Detailed Description . . . . .	611
10.113.2 Constructor & Destructor Documentation . . . . .	611
10.113.2.1 ~galileo_e5a_noncoherentIQ_acquisition_caf_cc() . . . . .	611
10.113.3 Member Function Documentation . . . . .	611
10.113.3.1 general_work() . . . . .	611
10.113.3.2 init() . . . . .	611
10.113.3.3 mag() . . . . .	611
10.113.3.4 set_active() . . . . .	612
10.113.3.5 set_channel() . . . . .	612
10.113.3.6 set_channel_fsm() . . . . .	612
10.113.3.7 set_doppler_max() . . . . .	612
10.113.3.8 set_doppler_step() . . . . .	612
10.113.3.9 set_gnss_synchro() . . . . .	613
10.113.3.10 set_local_code() . . . . .	614
10.113.3.11 set_state() . . . . .	614
10.113.3.12 set_threshold() . . . . .	614
10.114 galileo_e6_has_msg_receiver Class Reference . . . . .	614
10.114.1 Detailed Description . . . . .	616
10.114.2 Constructor & Destructor Documentation . . . . .	616



10.114.2.1 ~galileo_e6_has_msg_receiver()	616
10.114.3 Member Function Documentation	616
10.114.3.1 process_test_page()	616
10.115 Galileo_Ephemeris Class Reference	616
10.115.1 Detailed Description	621
10.115.2 Constructor & Destructor Documentation	621
10.115.2.1 Galileo_Ephemeris()	621
10.115.3 Member Function Documentation	621
10.115.3.1 Galileo_System_Time()	621
10.115.3.2 serialize()	621
10.115.4 Member Data Documentation	621
10.115.4.1 BGD_E1E5a	621
10.115.4.2 BGD_E1E5b	621
10.115.4.3 E1B_DVS	622
10.115.4.4 E1B_HS	622
10.115.4.5 E5a_DVS	622
10.115.4.6 E5a_HS	622
10.115.4.7 E5b_DVS	622
10.115.4.8 E5b_HS	622
10.115.4.9 flag_all_ephemeris	622
10.115.4.10 IOD_ephemeris	622
10.115.4.11 IOD_nav	622
10.115.4.12 SISA	623
10.116 Galileo_Fnav_Message Class Reference	623
10.116.1 Detailed Description	624
10.116.2 Member Function Documentation	624
10.116.2.1 get_flag_CRC_test()	624
10.116.2.2 get_flag_TOW_set()	624
10.116.2.3 get_TOW1()	624
10.116.2.4 get_TOW2()	624
10.116.2.5 get_TOW3()	624
10.116.2.6 get_TOW4()	624
10.116.2.7 is_TOW1_set()	624
10.116.2.8 is_TOW2_set()	625
10.116.2.9 is_TOW3_set()	625
10.116.2.10 is_TOW4_set()	625
10.116.2.11 set_flag_TOW_set()	625
10.116.2.12 set_TOW1_flag()	625
10.116.2.13 set_TOW2_flag()	625
10.116.2.14 set_TOW3_flag()	625
10.116.2.15 set_TOW4_flag()	625
10.117 Galileo_HAS_data Class Reference	625

10.117.1 Detailed Description . . . . .	629
10.117.2 Member Function Documentation . . . . .	629
10.117.2.1 get_clock_correction_mult_m() . . . . .	629
10.117.2.2 get_clock_subset_correction_mult_m() . . . . .	629
10.117.2.3 get_code_bias_m() [1/2] . . . . .	629
10.117.2.4 get_code_bias_m() [2/2] . . . . .	629
10.117.2.5 get_delta_clock_correction_m() [1/2] . . . . .	629
10.117.2.6 get_delta_clock_correction_m() [2/2] . . . . .	629
10.117.2.7 get_delta_clock_subset_correction_m() [1/2] . . . . .	630
10.117.2.8 get_delta_clock_subset_correction_m() [2/2] . . . . .	630
10.117.2.9 get_delta_cross_track_m() [1/3] . . . . .	630
10.117.2.10 get_delta_cross_track_m() [2/3] . . . . .	630
10.117.2.11 get_delta_cross_track_m() [3/3] . . . . .	630
10.117.2.12 get_delta_in_track_m() [1/3] . . . . .	630
10.117.2.13 get_delta_in_track_m() [2/3] . . . . .	630
10.117.2.14 get_delta_in_track_m() [3/3] . . . . .	630
10.117.2.15 get_delta_radial_m() [1/3] . . . . .	630
10.117.2.16 get_delta_radial_m() [2/3] . . . . .	630
10.117.2.17 get_delta_radial_m() [3/3] . . . . .	631
10.117.2.18 get_gnss_id() . . . . .	631
10.117.2.19 get_gnss_iod() [1/2] . . . . .	631
10.117.2.20 get_gnss_iod() [2/2] . . . . .	631
10.117.2.21 get_nsat() . . . . .	631
10.117.2.22 get_nsat_sub() . . . . .	631
10.117.2.23 get_num_satellites() . . . . .	631
10.117.2.24 get_num_subset_satellites() . . . . .	631
10.117.2.25 get_phase_bias_cycle() [1/2] . . . . .	631
10.117.2.26 get_phase_bias_cycle() [2/2] . . . . .	631
10.117.2.27 get_PRNs_in_mask() [1/2] . . . . .	632
10.117.2.28 get_PRNs_in_mask() [2/2] . . . . .	632
10.117.2.29 get_PRNs_in_submask() . . . . .	632
10.117.2.30 get_signals_in_mask() [1/2] . . . . .	632
10.117.2.31 get_signals_in_mask() [2/2] . . . . .	632
10.117.2.32 get_systems_string() . . . . .	632
10.117.2.33 get_systems_subset_string() . . . . .	632
10.117.2.34 get_validity_interval_s() . . . . .	632
10.117.3 Member Data Documentation . . . . .	632
10.117.3.1 cell_mask . . . . .	632
10.117.3.2 cell_mask_availability_flag . . . . .	632
10.117.3.3 code_bias . . . . .	633
10.117.3.4 delta_clock_correction . . . . .	633
10.117.3.5 delta_clock_correction_clock_subset . . . . .	633

10.117.3.6 delta_clock_multiplier . . . . .	633
10.117.3.7 delta_clock_multiplier_clock_subset . . . . .	633
10.117.3.8 delta_cross_track . . . . .	633
10.117.3.9 delta_in_track . . . . .	633
10.117.3.10 delta_radial . . . . .	633
10.117.3.11 gnss_id_clock_subset . . . . .	633
10.117.3.12 gnss_id_mask . . . . .	633
10.117.3.13 gnss_iod . . . . .	634
10.117.3.14 has_status . . . . .	634
10.117.3.15 header . . . . .	634
10.117.3.16 message_id . . . . .	634
10.117.3.17 nav_message . . . . .	634
10.117.3.18 Nsys . . . . .	634
10.117.3.19 Nsys_sub . . . . .	634
10.117.3.20 phase_bias . . . . .	634
10.117.3.21 phase_discontinuity_indicator . . . . .	634
10.117.3.22 satellite_mask . . . . .	634
10.117.3.23 satellite_submask . . . . .	635
10.117.3.24 signal_mask . . . . .	635
10.117.3.25 tow . . . . .	635
10.117.3.26 validity_interval_index_clock_fullset_corrections . . . . .	635
10.117.3.27 validity_interval_index_clock_subset_corrections . . . . .	635
10.117.3.28 validity_interval_index_code_bias_corrections . . . . .	635
10.117.3.29 validity_interval_index_orbit_corrections . . . . .	635
10.117.3.30 validity_interval_index_phase_bias_corrections . . . . .	635
10.118 Galileo_HAS_page Class Reference . . . . .	635
10.118.1 Detailed Description . . . . .	637
10.118.2 Member Data Documentation . . . . .	637
10.118.2.1 has_message_string . . . . .	637
10.118.2.2 has_status . . . . .	637
10.118.2.3 message_id . . . . .	637
10.118.2.4 message_page_id . . . . .	637
10.118.2.5 message_size . . . . .	637
10.118.2.6 message_type . . . . .	637
10.118.2.7 reserved . . . . .	637
10.118.2.8 time_stamp . . . . .	637
10.118.2.9 tow . . . . .	638
10.119 Galileo_Inav_Message Class Reference . . . . .	638
10.119.1 Detailed Description . . . . .	639
10.119.2 Member Function Documentation . . . . .	639
10.119.2.1 enable_reed_solomon() . . . . .	639
10.119.2.2 get_A0G() . . . . .	639

10.119.2.3 get_A1G()	639
10.119.2.4 get_flag_CRC_test()	639
10.119.2.5 get_flag_GGTO()	640
10.119.2.6 get_flag_TOW_set()	640
10.119.2.7 get_Galileo_week()	640
10.119.2.8 get_t0G()	640
10.119.2.9 get_TOW0()	640
10.119.2.10 get_TOW5()	640
10.119.2.11 get_TOW6()	640
10.119.2.12 get_WN0G()	640
10.119.2.13 init_PRN()	640
10.119.2.14 is_TOW0_set()	640
10.119.2.15 is_TOW5_set()	640
10.119.2.16 is_TOW6_set()	640
10.119.2.17 set_flag_TOW_set()	641
10.119.2.18 set_TOW0_flag()	641
10.119.2.19 set_TOW5_flag()	641
10.119.2.20 set_TOW6_flag()	641
10.120 Galileo_Iono Class Reference	641
10.120.1 Detailed Description	642
10.120.2 Constructor & Destructor Documentation	642
10.120.2.1 Galileo_Iono()	642
10.120.3 Member Function Documentation	642
10.120.3.1 serialize()	642
10.120.4 Member Data Documentation	643
10.120.4.1 ai0	643
10.120.4.2 ai1	643
10.120.4.3 ai2	643
10.120.4.4 Region1_flag	643
10.120.4.5 Region2_flag	643
10.120.4.6 Region3_flag	643
10.120.4.7 Region4_flag	643
10.120.4.8 Region5_flag	643
10.120.4.9 tow	644
10.120.4.10 WN	644
10.121 Galileo_ISM Class Reference	644
10.121.1 Detailed Description	645
10.121.2 Constructor & Destructor Documentation	645
10.121.2.1 Galileo_ISM()	645
10.122 galileo_pcps_8ms_acquisition_cc Class Reference	645
10.122.1 Detailed Description	648
10.122.2 Constructor & Destructor Documentation	648

10.122.2.1 ~galileo_pcps_8ms_acquisition_cc()	648
10.122.3 Member Function Documentation	648
10.122.3.1 general_work()	648
10.122.3.2 init()	648
10.122.3.3 mag()	648
10.122.3.4 set_active()	648
10.122.3.5 set_channel()	649
10.122.3.6 set_channel_fsm()	649
10.122.3.7 set_doppler_max()	649
10.122.3.8 set_doppler_step()	649
10.122.3.9 set_gnss_synchro()	649
10.122.3.10 set_local_code()	650
10.122.3.11 set_state()	650
10.122.3.12 set_threshold()	650
10.123 Galileo_Reduced_CED Class Reference	650
10.123.1 Detailed Description	652
10.123.2 Constructor & Destructor Documentation	652
10.123.2.1 Galileo_Reduced_CED()	652
10.123.3 Member Function Documentation	652
10.123.3.1 compute_eph()	652
10.123.4 Member Data Documentation	652
10.123.4.1 af0red	652
10.123.4.2 af1red	652
10.123.4.3 DeltaAred	652
10.123.4.4 Deltai0red	652
10.123.4.5 exred	652
10.123.4.6 eyred	652
10.123.4.7 IODnav	653
10.123.4.8 lambda0red	653
10.123.4.9 Omega0red	653
10.123.4.10 PRN	653
10.123.4.11 TOTRedCED	653
10.124 galileo_telemetry_decoder_gs Class Reference	653
10.124.1 Detailed Description	655
10.124.2 Member Function Documentation	655
10.124.2.1 general_work()	655
10.124.2.2 set_channel()	655
10.124.2.3 set_satellite()	655
10.125 galileo_tow_map Class Reference	656
10.125.1 Detailed Description	656
10.125.2 Constructor & Destructor Documentation	657
10.125.2.1 ~galileo_tow_map()	657

10.126 Galileo_Utc_Model Class Reference . . . . .	657
10.126.1 Detailed Description . . . . .	658
10.126.2 Constructor & Destructor Documentation . . . . .	658
10.126.2.1 Galileo_Utc_Model() . . . . .	658
10.126.3 Member Function Documentation . . . . .	658
10.126.3.1 GST_to_UTC_time() . . . . .	658
10.126.3.2 serialize() . . . . .	658
10.126.4 Member Data Documentation . . . . .	658
10.126.4.1 A0 . . . . .	658
10.126.4.2 A1 . . . . .	659
10.126.4.3 A_0G . . . . .	659
10.126.4.4 A_1G . . . . .	659
10.126.4.5 Delta_tLS . . . . .	659
10.126.4.6 Delta_tLSF . . . . .	659
10.126.4.7 DN . . . . .	659
10.126.4.8 flag_utc_model . . . . .	659
10.126.4.9 t_0G . . . . .	659
10.126.4.10 tot . . . . .	659
10.126.4.11 WN_0G . . . . .	659
10.126.4.12 WN_LSF . . . . .	659
10.126.4.13 WNot . . . . .	660
10.127 GalileoE1BTelemetryDecoder Class Reference . . . . .	660
10.127.1 Detailed Description . . . . .	663
10.127.2 Member Function Documentation . . . . .	663
10.127.2.1 connect() . . . . .	663
10.127.2.2 disconnect() . . . . .	663
10.127.2.3 get_left_block() . . . . .	663
10.127.2.4 get_right_block() . . . . .	663
10.127.2.5 implementation() . . . . .	663
10.127.2.6 item_size() . . . . .	663
10.127.2.7 reset() . . . . .	664
10.127.2.8 role() . . . . .	664
10.127.2.9 set_channel() . . . . .	664
10.127.2.10 set_satellite() . . . . .	664
10.128 GalileoE1DIPIIVemlTracking Class Reference . . . . .	664
10.128.1 Detailed Description . . . . .	667
10.128.2 Member Function Documentation . . . . .	667
10.128.2.1 connect() . . . . .	667
10.128.2.2 disconnect() . . . . .	667
10.128.2.3 get_left_block() . . . . .	667
10.128.2.4 get_right_block() . . . . .	667
10.128.2.5 implementation() . . . . .	667

10.128.2.6 item_size()	668
10.128.2.7 role()	668
10.128.2.8 set_channel()	668
10.128.2.9 set_gnss_synchro()	668
10.128.2.10 start_tracking()	668
10.128.2.11 stop_tracking()	668
10.129 GalileoE1DIIPIIVemlTrackingFpga Class Reference	668
10.129.1 Detailed Description	671
10.129.2 Constructor & Destructor Documentation	671
10.129.2.1 GalileoE1DIIPIIVemlTrackingFpga()	671
10.129.2.2 ~GalileoE1DIIPIIVemlTrackingFpga()	672
10.129.3 Member Function Documentation	672
10.129.3.1 connect()	672
10.129.3.2 disconnect()	672
10.129.3.3 get_left_block()	672
10.129.3.4 get_right_block()	672
10.129.3.5 implementation()	672
10.129.3.6 item_size()	672
10.129.3.7 role()	672
10.129.3.8 set_channel()	672
10.129.3.9 set_gnss_synchro()	673
10.129.3.10 start_tracking()	673
10.129.3.11 stop_tracking()	673
10.130 GalileoE1Pcps8msAmbiguousAcquisition Class Reference	673
10.130.1 Detailed Description	676
10.130.2 Member Function Documentation	676
10.130.2.1 connect()	676
10.130.2.2 disconnect()	677
10.130.2.3 get_left_block()	677
10.130.2.4 get_right_block()	677
10.130.2.5 implementation()	677
10.130.2.6 init()	677
10.130.2.7 item_size()	677
10.130.2.8 mag()	677
10.130.2.9 reset()	677
10.130.2.10 role()	677
10.130.2.11 set_channel()	678
10.130.2.12 set_channel_fsm()	678
10.130.2.13 set_doppler_max()	678
10.130.2.14 set_doppler_step()	678
10.130.2.15 set_gnss_synchro()	678
10.130.2.16 set_local_code()	678

10.130.2.17 set_resampler_latency()	678
10.130.2.18 set_state()	678
10.130.2.19 set_threshold()	678
10.130.2.20 stop_acquisition()	679
10.131 GalileoE1PcpsAmbiguousAcquisition Class Reference	679
10.131.1 Detailed Description	682
10.131.2 Member Function Documentation	683
10.131.2.1 connect()	683
10.131.2.2 disconnect()	683
10.131.2.3 get_left_block()	683
10.131.2.4 get_right_block()	683
10.131.2.5 implementation()	683
10.131.2.6 init()	683
10.131.2.7 item_size()	683
10.131.2.8 mag()	683
10.131.2.9 reset()	683
10.131.2.10 role()	683
10.131.2.11 set_channel()	684
10.131.2.12 set_channel_fsm()	684
10.131.2.13 set_doppler_center()	684
10.131.2.14 set_doppler_max()	684
10.131.2.15 set_doppler_step()	684
10.131.2.16 set_gnss_synchro()	684
10.131.2.17 set_local_code()	684
10.131.2.18 set_resampler_latency()	684
10.131.2.19 set_state()	685
10.131.2.20 set_threshold()	685
10.131.2.21 stop_acquisition()	685
10.132 GalileoE1PcpsAmbiguousAcquisitionFpga Class Reference	685
10.132.1 Detailed Description	689
10.132.2 Constructor & Destructor Documentation	689
10.132.2.1 GalileoE1PcpsAmbiguousAcquisitionFpga()	689
10.132.2.2 ~GalileoE1PcpsAmbiguousAcquisitionFpga()	689
10.132.3 Member Function Documentation	689
10.132.3.1 connect()	689
10.132.3.2 disconnect()	689
10.132.3.3 get_left_block()	689
10.132.3.4 get_right_block()	689
10.132.3.5 implementation()	690
10.132.3.6 init()	690
10.132.3.7 item_size()	690
10.132.3.8 mag()	690



10.132.3.9 reset()	690
10.132.3.10 role()	690
10.132.3.11 set_channel()	690
10.132.3.12 set_channel_fsm()	690
10.132.3.13 set_doppler_center()	691
10.132.3.14 set_doppler_max()	691
10.132.3.15 set_doppler_step()	691
10.132.3.16 set_gnss_synchro()	691
10.132.3.17 set_local_code()	691
10.132.3.18 set_resampler_latency()	691
10.132.3.19 set_state()	691
10.132.3.20 set_threshold()	691
10.132.3.21 stop_acquisition()	691
10.133 GalileoE1PcpsCccwsrAmbiguousAcquisition Class Reference	692
10.133.1 Detailed Description	694
10.133.2 Member Function Documentation	694
10.133.2.1 connect()	694
10.133.2.2 disconnect()	695
10.133.2.3 get_left_block()	695
10.133.2.4 get_right_block()	695
10.133.2.5 implementation()	695
10.133.2.6 init()	695
10.133.2.7 item_size()	695
10.133.2.8 mag()	695
10.133.2.9 reset()	695
10.133.2.10 role()	695
10.133.2.11 set_channel()	696
10.133.2.12 set_channel_fsm()	696
10.133.2.13 set_doppler_max()	696
10.133.2.14 set_doppler_step()	696
10.133.2.15 set_gnss_synchro()	696
10.133.2.16 set_local_code()	696
10.133.2.17 set_resampler_latency()	696
10.133.2.18 set_state()	696
10.133.2.19 set_threshold()	696
10.133.2.20 stop_acquisition()	697
10.134 GalileoE1PcpsQuickSyncAmbiguousAcquisition Class Reference	697
10.134.1 Detailed Description	700
10.134.2 Member Function Documentation	700
10.134.2.1 connect()	700
10.134.2.2 disconnect()	701
10.134.2.3 get_left_block()	701

10.134.2.4 get_right_block()	701
10.134.2.5 implementation()	701
10.134.2.6 init()	701
10.134.2.7 item_size()	701
10.134.2.8 mag()	701
10.134.2.9 reset()	701
10.134.2.10 role()	701
10.134.2.11 set_channel()	702
10.134.2.12 set_channel_fsm()	702
10.134.2.13 set_doppler_max()	702
10.134.2.14 set_doppler_step()	702
10.134.2.15 set_gnss_synchro()	702
10.134.2.16 set_local_code()	702
10.134.2.17 set_resampler_latency()	702
10.134.2.18 set_state()	702
10.134.2.19 set_threshold()	703
10.134.2.20 stop_acquisition()	703
10.135 GalileoE1PcpsTongAmbiguousAcquisition Class Reference	703
10.135.1 Detailed Description	706
10.135.2 Member Function Documentation	706
10.135.2.1 connect()	706
10.135.2.2 disconnect()	707
10.135.2.3 get_left_block()	707
10.135.2.4 get_right_block()	707
10.135.2.5 implementation()	707
10.135.2.6 init()	707
10.135.2.7 item_size()	707
10.135.2.8 mag()	707
10.135.2.9 reset()	707
10.135.2.10 role()	707
10.135.2.11 set_channel()	708
10.135.2.12 set_channel_fsm()	708
10.135.2.13 set_doppler_max()	708
10.135.2.14 set_doppler_step()	708
10.135.2.15 set_gnss_synchro()	708
10.135.2.16 set_local_code()	708
10.135.2.17 set_resampler_latency()	708
10.135.2.18 set_state()	708
10.135.2.19 set_threshold()	709
10.135.2.20 stop_acquisition()	709
10.136 GalileoE1TcpConnectorTracking Class Reference	709
10.136.1 Detailed Description	712

10.136.2 Member Function Documentation	712
10.136.2.1 connect()	712
10.136.2.2 disconnect()	712
10.136.2.3 get_left_block()	712
10.136.2.4 get_right_block()	712
10.136.2.5 implementation()	712
10.136.2.6 item_size()	713
10.136.2.7 role()	713
10.136.2.8 set_channel()	713
10.136.2.9 set_gnss_synchro()	713
10.136.2.10 start_tracking()	713
10.136.2.11 stop_tracking()	713
10.137 GalileoE5aDIIPIITracking Class Reference	713
10.137.1 Detailed Description	716
10.137.2 Member Function Documentation	716
10.137.2.1 connect()	716
10.137.2.2 disconnect()	716
10.137.2.3 get_left_block()	716
10.137.2.4 get_right_block()	716
10.137.2.5 implementation()	716
10.137.2.6 item_size()	717
10.137.2.7 role()	717
10.137.2.8 set_channel()	717
10.137.2.9 set_gnss_synchro()	717
10.137.2.10 start_tracking()	717
10.137.2.11 stop_tracking()	717
10.138 GalileoE5aDIIPIITrackingFpga Class Reference	717
10.138.1 Detailed Description	720
10.138.2 Constructor & Destructor Documentation	720
10.138.2.1 GalileoE5aDIIPIITrackingFpga()	720
10.138.2.2 ~GalileoE5aDIIPIITrackingFpga()	721
10.138.3 Member Function Documentation	721
10.138.3.1 connect()	721
10.138.3.2 disconnect()	721
10.138.3.3 get_left_block()	721
10.138.3.4 get_right_block()	721
10.138.3.5 implementation()	721
10.138.3.6 item_size()	721
10.138.3.7 role()	721
10.138.3.8 set_channel()	721
10.138.3.9 set_gnss_synchro()	722
10.138.3.10 start_tracking()	722

10.138.3.11 stop_tracking()	722
10.139 GalileoE5aNoncoherentIQAcquisitionCaf Class Reference	723
10.139.1 Detailed Description	725
10.139.2 Member Function Documentation	725
10.139.2.1 connect()	725
10.139.2.2 disconnect()	726
10.139.2.3 get_left_block()	726
10.139.2.4 get_right_block()	726
10.139.2.5 implementation()	726
10.139.2.6 init()	726
10.139.2.7 item_size()	726
10.139.2.8 mag()	726
10.139.2.9 reset()	726
10.139.2.10 role()	726
10.139.2.11 set_channel()	727
10.139.2.12 set_channel_fsm()	727
10.139.2.13 set_doppler_max()	727
10.139.2.14 set_doppler_step()	727
10.139.2.15 set_gnss_synchro()	727
10.139.2.16 set_local_code()	727
10.139.2.17 set_resampler_latency()	727
10.139.2.18 set_state()	727
10.139.2.19 set_threshold()	728
10.139.2.20 stop_acquisition()	728
10.140 GalileoE5aPcpsAcquisition Class Reference	729
10.140.1 Detailed Description	731
10.140.2 Member Function Documentation	731
10.140.2.1 connect()	731
10.140.2.2 disconnect()	732
10.140.2.3 get_left_block()	732
10.140.2.4 get_right_block()	732
10.140.2.5 implementation()	732
10.140.2.6 init()	732
10.140.2.7 item_size()	732
10.140.2.8 mag()	732
10.140.2.9 reset()	732
10.140.2.10 role()	732
10.140.2.11 set_channel()	732
10.140.2.12 set_channel_fsm()	733
10.140.2.13 set_doppler_center()	733
10.140.2.14 set_doppler_max()	733
10.140.2.15 set_doppler_step()	733

10.140.2.16 set_gnss_synchro()	733
10.140.2.17 set_local_code()	733
10.140.2.18 set_resampler_latency()	733
10.140.2.19 set_state()	733
10.140.2.20 set_threshold()	734
10.140.2.21 stop_acquisition()	734
10.141 GalileoE5aPcpsAcquisitionFpga Class Reference	734
10.141.1 Detailed Description	738
10.141.2 Constructor & Destructor Documentation	738
10.141.2.1 GalileoE5aPcpsAcquisitionFpga()	738
10.141.2.2 ~GalileoE5aPcpsAcquisitionFpga()	738
10.141.3 Member Function Documentation	738
10.141.3.1 connect()	738
10.141.3.2 disconnect()	738
10.141.3.3 get_left_block()	739
10.141.3.4 get_right_block()	739
10.141.3.5 implementation()	739
10.141.3.6 init()	739
10.141.3.7 item_size()	739
10.141.3.8 mag()	739
10.141.3.9 reset()	739
10.141.3.10 role()	739
10.141.3.11 set_channel()	739
10.141.3.12 set_channel_fsm()	740
10.141.3.13 set_doppler_center()	740
10.141.3.14 set_doppler_max()	740
10.141.3.15 set_doppler_step()	740
10.141.3.16 set_gnss_synchro()	740
10.141.3.17 set_local_code()	740
10.141.3.18 set_resampler_latency()	740
10.141.3.19 set_single_doppler_flag()	740
10.141.3.20 set_state()	740
10.141.3.21 set_threshold()	741
10.141.3.22 stop_acquisition()	741
10.142 GalileoE5aTelemetryDecoder Class Reference	741
10.142.1 Detailed Description	744
10.142.2 Member Function Documentation	744
10.142.2.1 connect()	744
10.142.2.2 disconnect()	744
10.142.2.3 get_left_block()	744
10.142.2.4 get_right_block()	744
10.142.2.5 implementation()	744

10.142.2.6 item_size()	744
10.142.2.7 reset()	745
10.142.2.8 role()	745
10.142.2.9 set_channel()	745
10.142.2.10 set_satellite()	745
10.143 GalileoE5bDIIPIITracking Class Reference	745
10.143.1 Detailed Description	748
10.143.2 Member Function Documentation	748
10.143.2.1 connect()	748
10.143.2.2 disconnect()	748
10.143.2.3 get_left_block()	748
10.143.2.4 get_right_block()	748
10.143.2.5 implementation()	749
10.143.2.6 item_size()	749
10.143.2.7 role()	749
10.143.2.8 set_channel()	749
10.143.2.9 set_gnss_synchro()	749
10.143.2.10 start_tracking()	749
10.143.2.11 stop_tracking()	749
10.144 GalileoE5bPcpsAcquisition Class Reference	750
10.144.1 Detailed Description	753
10.144.2 Constructor & Destructor Documentation	753
10.144.2.1 GalileoE5bPcpsAcquisition()	753
10.144.2.2 ~GalileoE5bPcpsAcquisition()	753
10.144.3 Member Function Documentation	753
10.144.3.1 connect()	753
10.144.3.2 disconnect()	753
10.144.3.3 get_left_block()	753
10.144.3.4 get_right_block()	753
10.144.3.5 implementation()	754
10.144.3.6 init()	754
10.144.3.7 item_size()	754
10.144.3.8 mag()	754
10.144.3.9 reset()	754
10.144.3.10 role()	754
10.144.3.11 set_channel()	754
10.144.3.12 set_channel_fsm()	754
10.144.3.13 set_doppler_center()	754
10.144.3.14 set_doppler_max()	755
10.144.3.15 set_doppler_step()	755
10.144.3.16 set_gnss_synchro()	755
10.144.3.17 set_local_code()	755

10.144.3.18 set_resampler_latency()	755
10.144.3.19 set_state()	755
10.144.3.20 set_threshold()	755
10.144.3.21 stop_acquisition()	756
10.145 GalileoE5bPcpsAcquisitionFpga Class Reference	756
10.145.1 Detailed Description	760
10.145.2 Constructor & Destructor Documentation	760
10.145.2.1 GalileoE5bPcpsAcquisitionFpga()	760
10.145.2.2 ~GalileoE5bPcpsAcquisitionFpga()	760
10.145.3 Member Function Documentation	760
10.145.3.1 connect()	760
10.145.3.2 disconnect()	760
10.145.3.3 get_left_block()	761
10.145.3.4 get_right_block()	761
10.145.3.5 implementation()	761
10.145.3.6 init()	761
10.145.3.7 item_size()	761
10.145.3.8 mag()	761
10.145.3.9 reset()	761
10.145.3.10 role()	761
10.145.3.11 set_channel()	761
10.145.3.12 set_channel_fsm()	762
10.145.3.13 set_doppler_center()	762
10.145.3.14 set_doppler_max()	762
10.145.3.15 set_doppler_step()	762
10.145.3.16 set_gnss_synchro()	762
10.145.3.17 set_local_code()	762
10.145.3.18 set_resampler_latency()	762
10.145.3.19 set_single_doppler_flag()	762
10.145.3.20 set_state()	762
10.145.3.21 set_threshold()	763
10.145.3.22 stop_acquisition()	763
10.146 GalileoE5bTelemetryDecoder Class Reference	763
10.146.1 Detailed Description	766
10.146.2 Member Function Documentation	766
10.146.2.1 connect()	766
10.146.2.2 disconnect()	766
10.146.2.3 get_left_block()	766
10.146.2.4 get_right_block()	766
10.146.2.5 implementation()	766
10.146.2.6 item_size()	767
10.146.2.7 reset()	767

10.146.2.8 role()	767
10.146.2.9 set_channel()	767
10.146.2.10 set_satellite()	767
10.147 GalileoE6DIIPIITracking Class Reference	767
10.147.1 Detailed Description	770
10.147.2 Member Function Documentation	770
10.147.2.1 connect()	770
10.147.2.2 disconnect()	770
10.147.2.3 get_left_block()	770
10.147.2.4 get_right_block()	770
10.147.2.5 implementation()	771
10.147.2.6 item_size()	771
10.147.2.7 role()	771
10.147.2.8 set_channel()	771
10.147.2.9 set_gnss_synchro()	771
10.147.2.10 start_tracking()	771
10.147.2.11 stop_tracking()	771
10.148 GalileoE6PcpsAcquisition Class Reference	771
10.148.1 Detailed Description	774
10.148.2 Member Function Documentation	775
10.148.2.1 connect()	775
10.148.2.2 disconnect()	775
10.148.2.3 get_left_block()	775
10.148.2.4 get_right_block()	775
10.148.2.5 implementation()	775
10.148.2.6 init()	775
10.148.2.7 item_size()	775
10.148.2.8 mag()	775
10.148.2.9 reset()	775
10.148.2.10 role()	775
10.148.2.11 set_channel()	776
10.148.2.12 set_channel_fsm()	776
10.148.2.13 set_doppler_center()	776
10.148.2.14 set_doppler_max()	776
10.148.2.15 set_doppler_step()	776
10.148.2.16 set_gnss_synchro()	776
10.148.2.17 set_local_code()	776
10.148.2.18 set_resampler_latency()	776
10.148.2.19 set_state()	777
10.148.2.20 set_threshold()	777
10.148.2.21 stop_acquisition()	777
10.149 GalileoE6TelemetryDecoder Class Reference	777



10.149.1 Detailed Description . . . . .	780
10.149.2 Member Function Documentation . . . . .	780
10.149.2.1 connect() . . . . .	780
10.149.2.2 disconnect() . . . . .	780
10.149.2.3 get_left_block() . . . . .	780
10.149.2.4 get_right_block() . . . . .	780
10.149.2.5 implementation() . . . . .	780
10.149.2.6 item_size() . . . . .	781
10.149.2.7 reset() . . . . .	781
10.149.2.8 role() . . . . .	781
10.149.2.9 set_channel() . . . . .	781
10.149.2.10 set_satellite() . . . . .	781
10.150 GenSignalSource Class Reference . . . . .	781
10.150.1 Detailed Description . . . . .	784
10.150.2 Constructor & Destructor Documentation . . . . .	784
10.150.2.1 GenSignalSource() . . . . .	784
10.150.3 Member Function Documentation . . . . .	784
10.150.3.1 connect() . . . . .	784
10.150.3.2 disconnect() . . . . .	784
10.150.3.3 get_left_block() . . . . .	784
10.150.3.4 get_right_block() . . . . .	784
10.150.3.5 getRfChannels() . . . . .	784
10.150.3.6 implementation() . . . . .	785
10.150.3.7 item_size() . . . . .	785
10.150.3.8 role() . . . . .	785
10.150.3.9 signal_generator() . . . . .	785
10.151 Geohash Class Reference . . . . .	785
10.151.1 Detailed Description . . . . .	785
10.151.2 Member Function Documentation . . . . .	786
10.151.2.1 decode() . . . . .	786
10.151.2.2 encode() . . . . .	786
10.152 GeoJSON_Printer Class Reference . . . . .	786
10.152.1 Detailed Description . . . . .	787
10.153 geph_t Struct Reference . . . . .	788
10.153.1 Detailed Description . . . . .	789
10.153.2 Member Data Documentation . . . . .	789
10.153.2.1 acc . . . . .	789
10.153.2.2 age . . . . .	789
10.153.2.3 dtaun . . . . .	789
10.153.2.4 frq . . . . .	789
10.153.2.5 gamn . . . . .	789
10.153.2.6 iode . . . . .	789

10.153.2.7 pos . . . . .	789
10.153.2.8 sat . . . . .	789
10.153.2.9 sva . . . . .	789
10.153.2.10 svh . . . . .	789
10.153.2.11 taun . . . . .	789
10.153.2.12 toe . . . . .	790
10.153.2.13 tof . . . . .	790
10.153.2.14 vel . . . . .	790
10.154 Glonass_Gnav_Almanac Class Reference . . . . .	790
10.154.1 Detailed Description . . . . .	791
10.154.2 Constructor & Destructor Documentation . . . . .	792
10.154.2.1 Glonass_Gnav_Almanac() . . . . .	792
10.154.3 Member Function Documentation . . . . .	792
10.154.3.1 serialize() . . . . .	792
10.154.4 Member Data Documentation . . . . .	792
10.154.4.1 d_C_n . . . . .	792
10.154.4.2 d_Delta_i_n_A . . . . .	792
10.154.4.3 d_Delta_T_n_A . . . . .	792
10.154.4.4 d_Delta_T_n_A_dot . . . . .	792
10.154.4.5 d_epsilon_n_A . . . . .	792
10.154.4.6 d_H_n_A . . . . .	793
10.154.4.7 d_KP . . . . .	793
10.154.4.8 d_I_n . . . . .	793
10.154.4.9 d_lambda_n_A . . . . .	793
10.154.4.10 d_M_n_A . . . . .	793
10.154.4.11 d_n_A . . . . .	793
10.154.4.12 d_omega_n_A . . . . .	793
10.154.4.13 d_t_lambda_n_A . . . . .	793
10.154.4.14 d_tau_n_A . . . . .	794
10.154.4.15 i_satellite_freq_channel . . . . .	794
10.154.4.16 i_satellite_slot_number . . . . .	794
10.154.4.17 PRN . . . . .	794
10.155 Glonass_Gnav_Ephemeris Class Reference . . . . .	794
10.155.1 Detailed Description . . . . .	797
10.155.2 Constructor & Destructor Documentation . . . . .	797
10.155.2.1 Glonass_Gnav_Ephemeris() . . . . .	797
10.155.3 Member Function Documentation . . . . .	798
10.155.3.1 compute_GLONASS_time() . . . . .	798
10.155.3.2 glot_to_gpst() . . . . .	798
10.155.3.3 glot_to_utc() . . . . .	798
10.155.3.4 serialize() . . . . .	798
10.155.3.5 sv_clock_drift() . . . . .	799

10.155.4 Member Data Documentation . . . . .	799
10.155.4.1 d_AXn . . . . .	799
10.155.4.2 d_AYn . . . . .	799
10.155.4.3 d_AZn . . . . .	800
10.155.4.4 d_B_n . . . . .	800
10.155.4.5 d_Delta_tau_n . . . . .	800
10.155.4.6 d_dtr . . . . .	800
10.155.4.7 d_E_n . . . . .	800
10.155.4.8 d_F_T . . . . .	800
10.155.4.9 d_gamma_n . . . . .	800
10.155.4.10 d_iode . . . . .	800
10.155.4.11 d_l3rd_n . . . . .	801
10.155.4.12 d_l5th_n . . . . .	801
10.155.4.13 d_M . . . . .	801
10.155.4.14 d_m . . . . .	801
10.155.4.15 d_n . . . . .	801
10.155.4.16 d_N_T . . . . .	801
10.155.4.17 d_P . . . . .	801
10.155.4.18 d_P_1 . . . . .	801
10.155.4.19 d_P_2 . . . . .	802
10.155.4.20 d_P_3 . . . . .	802
10.155.4.21 d_P_4 . . . . .	802
10.155.4.22 d_satClkDrift . . . . .	802
10.155.4.23 d_t_b . . . . .	802
10.155.4.24 d_t_k . . . . .	802
10.155.4.25 d_tau_c . . . . .	802
10.155.4.26 d_tau_n . . . . .	802
10.155.4.27 d_tod . . . . .	803
10.155.4.28 d_TOW . . . . .	803
10.155.4.29 d_VXn . . . . .	803
10.155.4.30 d_VYn . . . . .	803
10.155.4.31 d_VZn . . . . .	803
10.155.4.32 d_WN . . . . .	803
10.155.4.33 d_Xn . . . . .	803
10.155.4.34 d_Yn . . . . .	803
10.155.4.35 d_yr . . . . .	803
10.155.4.36 d_Zn . . . . .	804
10.155.4.37 i_satellite_freq_channel . . . . .	804
10.155.4.38 i_satellite_slot_number . . . . .	804
10.155.4.39 PRN . . . . .	804
10.156 Glonass_Gnav_Navigation_Message Class Reference . . . . .	804
10.156.1 Detailed Description . . . . .	806

10.156.2 Constructor & Destructor Documentation	806
10.156.2.1 Glonass_Gnav_Navigation_Message()	806
10.156.3 Member Function Documentation	806
10.156.3.1 CRC_test()	806
10.156.3.2 get_alm_satellite_slot_number()	806
10.156.3.3 get_almanac()	806
10.156.3.4 get_ephemeris()	807
10.156.3.5 get_flag_CRC_test()	807
10.156.3.6 get_flag_TOW_new()	807
10.156.3.7 get_flag_update_slot_number()	807
10.156.3.8 get_frame_number()	807
10.156.3.9 get_utc_model()	807
10.156.3.10 have_new_almanac()	807
10.156.3.11 have_new_ephemeris()	807
10.156.3.12 have_new_utc_model()	807
10.156.3.13 is_flag_TOW_set()	807
10.156.3.14 set_flag_ephemeris_str_1()	808
10.156.3.15 set_flag_ephemeris_str_2()	808
10.156.3.16 set_flag_ephemeris_str_3()	808
10.156.3.17 set_flag_ephemeris_str_4()	808
10.156.3.18 set_flag_TOW_new()	808
10.156.3.19 set_flag_update_slot_number()	808
10.156.3.20 set_rf_link()	808
10.156.3.21 string_decoder()	808
10.157 Glonass_Gnav_Utc_Model Class Reference	809
10.157.1 Detailed Description	810
10.157.2 Constructor & Destructor Documentation	810
10.157.2.1 Glonass_Gnav_Utc_Model()	810
10.157.3 Member Function Documentation	810
10.157.3.1 serialize()	810
10.157.3.2 utc_time()	810
10.157.4 Member Data Documentation	810
10.157.4.1 d_B1	810
10.157.4.2 d_B2	810
10.157.4.3 d_N_4	810
10.157.4.4 d_N_A	811
10.157.4.5 d_tau_c	811
10.157.4.6 d_tau_gps	811
10.157.4.7 valid	811
10.158 glonass_l1_ca_dll_pll_c_aid_tracking_cc Class Reference	811
10.158.1 Detailed Description	813
10.159 glonass_l1_ca_dll_pll_c_aid_tracking_sc Class Reference	813

10.159.1 Detailed Description . . . . .	814
10.160 Glonass_L1_Ca_DLL_PII_Tracking_cc Class Reference . . . . .	814
10.160.1 Detailed Description . . . . .	816
10.161 glonass_l1_ca_telemetry_decoder_gs Class Reference . . . . .	816
10.161.1 Detailed Description . . . . .	817
10.161.2 Constructor & Destructor Documentation . . . . .	818
10.161.2.1 ~glonass_l1_ca_telemetry_decoder_gs() . . . . .	818
10.161.3 Member Function Documentation . . . . .	818
10.161.3.1 general_work() . . . . .	818
10.161.3.2 reset() . . . . .	818
10.161.3.3 set_channel() . . . . .	818
10.161.3.4 set_satellite() . . . . .	818
10.162 glonass_l2_ca_dll_pll_c_aid_tracking_cc Class Reference . . . . .	818
10.162.1 Detailed Description . . . . .	820
10.163 glonass_l2_ca_dll_pll_c_aid_tracking_sc Class Reference . . . . .	820
10.163.1 Detailed Description . . . . .	821
10.164 Glonass_L2_Ca_DLL_PII_Tracking_cc Class Reference . . . . .	821
10.164.1 Detailed Description . . . . .	823
10.165 glonass_l2_ca_telemetry_decoder_gs Class Reference . . . . .	823
10.165.1 Detailed Description . . . . .	824
10.165.2 Constructor & Destructor Documentation . . . . .	825
10.165.2.1 ~glonass_l2_ca_telemetry_decoder_gs() . . . . .	825
10.165.3 Member Function Documentation . . . . .	825
10.165.3.1 general_work() . . . . .	825
10.165.3.2 reset() . . . . .	825
10.165.3.3 set_channel() . . . . .	825
10.165.3.4 set_satellite() . . . . .	825
10.166 GlonassL1CaDIIPICAidTracking Class Reference . . . . .	825
10.166.1 Detailed Description . . . . .	828
10.166.2 Member Function Documentation . . . . .	828
10.166.2.1 connect() . . . . .	828
10.166.2.2 disconnect() . . . . .	828
10.166.2.3 get_left_block() . . . . .	828
10.166.2.4 get_right_block() . . . . .	828
10.166.2.5 implementation() . . . . .	828
10.166.2.6 item_size() . . . . .	829
10.166.2.7 role() . . . . .	829
10.166.2.8 set_channel() . . . . .	829
10.166.2.9 set_gnss_synchro() . . . . .	829
10.166.2.10 start_tracking() . . . . .	829
10.166.2.11 stop_tracking() . . . . .	829
10.167 GlonassL1CaDIPIITracking Class Reference . . . . .	829

10.167.1 Detailed Description . . . . .	832
10.167.2 Member Function Documentation . . . . .	832
10.167.2.1 connect() . . . . .	832
10.167.2.2 disconnect() . . . . .	832
10.167.2.3 get_left_block() . . . . .	832
10.167.2.4 get_right_block() . . . . .	832
10.167.2.5 implementation() . . . . .	832
10.167.2.6 item_size() . . . . .	833
10.167.2.7 role() . . . . .	833
10.167.2.8 set_channel() . . . . .	833
10.167.2.9 set_gnss_synchro() . . . . .	833
10.167.2.10 start_tracking() . . . . .	833
10.167.2.11 stop_tracking() . . . . .	833
10.168 GlonassL1CaPcpsAcquisition Class Reference . . . . .	833
10.168.1 Detailed Description . . . . .	836
10.168.2 Member Function Documentation . . . . .	836
10.168.2.1 connect() . . . . .	836
10.168.2.2 disconnect() . . . . .	837
10.168.2.3 get_left_block() . . . . .	837
10.168.2.4 get_right_block() . . . . .	837
10.168.2.5 implementation() . . . . .	837
10.168.2.6 init() . . . . .	837
10.168.2.7 item_size() . . . . .	837
10.168.2.8 mag() . . . . .	837
10.168.2.9 reset() . . . . .	837
10.168.2.10 role() . . . . .	837
10.168.2.11 set_channel() . . . . .	837
10.168.2.12 set_channel_fsm() . . . . .	838
10.168.2.13 set_doppler_max() . . . . .	838
10.168.2.14 set_doppler_step() . . . . .	838
10.168.2.15 set_gnss_synchro() . . . . .	838
10.168.2.16 set_local_code() . . . . .	838
10.168.2.17 set_resampler_latency() . . . . .	838
10.168.2.18 set_state() . . . . .	838
10.168.2.19 set_threshold() . . . . .	838
10.168.2.20 stop_acquisition() . . . . .	838
10.169 GlonassL1CaTelemetryDecoder Class Reference . . . . .	839
10.169.1 Detailed Description . . . . .	841
10.169.2 Member Function Documentation . . . . .	841
10.169.2.1 connect() . . . . .	841
10.169.2.2 disconnect() . . . . .	841
10.169.2.3 get_left_block() . . . . .	841

10.169.2.4 get_right_block()	841
10.169.2.5 implementation()	841
10.169.2.6 item_size()	841
10.169.2.7 reset()	842
10.169.2.8 role()	842
10.169.2.9 set_channel()	842
10.169.2.10 set_satellite()	842
10.170 GlonassL2CaDIIPICAidTracking Class Reference	842
10.170.1 Detailed Description	845
10.170.2 Member Function Documentation	845
10.170.2.1 connect()	845
10.170.2.2 disconnect()	845
10.170.2.3 get_left_block()	845
10.170.2.4 get_right_block()	845
10.170.2.5 implementation()	845
10.170.2.6 item_size()	846
10.170.2.7 role()	846
10.170.2.8 set_channel()	846
10.170.2.9 set_gnss_synchro()	846
10.170.2.10 start_tracking()	846
10.170.2.11 stop_tracking()	846
10.171 GlonassL2CaDIIPITracking Class Reference	846
10.171.1 Detailed Description	849
10.171.2 Member Function Documentation	849
10.171.2.1 connect()	849
10.171.2.2 disconnect()	849
10.171.2.3 get_left_block()	849
10.171.2.4 get_right_block()	849
10.171.2.5 implementation()	849
10.171.2.6 item_size()	850
10.171.2.7 role()	850
10.171.2.8 set_channel()	850
10.171.2.9 set_gnss_synchro()	850
10.171.2.10 start_tracking()	850
10.171.2.11 stop_tracking()	850
10.172 GlonassL2CaPcpsAcquisition Class Reference	850
10.172.1 Detailed Description	853
10.172.2 Member Function Documentation	853
10.172.2.1 connect()	853
10.172.2.2 disconnect()	854
10.172.2.3 get_left_block()	854
10.172.2.4 get_right_block()	854

10.172.2.5 implementation()	854
10.172.2.6 init()	854
10.172.2.7 item_size()	854
10.172.2.8 mag()	854
10.172.2.9 reset()	854
10.172.2.10 role()	854
10.172.2.11 set_channel()	854
10.172.2.12 set_channel_fsm()	855
10.172.2.13 set_doppler_max()	855
10.172.2.14 set_doppler_step()	855
10.172.2.15 set_gnss_synchro()	855
10.172.2.16 set_local_code()	855
10.172.2.17 set_resampler_latency()	855
10.172.2.18 set_state()	855
10.172.2.19 set_threshold()	855
10.172.2.20 stop_acquisition()	855
10.173 GlonassL2CaTelemetryDecoder Class Reference	856
10.173.1 Detailed Description	858
10.173.2 Member Function Documentation	858
10.173.2.1 connect()	858
10.173.2.2 disconnect()	858
10.173.2.3 get_left_block()	858
10.173.2.4 get_right_block()	858
10.173.2.5 implementation()	858
10.173.2.6 item_size()	858
10.173.2.7 reset()	859
10.173.2.8 role()	859
10.173.2.9 set_channel()	859
10.173.2.10 set_satellite()	859
10.174 Gnss_Almanac Class Reference	859
10.174.1 Detailed Description	862
10.174.2 Constructor & Destructor Documentation	862
10.174.2.1 Gnss_Almanac()	862
10.174.3 Member Function Documentation	862
10.174.3.1 predicted_doppler()	862
10.174.3.2 satellitePosVelComputation()	863
10.174.4 Member Data Documentation	863
10.174.4.1 af0	863
10.174.4.2 af1	863
10.174.4.3 delta_i	863
10.174.4.4 ecc	863
10.174.4.5 M_0	863



10.174.4.6 omega	863
10.174.4.7 OMEGA_0	864
10.174.4.8 OMEGAdot	864
10.174.4.9 PRN	864
10.174.4.10 sqrtA	864
10.174.4.11 System	864
10.174.4.12 toa	864
10.174.4.13 WNa	864
10.175 Gnss_circular_deque< T > Class Template Reference	865
10.175.1 Detailed Description	867
10.176 Gnss_Crypto Class Reference	867
10.176.1 Detailed Description	868
10.176.2 Constructor & Destructor Documentation	868
10.176.2.1 Gnss_Crypto() [1/2]	868
10.176.2.2 Gnss_Crypto() [2/2]	868
10.176.2.3 ~Gnss_Crypto()	868
10.176.3 Member Function Documentation	868
10.176.3.1 compute_CMAC_AES()	868
10.176.3.2 compute_HMAC_SHA_256()	869
10.176.3.3 compute_SHA3_256()	869
10.176.3.4 compute_SHA_256()	869
10.176.3.5 get_merkle_root()	869
10.176.3.6 get_public_key_type()	869
10.176.3.7 have_public_key()	869
10.176.3.8 set_merkle_root()	869
10.176.3.9 set_public_key()	869
10.176.3.10 set_public_key_type()	869
10.176.3.11 store_public_key()	869
10.176.3.12 verify_signature_ecdsa_p256()	870
10.176.3.13 verify_signature_ecdsa_p521()	870
10.177 Gnss_Ephemeris Class Reference	870
10.177.1 Detailed Description	873
10.177.2 Member Function Documentation	873
10.177.2.1 predicted_doppler()	873
10.177.2.2 satellitePosition()	873
10.177.2.3 sv_clock_drift()	874
10.177.3 Member Data Documentation	874
10.177.3.1 af0	874
10.177.3.2 af1	874
10.177.3.3 af2	874
10.177.3.4 Cic	874
10.177.3.5 Cis	874

10.177.3.6 Crc	874
10.177.3.7 Crs	875
10.177.3.8 Cuc	875
10.177.3.9 Cus	875
10.177.3.10 delta_n	875
10.177.3.11 dtr	875
10.177.3.12 ecc	875
10.177.3.13 i_0	875
10.177.3.14 idot	875
10.177.3.15 M_0	876
10.177.3.16 omega	876
10.177.3.17 OMEGA_0	876
10.177.3.18 OMEGAdot	876
10.177.3.19 PRN	876
10.177.3.20 satClkDrift	876
10.177.3.21 satpos_X	876
10.177.3.22 satpos_Y	877
10.177.3.23 satpos_Z	877
10.177.3.24 satvel_X	877
10.177.3.25 satvel_Y	877
10.177.3.26 satvel_Z	877
10.177.3.27 sqrtA	877
10.177.3.28 System	877
10.177.3.29 toc	877
10.177.3.30 toe	877
10.177.3.31 tow	878
10.177.3.32 WN	878
10.178 Gnss_Satellite Class Reference	878
10.178.1 Detailed Description	879
10.178.2 Constructor & Destructor Documentation	879
10.178.2.1 Gnss_Satellite() [1/4]	879
10.178.2.2 Gnss_Satellite() [2/4]	880
10.178.2.3 ~Gnss_Satellite()	880
10.178.2.4 Gnss_Satellite() [3/4]	880
10.178.2.5 Gnss_Satellite() [4/4]	880
10.178.3 Member Function Documentation	880
10.178.3.1 get_block()	880
10.178.3.2 get_PRN()	881
10.178.3.3 get_rf_link()	881
10.178.3.4 get_system()	881
10.178.3.5 get_system_short()	881
10.178.3.6 operator=() [1/2]	881

10.178.3.7 operator=() [2/2] . . . . .	881
10.178.3.8 update_PRN() . . . . .	882
10.178.3.9 what_block() . . . . .	882
10.178.4 Friends And Related Symbol Documentation . . . . .	882
10.178.4.1 operator<< . . . . .	882
10.178.4.2 operator== . . . . .	882
10.179 gnss_sdr_fpga_sample_counter Class Reference . . . . .	882
10.179.1 Detailed Description . . . . .	883
10.180 gnss_sdr_sample_counter Class Reference . . . . .	884
10.180.1 Detailed Description . . . . .	885
10.181 Gnss_Sdr_Supl_Client Class Reference . . . . .	885
10.181.1 Detailed Description . . . . .	887
10.181.2 Member Function Documentation . . . . .	887
10.181.2.1 load_cnav_ephemeris_xml() . . . . .	887
10.181.2.2 load_cnav_utc_xml() . . . . .	887
10.181.2.3 load_ephemeris_xml() . . . . .	887
10.181.2.4 load_gal_almanac_xml() . . . . .	887
10.181.2.5 load_gal_ephemeris_xml() . . . . .	887
10.181.2.6 load_gal_iono_xml() . . . . .	888
10.181.2.7 load_gal_utc_xml() . . . . .	888
10.181.2.8 load_glo_utc_xml() . . . . .	888
10.181.2.9 load_gnav_ephemeris_xml() . . . . .	888
10.181.2.10 load_gps_almanac_xml() . . . . .	888
10.181.2.11 load_iono_xml() . . . . .	888
10.181.2.12 load_ref_location_xml() . . . . .	888
10.181.2.13 load_ref_time_xml() . . . . .	888
10.181.2.14 load_utc_xml() . . . . .	888
10.181.2.15 save_cnav_ephemeris_map_xml() . . . . .	888
10.181.2.16 save_cnav_utc_xml() . . . . .	889
10.181.2.17 save_ephemeris_map_xml() . . . . .	889
10.181.2.18 save_gal_almanac_xml() . . . . .	889
10.181.2.19 save_gal_ephemeris_map_xml() . . . . .	889
10.181.2.20 save_gal_iono_xml() . . . . .	889
10.181.2.21 save_gal_utc_xml() . . . . .	889
10.181.2.22 save_glo_utc_xml() . . . . .	889
10.181.2.23 save_gnav_ephemeris_map_xml() . . . . .	889
10.181.2.24 save_gps_almanac_xml() . . . . .	890
10.181.2.25 save_iono_xml() . . . . .	890
10.181.2.26 save_ref_location_xml() . . . . .	890
10.181.2.27 save_ref_time_xml() . . . . .	890
10.181.2.28 save_utc_xml() . . . . .	890
10.181.3 Member Data Documentation . . . . .	890

10.181.3.1 gal_almanac_map . . . . .	890
10.181.3.2 gal_ephemeris_map . . . . .	890
10.181.3.3 gal_iono . . . . .	890
10.181.3.4 gal_utc . . . . .	890
10.181.3.5 glo_gnav_utc . . . . .	890
10.181.3.6 glonass_gnav_ephemeris_map . . . . .	891
10.181.3.7 gps_acq_map . . . . .	891
10.181.3.8 gps_almanac_map . . . . .	891
10.181.3.9 gps_cnav_ephemeris_map . . . . .	891
10.181.3.10 gps_cnav_utc . . . . .	891
10.181.3.11 gps_ephemeris_map . . . . .	891
10.181.3.12 gps_iono . . . . .	891
10.181.3.13 gps_ref_loc . . . . .	891
10.181.3.14 gps_time . . . . .	891
10.181.3.15 gps_utc . . . . .	891
10.181.3.16 request . . . . .	891
10.181.3.17 server_name . . . . .	891
10.181.3.18 server_port . . . . .	892
10.182 gnss_sdr_time_counter Class Reference . . . . .	892
10.182.1 Detailed Description . . . . .	893
10.183 Gnss_Sdr_Timestamp Class Reference . . . . .	893
10.183.1 Detailed Description . . . . .	894
10.184 Gnss_Sdr_Valve Class Reference . . . . .	894
10.184.1 Detailed Description . . . . .	896
10.185 Gnss_Signal Class Reference . . . . .	896
10.185.1 Detailed Description . . . . .	896
10.185.2 Member Function Documentation . . . . .	897
10.185.2.1 get_satellite() . . . . .	897
10.185.2.2 get_signal_str() . . . . .	897
10.185.3 Friends And Related Symbol Documentation . . . . .	897
10.185.3.1 operator<< . . . . .	897
10.185.3.2 operator== . . . . .	897
10.186 Gnss_Synchro Class Reference . . . . .	897
10.186.1 Detailed Description . . . . .	900
10.186.2 Constructor & Destructor Documentation . . . . .	900
10.186.2.1 Gnss_Synchro() [1/3] . . . . .	900
10.186.2.2 ~Gnss_Synchro() . . . . .	900
10.186.2.3 Gnss_Synchro() [2/3] . . . . .	900
10.186.2.4 Gnss_Synchro() [3/3] . . . . .	900
10.186.3 Member Function Documentation . . . . .	901
10.186.3.1 operator=() [1/2] . . . . .	901
10.186.3.2 operator=() [2/2] . . . . .	901

10.186.3.3 serialize()	901
10.186.4 Member Data Documentation	902
10.186.4.1 Acq_delay_samples	902
10.186.4.2 Acq_doppler_hz	902
10.186.4.3 Acq_doppler_step	902
10.186.4.4 Acq_samplestamp_samples	902
10.186.4.5 Carrier_Doppler_hz	902
10.186.4.6 Carrier_phase_rads	902
10.186.4.7 Channel_ID	902
10.186.4.8 CN0_dB_hz	903
10.186.4.9 Code_phase_samples	903
10.186.4.10 correlation_length_ms	903
10.186.4.11 Flag_PLL_180_deg_phase_locked	903
10.186.4.12 Flag_valid_acquisition	903
10.186.4.13 Flag_valid_pseudorange	903
10.186.4.14 Flag_valid_symbol_output	903
10.186.4.15 Flag_valid_word	903
10.186.4.16 fs	903
10.186.4.17 interp_TOW_ms	904
10.186.4.18 PRN	904
10.186.4.19 Prompt_I	904
10.186.4.20 Prompt_Q	904
10.186.4.21 Pseudorange_m	904
10.186.4.22 RX_time	904
10.186.4.23 Signal	904
10.186.4.24 System	904
10.186.4.25 TOW_at_current_symbol_ms	905
10.186.4.26 Tracking_sample_counter	905
10.187 gnss_synchro_monitor Class Reference	905
10.187.1 Detailed Description	906
10.187.2 Constructor & Destructor Documentation	906
10.187.2.1 ~gnss_synchro_monitor()	906
10.188 Gnss_Synchro_Udp_Sink Class Reference	906
10.188.1 Detailed Description	907
10.189 GNSSBlockFactory Class Reference	907
10.189.1 Detailed Description	908
10.189.2 Member Function Documentation	908
10.189.2.1 GetBlock()	908
10.190 GNSSBlockInterface Class Reference	908
10.190.1 Detailed Description	909
10.190.2 Member Function Documentation	909
10.190.2.1 connect()	909

10.190.2.2 disconnect()	910
10.190.2.3 get_left_block() [1/2]	910
10.190.2.4 get_left_block() [2/2]	910
10.190.2.5 get_right_block() [1/2]	910
10.190.2.6 get_right_block() [2/2]	910
10.190.2.7 implementation()	910
10.190.2.8 item_size()	911
10.190.2.9 role()	911
10.190.2.10 start()	911
10.191 GNSSFlowgraph Class Reference	911
10.191.1 Detailed Description	913
10.191.2 Constructor & Destructor Documentation	913
10.191.2.1 GNSSFlowgraph()	913
10.191.2.2 ~GNSSFlowgraph()	913
10.191.3 Member Function Documentation	913
10.191.3.1 acquisition_manager()	913
10.191.3.2 apply_action()	913
10.191.3.3 connect()	914
10.191.3.4 connected()	914
10.191.3.5 disconnect()	914
10.191.3.6 get_pvt()	914
10.191.3.7 prioritize_satellites()	914
10.191.3.8 running()	914
10.191.3.9 send_telemetry_msg()	914
10.191.3.10 set_configuration()	914
10.191.3.11 start()	914
10.191.3.12 stop()	914
10.191.3.13 wait()	915
10.192 GnssTime Class Reference	915
10.192.1 Detailed Description	915
10.192.2 Member Data Documentation	915
10.192.2.1 rx_time	915
10.192.2.2 tow_ms	915
10.192.2.3 tow_ms_fraction	915
10.192.2.4 week	916
10.193 Gps_Acq_Assist Class Reference	916
10.193.1 Detailed Description	917
10.193.2 Constructor & Destructor Documentation	917
10.193.2.1 Gps_Acq_Assist()	917
10.193.3 Member Data Documentation	917
10.193.3.1 Azimuth	917
10.193.3.2 Code_Phase	917

10.193.3.3 Code_Phase_int . . . . .	917
10.193.3.4 Code_Phase_window . . . . .	917
10.193.3.5 Doppler0 . . . . .	917
10.193.3.6 Doppler1 . . . . .	918
10.193.3.7 dopplerUncertainty . . . . .	918
10.193.3.8 Elevation . . . . .	918
10.193.3.9 GPS_Bit_Number . . . . .	918
10.193.3.10 PRN . . . . .	918
10.193.3.11 tow . . . . .	918
10.194 Gps_Almanac Class Reference . . . . .	918
10.194.1 Detailed Description . . . . .	921
10.194.2 Constructor & Destructor Documentation . . . . .	921
10.194.2.1 Gps_Almanac() . . . . .	921
10.194.3 Member Function Documentation . . . . .	922
10.194.3.1 serialize() . . . . .	922
10.194.4 Member Data Documentation . . . . .	922
10.194.4.1 AS_status . . . . .	922
10.194.4.2 SV_health . . . . .	922
10.195 Gps_CNAV_Ephemeris Class Reference . . . . .	922
10.195.1 Detailed Description . . . . .	927
10.195.2 Constructor & Destructor Documentation . . . . .	927
10.195.2.1 Gps_CNAV_Ephemeris() . . . . .	927
10.195.3 Member Function Documentation . . . . .	927
10.195.3.1 serialize() . . . . .	927
10.195.4 Member Data Documentation . . . . .	928
10.195.4.1 Adot . . . . .	928
10.195.4.2 alert_flag . . . . .	928
10.195.4.3 antispoofing_flag . . . . .	928
10.195.4.4 delta_A . . . . .	928
10.195.4.5 delta_ndot . . . . .	928
10.195.4.6 delta_OMEGAAdot . . . . .	928
10.195.4.7 integrity_status_flag . . . . .	928
10.195.4.8 ISCL1 . . . . .	928
10.195.4.9 ISCL2 . . . . .	929
10.195.4.10 ISCL5I . . . . .	929
10.195.4.11 ISCL5Q . . . . .	929
10.195.4.12 l2c_phasing_flag . . . . .	929
10.195.4.13 signal_health . . . . .	929
10.195.4.14 TGD . . . . .	929
10.195.4.15 toe1 . . . . .	929
10.195.4.16 toe2 . . . . .	929
10.195.4.17 top . . . . .	929

10.195.4.18 URA . . . . .	929
10.195.4.19 URA0 . . . . .	930
10.195.4.20 URA1 . . . . .	930
10.195.4.21 URA2 . . . . .	930
10.196 Gps_CNAV_Iono Class Reference . . . . .	930
10.196.1 Detailed Description . . . . .	932
10.196.2 Constructor & Destructor Documentation . . . . .	932
10.196.2.1 Gps_CNAV_Iono() . . . . .	932
10.197 Gps_CNAV_Navigation_Message Class Reference . . . . .	932
10.197.1 Detailed Description . . . . .	933
10.197.2 Constructor & Destructor Documentation . . . . .	933
10.197.2.1 Gps_CNAV_Navigation_Message() . . . . .	933
10.197.3 Member Function Documentation . . . . .	933
10.197.3.1 get_ephemeris() . . . . .	933
10.197.3.2 get_iono() . . . . .	933
10.197.3.3 get_utc_model() . . . . .	933
10.197.3.4 have_new_ephemeris() . . . . .	933
10.197.3.5 have_new_iono() . . . . .	933
10.197.3.6 have_new_utc_model() . . . . .	934
10.198 Gps_CNAV_Utc_Model Class Reference . . . . .	934
10.198.1 Detailed Description . . . . .	936
10.198.2 Constructor & Destructor Documentation . . . . .	936
10.198.2.1 Gps_CNAV_Utc_Model() . . . . .	936
10.199 Gps_Ephemeris Class Reference . . . . .	936
10.199.1 Detailed Description . . . . .	941
10.199.2 Constructor & Destructor Documentation . . . . .	941
10.199.2.1 Gps_Ephemeris() . . . . .	941
10.199.3 Member Function Documentation . . . . .	941
10.199.3.1 serialize() . . . . .	941
10.199.4 Member Data Documentation . . . . .	942
10.199.4.1 alert_flag . . . . .	942
10.199.4.2 antispoofing_flag . . . . .	942
10.199.4.3 AODO . . . . .	942
10.199.4.4 code_on_L2 . . . . .	942
10.199.4.5 fit_interval_flag . . . . .	942
10.199.4.6 integrity_status_flag . . . . .	942
10.199.4.7 IODC . . . . .	942
10.199.4.8 IODE_SF2 . . . . .	943
10.199.4.9 IODE_SF3 . . . . .	943
10.199.4.10 L2_P_data_flag . . . . .	943
10.199.4.11 satelliteBlock . . . . .	943
10.199.4.12 spare1 . . . . .	943



10.199.4.13 spare2 . . . . .	943
10.199.4.14 SV_accuracy . . . . .	943
10.199.4.15 SV_health . . . . .	943
10.199.4.16 TGD . . . . .	943
10.200 Gps_Iono Class Reference . . . . .	944
10.200.1 Detailed Description . . . . .	946
10.200.2 Constructor & Destructor Documentation . . . . .	946
10.200.2.1 Gps_Iono() . . . . .	946
10.200.3 Member Function Documentation . . . . .	946
10.200.3.1 serialize() . . . . .	946
10.200.4 Member Data Documentation . . . . .	946
10.200.4.1 alpha0 . . . . .	946
10.200.4.2 alpha1 . . . . .	946
10.200.4.3 alpha2 . . . . .	946
10.200.4.4 alpha3 . . . . .	946
10.200.4.5 beta0 . . . . .	946
10.200.4.6 beta1 . . . . .	947
10.200.4.7 beta2 . . . . .	947
10.200.4.8 beta3 . . . . .	947
10.200.4.9 valid . . . . .	947
10.201 Gps_L1_Ca_Dll_Pll_Tracking_GPU_cc Class Reference . . . . .	947
10.201.1 Detailed Description . . . . .	949
10.202 Gps_L1_Ca_Gaussian_Tracking_cc Class Reference . . . . .	949
10.202.1 Detailed Description . . . . .	950
10.203 Gps_L1_Ca_Tcp_Connector_Tracking_cc Class Reference . . . . .	950
10.203.1 Detailed Description . . . . .	952
10.204 gps_l1_ca_telemetry_decoder_gs Class Reference . . . . .	952
10.204.1 Detailed Description . . . . .	953
10.204.2 Member Function Documentation . . . . .	953
10.204.2.1 general_work() . . . . .	953
10.204.2.2 set_channel() . . . . .	954
10.204.2.3 set_satellite() . . . . .	954
10.205 gps_l2c_telemetry_decoder_gs Class Reference . . . . .	954
10.205.1 Detailed Description . . . . .	955
10.205.2 Member Function Documentation . . . . .	955
10.205.2.1 general_work() . . . . .	955
10.205.2.2 set_channel() . . . . .	956
10.205.2.3 set_satellite() . . . . .	956
10.206 gps_l5_telemetry_decoder_gs Class Reference . . . . .	956
10.206.1 Detailed Description . . . . .	957
10.206.2 Member Function Documentation . . . . .	957
10.206.2.1 set_channel() . . . . .	957

10.206.2.2 set_satellite()	957
10.207 Gps_Navigation_Message Class Reference	958
10.207.1 Detailed Description	959
10.207.2 Constructor & Destructor Documentation	959
10.207.2.1 Gps_Navigation_Message()	959
10.207.3 Member Function Documentation	959
10.207.3.1 get_almanac()	959
10.207.3.2 get_ephemeris()	959
10.207.3.3 get_flag_iono_valid()	959
10.207.3.4 get_flag_utc_model_valid()	959
10.207.3.5 get_GPS_week()	959
10.207.3.6 get_iono()	960
10.207.3.7 get_satellite_PRN()	960
10.207.3.8 get_TOW()	960
10.207.3.9 get_utc_model()	960
10.207.3.10 set_channel()	960
10.207.3.11 set_satellite_PRN()	960
10.207.3.12 subframe_decoder()	960
10.207.3.13 utc_time()	960
10.208 Gps_Utc_Model Class Reference	960
10.208.1 Detailed Description	963
10.208.2 Constructor & Destructor Documentation	963
10.208.2.1 Gps_Utc_Model()	963
10.208.3 Member Function Documentation	963
10.208.3.1 serialize()	963
10.208.4 Member Data Documentation	963
10.208.4.1 A0	963
10.208.4.2 A1	963
10.208.4.3 A2	963
10.208.4.4 DeltaT_LS	963
10.208.4.5 DeltaT_LSF	963
10.208.4.6 DN	963
10.208.4.7 tot	964
10.208.4.8 valid	964
10.208.4.9 WN_LSF	964
10.208.4.10 WN_T	964
10.209 GpsL1CaDIIPIITracking Class Reference	964
10.209.1 Detailed Description	967
10.209.2 Member Function Documentation	967
10.209.2.1 connect()	967
10.209.2.2 disconnect()	967
10.209.2.3 get_left_block()	967

10.209.2.4	<a href="#">get_right_block()</a>	967
10.209.2.5	<a href="#">implementation()</a>	967
10.209.2.6	<a href="#">item_size()</a>	968
10.209.2.7	<a href="#">role()</a>	968
10.209.2.8	<a href="#">set_channel()</a>	968
10.209.2.9	<a href="#">set_gnss_synchro()</a>	968
10.209.2.10	<a href="#">start_tracking()</a>	968
10.209.2.11	<a href="#">stop_tracking()</a>	968
10.210	<a href="#">GpsL1CaDIIPIITrackingFpga Class Reference</a>	968
10.210.1	<a href="#">Detailed Description</a>	971
10.210.2	<a href="#">Constructor &amp; Destructor Documentation</a>	971
10.210.2.1	<a href="#">GpsL1CaDIIPIITrackingFpga()</a>	971
10.210.2.2	<a href="#">~GpsL1CaDIIPIITrackingFpga()</a>	972
10.210.3	<a href="#">Member Function Documentation</a>	972
10.210.3.1	<a href="#">connect()</a>	972
10.210.3.2	<a href="#">disconnect()</a>	972
10.210.3.3	<a href="#">get_left_block()</a>	972
10.210.3.4	<a href="#">get_right_block()</a>	972
10.210.3.5	<a href="#">implementation()</a>	972
10.210.3.6	<a href="#">item_size()</a>	972
10.210.3.7	<a href="#">role()</a>	972
10.210.3.8	<a href="#">set_channel()</a>	972
10.210.3.9	<a href="#">set_gnss_synchro()</a>	973
10.210.3.10	<a href="#">start_tracking()</a>	973
10.210.3.11	<a href="#">stop_tracking()</a>	973
10.211	<a href="#">GpsL1CaDIIPIITrackingGPU Class Reference</a>	973
10.211.1	<a href="#">Detailed Description</a>	976
10.211.2	<a href="#">Member Function Documentation</a>	976
10.211.2.1	<a href="#">connect()</a>	976
10.211.2.2	<a href="#">disconnect()</a>	976
10.211.2.3	<a href="#">get_left_block()</a>	976
10.211.2.4	<a href="#">get_right_block()</a>	976
10.211.2.5	<a href="#">implementation()</a>	976
10.211.2.6	<a href="#">item_size()</a>	977
10.211.2.7	<a href="#">role()</a>	977
10.211.2.8	<a href="#">set_channel()</a>	977
10.211.2.9	<a href="#">set_gnss_synchro()</a>	977
10.211.2.10	<a href="#">start_tracking()</a>	977
10.211.2.11	<a href="#">stop_tracking()</a>	977
10.212	<a href="#">GpsL1CaGaussianTracking Class Reference</a>	977
10.212.1	<a href="#">Detailed Description</a>	980
10.212.2	<a href="#">Member Function Documentation</a>	980

10.212.2.1 connect()	980
10.212.2.2 disconnect()	980
10.212.2.3 get_left_block()	980
10.212.2.4 get_right_block()	980
10.212.2.5 implementation()	980
10.212.2.6 item_size()	981
10.212.2.7 role()	981
10.212.2.8 set_channel()	981
10.212.2.9 set_gnss_synchro()	981
10.212.2.10 start_tracking()	981
10.212.2.11 stop_tracking()	981
10.213 GpsL1CaKfTracking Class Reference	981
10.213.1 Detailed Description	984
10.213.2 Member Function Documentation	984
10.213.2.1 connect()	984
10.213.2.2 disconnect()	984
10.213.2.3 get_left_block()	984
10.213.2.4 get_right_block()	984
10.213.2.5 implementation()	984
10.213.2.6 item_size()	985
10.213.2.7 role()	985
10.213.2.8 set_channel()	985
10.213.2.9 set_gnss_synchro()	985
10.213.2.10 start_tracking()	985
10.213.2.11 stop_tracking()	985
10.214 GpsL1CaPcpsAcquisition Class Reference	985
10.214.1 Detailed Description	988
10.214.2 Member Function Documentation	989
10.214.2.1 connect()	989
10.214.2.2 disconnect()	989
10.214.2.3 get_left_block()	989
10.214.2.4 get_right_block()	989
10.214.2.5 implementation()	989
10.214.2.6 init()	989
10.214.2.7 item_size()	989
10.214.2.8 mag()	989
10.214.2.9 reset()	989
10.214.2.10 role()	989
10.214.2.11 set_channel()	990
10.214.2.12 set_channel_fsm()	990
10.214.2.13 set_doppler_center()	990
10.214.2.14 set_doppler_max()	990

10.214.2.15 set_doppler_step()	990
10.214.2.16 set_gnss_synchro()	990
10.214.2.17 set_local_code()	990
10.214.2.18 set_resampler_latency()	990
10.214.2.19 set_state()	991
10.214.2.20 set_threshold()	991
10.214.2.21 stop_acquisition()	991
10.215 GpsL1CaPcpsAcquisitionFineDoppler Class Reference	991
10.215.1 Detailed Description	994
10.215.2 Member Function Documentation	995
10.215.2.1 get_left_block()	995
10.215.2.2 get_right_block()	995
10.215.2.3 implementation()	995
10.215.2.4 init()	995
10.215.2.5 item_size()	995
10.215.2.6 mag()	995
10.215.2.7 reset()	995
10.215.2.8 role()	995
10.215.2.9 set_channel()	995
10.215.2.10 set_channel_fsm()	996
10.215.2.11 set_doppler_max()	996
10.215.2.12 set_doppler_step()	996
10.215.2.13 set_gnss_synchro()	996
10.215.2.14 set_local_code()	996
10.215.2.15 set_resampler_latency()	996
10.215.2.16 set_state()	996
10.215.2.17 set_threshold()	996
10.215.2.18 stop_acquisition()	996
10.216 GpsL1CaPcpsAcquisitionFpga Class Reference	997
10.216.1 Detailed Description	1000
10.216.2 Constructor & Destructor Documentation	1000
10.216.2.1 GpsL1CaPcpsAcquisitionFpga()	1000
10.216.2.2 ~GpsL1CaPcpsAcquisitionFpga()	1000
10.216.3 Member Function Documentation	1000
10.216.3.1 connect()	1000
10.216.3.2 disconnect()	1000
10.216.3.3 get_left_block()	1000
10.216.3.4 get_right_block()	1001
10.216.3.5 implementation()	1001
10.216.3.6 init()	1001
10.216.3.7 item_size()	1001
10.216.3.8 mag()	1001

10.216.3.9 reset()	1001
10.216.3.10 role()	1001
10.216.3.11 set_channel()	1001
10.216.3.12 set_channel_fsm()	1001
10.216.3.13 set_doppler_center()	1002
10.216.3.14 set_doppler_max()	1002
10.216.3.15 set_doppler_step()	1002
10.216.3.16 set_gnss_synchro()	1002
10.216.3.17 set_local_code()	1002
10.216.3.18 set_resampler_latency()	1002
10.216.3.19 set_state()	1002
10.216.3.20 set_threshold()	1002
10.216.3.21 stop_acquisition()	1003
10.216.4 Member Data Documentation	1003
10.216.4.1 item_type_	1003
10.217 GpsL1CaPcpsAssistedAcquisition Class Reference	1003
10.217.1 Detailed Description	1006
10.217.2 Member Function Documentation	1006
10.217.2.1 connect()	1006
10.217.2.2 disconnect()	1007
10.217.2.3 get_left_block()	1007
10.217.2.4 get_right_block()	1007
10.217.2.5 implementation()	1007
10.217.2.6 init()	1007
10.217.2.7 item_size()	1007
10.217.2.8 mag()	1007
10.217.2.9 reset()	1007
10.217.2.10 role()	1007
10.217.2.11 set_channel()	1007
10.217.2.12 set_channel_fsm()	1008
10.217.2.13 set_doppler_max()	1008
10.217.2.14 set_doppler_step()	1008
10.217.2.15 set_gnss_synchro()	1008
10.217.2.16 set_local_code()	1008
10.217.2.17 set_resampler_latency()	1008
10.217.2.18 set_state()	1008
10.217.2.19 set_threshold()	1008
10.217.2.20 stop_acquisition()	1008
10.218 GpsL1CaPcpsOpenCIAcquisition Class Reference	1009
10.218.1 Detailed Description	1011
10.218.2 Member Function Documentation	1011
10.218.2.1 connect()	1011

10.218.2.2 disconnect()	1012
10.218.2.3 get_left_block()	1012
10.218.2.4 get_right_block()	1012
10.218.2.5 implementation()	1012
10.218.2.6 init()	1012
10.218.2.7 item_size()	1012
10.218.2.8 mag()	1012
10.218.2.9 opencl_ready()	1012
10.218.2.10 reset()	1012
10.218.2.11 role()	1012
10.218.2.12 set_channel()	1013
10.218.2.13 set_channel_fsm()	1013
10.218.2.14 set_doppler_max()	1013
10.218.2.15 set_doppler_step()	1013
10.218.2.16 set_gnss_synchro()	1013
10.218.2.17 set_local_code()	1013
10.218.2.18 set_resampler_latency()	1013
10.218.2.19 set_state()	1013
10.218.2.20 set_threshold()	1013
10.218.2.21 stop_acquisition()	1014
10.219 GpsL1CaPcpsQuickSyncAcquisition Class Reference	1014
10.219.1 Detailed Description	1017
10.219.2 Member Function Documentation	1017
10.219.2.1 connect()	1017
10.219.2.2 disconnect()	1018
10.219.2.3 get_left_block()	1018
10.219.2.4 get_right_block()	1018
10.219.2.5 implementation()	1018
10.219.2.6 init()	1018
10.219.2.7 item_size()	1018
10.219.2.8 mag()	1018
10.219.2.9 reset()	1018
10.219.2.10 role()	1018
10.219.2.11 set_channel()	1018
10.219.2.12 set_channel_fsm()	1019
10.219.2.13 set_doppler_max()	1019
10.219.2.14 set_doppler_step()	1019
10.219.2.15 set_gnss_synchro()	1019
10.219.2.16 set_local_code()	1019
10.219.2.17 set_resampler_latency()	1019
10.219.2.18 set_state()	1019
10.219.2.19 set_threshold()	1019

10.219.2.20 stop_acquisition()	1019
10.220 GpsL1CaPcpsTongAcquisition Class Reference	1020
10.220.1 Detailed Description	1022
10.220.2 Member Function Documentation	1022
10.220.2.1 connect()	1022
10.220.2.2 disconnect()	1023
10.220.2.3 get_left_block()	1023
10.220.2.4 get_right_block()	1023
10.220.2.5 implementation()	1023
10.220.2.6 init()	1023
10.220.2.7 item_size()	1023
10.220.2.8 mag()	1023
10.220.2.9 reset()	1023
10.220.2.10 role()	1023
10.220.2.11 set_channel()	1023
10.220.2.12 set_channel_fsm()	1024
10.220.2.13 set_doppler_max()	1024
10.220.2.14 set_doppler_step()	1024
10.220.2.15 set_gnss_synchro()	1024
10.220.2.16 set_local_code()	1024
10.220.2.17 set_resampler_latency()	1024
10.220.2.18 set_state()	1024
10.220.2.19 set_threshold()	1024
10.220.2.20 stop_acquisition()	1024
10.221 GpsL1CaTcpConnectorTracking Class Reference	1025
10.221.1 Detailed Description	1027
10.221.2 Member Function Documentation	1027
10.221.2.1 connect()	1027
10.221.2.2 disconnect()	1027
10.221.2.3 get_left_block()	1027
10.221.2.4 get_right_block()	1027
10.221.2.5 implementation()	1027
10.221.2.6 item_size()	1028
10.221.2.7 role()	1028
10.221.2.8 set_channel()	1028
10.221.2.9 set_gnss_synchro()	1028
10.221.2.10 start_tracking()	1028
10.221.2.11 stop_tracking()	1028
10.222 GpsL1CaTelemetryDecoder Class Reference	1028
10.222.1 Detailed Description	1031
10.222.2 Member Function Documentation	1031
10.222.2.1 connect()	1031



---

10.222.2.2 disconnect()	1031
10.222.2.3 get_left_block()	1031
10.222.2.4 get_right_block()	1031
10.222.2.5 implementation()	1031
10.222.2.6 item_size()	1031
10.222.2.7 reset()	1032
10.222.2.8 role()	1032
10.222.2.9 set_channel()	1032
10.222.2.10 set_satellite()	1032
10.223 GpsL2CTelemetryDecoder Class Reference	1032
10.223.1 Detailed Description	1035
10.223.2 Member Function Documentation	1035
10.223.2.1 connect()	1035
10.223.2.2 disconnect()	1035
10.223.2.3 get_left_block()	1035
10.223.2.4 get_right_block()	1035
10.223.2.5 implementation()	1035
10.223.2.6 item_size()	1035
10.223.2.7 reset()	1036
10.223.2.8 role()	1036
10.223.2.9 set_channel()	1036
10.223.2.10 set_satellite()	1036
10.224 GpsL2MDIPIITracking Class Reference	1036
10.224.1 Detailed Description	1039
10.224.2 Member Function Documentation	1039
10.224.2.1 connect()	1039
10.224.2.2 disconnect()	1039
10.224.2.3 get_left_block()	1039
10.224.2.4 get_right_block()	1039
10.224.2.5 implementation()	1039
10.224.2.6 item_size()	1040
10.224.2.7 role()	1040
10.224.2.8 set_channel()	1040
10.224.2.9 set_gnss_synchro()	1040
10.224.2.10 start_tracking()	1040
10.224.2.11 stop_tracking()	1040
10.225 GpsL2MDIPIITrackingFpga Class Reference	1040
10.225.1 Detailed Description	1043
10.225.2 Member Function Documentation	1043
10.225.2.1 connect()	1043
10.225.2.2 disconnect()	1043
10.225.2.3 get_left_block()	1043

10.225.2.4	<a href="#">get_right_block()</a>	1043
10.225.2.5	<a href="#">implementation()</a>	1043
10.225.2.6	<a href="#">item_size()</a>	1044
10.225.2.7	<a href="#">role()</a>	1044
10.225.2.8	<a href="#">set_channel()</a>	1044
10.225.2.9	<a href="#">set_gnss_synchro()</a>	1044
10.225.2.10	<a href="#">start_tracking()</a>	1044
10.225.2.11	<a href="#">stop_tracking()</a>	1044
10.226	<a href="#">GpsL2MPcpsAcquisition Class Reference</a>	1044
10.226.1	<a href="#">Detailed Description</a>	1047
10.226.2	<a href="#">Member Function Documentation</a>	1048
10.226.2.1	<a href="#">connect()</a>	1048
10.226.2.2	<a href="#">disconnect()</a>	1048
10.226.2.3	<a href="#">get_left_block()</a>	1048
10.226.2.4	<a href="#">get_right_block()</a>	1048
10.226.2.5	<a href="#">implementation()</a>	1048
10.226.2.6	<a href="#">init()</a>	1048
10.226.2.7	<a href="#">item_size()</a>	1048
10.226.2.8	<a href="#">mag()</a>	1048
10.226.2.9	<a href="#">reset()</a>	1048
10.226.2.10	<a href="#">role()</a>	1048
10.226.2.11	<a href="#">set_channel()</a>	1049
10.226.2.12	<a href="#">set_channel_fsm()</a>	1049
10.226.2.13	<a href="#">set_doppler_center()</a>	1049
10.226.2.14	<a href="#">set_doppler_max()</a>	1049
10.226.2.15	<a href="#">set_doppler_step()</a>	1049
10.226.2.16	<a href="#">set_gnss_synchro()</a>	1049
10.226.2.17	<a href="#">set_local_code()</a>	1049
10.226.2.18	<a href="#">set_resampler_latency()</a>	1049
10.226.2.19	<a href="#">set_state()</a>	1050
10.226.2.20	<a href="#">set_threshold()</a>	1050
10.226.2.21	<a href="#">stop_acquisition()</a>	1050
10.227	<a href="#">GpsL2MPcpsAcquisitionFpga Class Reference</a>	1050
10.227.1	<a href="#">Detailed Description</a>	1053
10.227.2	<a href="#">Member Function Documentation</a>	1053
10.227.2.1	<a href="#">connect()</a>	1053
10.227.2.2	<a href="#">disconnect()</a>	1054
10.227.2.3	<a href="#">get_left_block()</a>	1054
10.227.2.4	<a href="#">get_right_block()</a>	1054
10.227.2.5	<a href="#">implementation()</a>	1054
10.227.2.6	<a href="#">init()</a>	1054
10.227.2.7	<a href="#">item_size()</a>	1054

10.227.2.8 mag()	1054
10.227.2.9 reset()	1054
10.227.2.10 role()	1054
10.227.2.11 set_channel()	1054
10.227.2.12 set_channel_fsm()	1055
10.227.2.13 set_doppler_max()	1055
10.227.2.14 set_doppler_step()	1055
10.227.2.15 set_gnss_synchro()	1055
10.227.2.16 set_local_code()	1055
10.227.2.17 set_resampler_latency()	1055
10.227.2.18 set_state()	1055
10.227.2.19 set_threshold()	1055
10.227.2.20 stop_acquisition()	1055
10.228 GpsL5DIIPITracking Class Reference	1056
10.228.1 Detailed Description	1058
10.228.2 Member Function Documentation	1058
10.228.2.1 connect()	1058
10.228.2.2 disconnect()	1058
10.228.2.3 get_left_block()	1058
10.228.2.4 get_right_block()	1058
10.228.2.5 implementation()	1058
10.228.2.6 item_size()	1059
10.228.2.7 role()	1059
10.228.2.8 set_channel()	1059
10.228.2.9 set_gnss_synchro()	1059
10.228.2.10 start_tracking()	1059
10.228.2.11 stop_tracking()	1059
10.229 GpsL5DIIPITrackingFpga Class Reference	1059
10.229.1 Detailed Description	1062
10.229.2 Constructor & Destructor Documentation	1062
10.229.2.1 GpsL5DIIPITrackingFpga()	1062
10.229.2.2 ~GpsL5DIIPITrackingFpga()	1063
10.229.3 Member Function Documentation	1063
10.229.3.1 connect()	1063
10.229.3.2 disconnect()	1063
10.229.3.3 get_left_block()	1063
10.229.3.4 get_right_block()	1063
10.229.3.5 implementation()	1063
10.229.3.6 item_size()	1063
10.229.3.7 role()	1063
10.229.3.8 set_channel()	1063
10.229.3.9 set_gnss_synchro()	1064

10.229.3.10 start_tracking()	1064
10.229.3.11 stop_tracking()	1064
10.230 GpsL5iPcpsAcquisition Class Reference	1064
10.230.1 Detailed Description	1067
10.230.2 Member Function Documentation	1068
10.230.2.1 connect()	1068
10.230.2.2 disconnect()	1068
10.230.2.3 get_left_block()	1068
10.230.2.4 get_right_block()	1068
10.230.2.5 implementation()	1068
10.230.2.6 init()	1068
10.230.2.7 item_size()	1068
10.230.2.8 mag()	1068
10.230.2.9 reset()	1068
10.230.2.10 role()	1068
10.230.2.11 set_channel()	1069
10.230.2.12 set_channel_fsm()	1069
10.230.2.13 set_doppler_center()	1069
10.230.2.14 set_doppler_max()	1069
10.230.2.15 set_doppler_step()	1069
10.230.2.16 set_gnss_synchro()	1069
10.230.2.17 set_local_code()	1069
10.230.2.18 set_resampler_latency()	1069
10.230.2.19 set_state()	1070
10.230.2.20 set_threshold()	1070
10.230.2.21 stop_acquisition()	1070
10.231 GpsL5iPcpsAcquisitionFpga Class Reference	1070
10.231.1 Detailed Description	1074
10.231.2 Constructor & Destructor Documentation	1074
10.231.2.1 GpsL5iPcpsAcquisitionFpga()	1074
10.231.2.2 ~GpsL5iPcpsAcquisitionFpga()	1074
10.231.3 Member Function Documentation	1074
10.231.3.1 connect()	1074
10.231.3.2 disconnect()	1074
10.231.3.3 get_left_block()	1074
10.231.3.4 get_right_block()	1074
10.231.3.5 implementation()	1075
10.231.3.6 init()	1075
10.231.3.7 item_size()	1075
10.231.3.8 mag()	1075
10.231.3.9 reset()	1075
10.231.3.10 role()	1075

10.231.3.11 set_channel()	1075
10.231.3.12 set_channel_fsm()	1075
10.231.3.13 set_doppler_center()	1075
10.231.3.14 set_doppler_max()	1076
10.231.3.15 set_doppler_step()	1076
10.231.3.16 set_gnss_synchro()	1076
10.231.3.17 set_local_code()	1076
10.231.3.18 set_resampler_latency()	1076
10.231.3.19 set_state()	1076
10.231.3.20 set_threshold()	1076
10.231.3.21 stop_acquisition()	1076
10.232 GpsL5TelemetryDecoder Class Reference	1077
10.232.1 Detailed Description	1079
10.232.2 Member Function Documentation	1079
10.232.2.1 connect()	1079
10.232.2.2 disconnect()	1079
10.232.2.3 get_left_block()	1079
10.232.2.4 get_right_block()	1079
10.232.2.5 implementation()	1079
10.232.2.6 item_size()	1079
10.232.2.7 reset()	1080
10.232.2.8 role()	1080
10.232.2.9 set_channel()	1080
10.232.2.10 set_satellite()	1080
10.233 GPU_Complex Struct Reference	1080
10.233.1 Detailed Description	1081
10.233.2 Constructor & Destructor Documentation	1081
10.233.2.1 GPU_Complex() [1/2]	1081
10.233.2.2 GPU_Complex() [2/2]	1081
10.233.3 Member Function Documentation	1081
10.233.3.1 magnitude2()	1081
10.233.3.2 multiply_acc()	1081
10.233.3.3 operator*()	1081
10.233.3.4 operator+()	1081
10.233.3.5 operator+=()	1081
10.233.4 Member Data Documentation	1082
10.233.4.1 i	1082
10.233.4.2 r	1082
10.234 GPU_Complex_Short Struct Reference	1082
10.234.1 Detailed Description	1082
10.234.2 Constructor & Destructor Documentation	1082
10.234.2.1 GPU_Complex_Short()	1082

10.234.3 Member Function Documentation . . . . .	1083
10.234.3.1 magnitude2() . . . . .	1083
10.234.3.2 operator*() . . . . .	1083
10.234.3.3 operator+() . . . . .	1083
10.234.4 Member Data Documentation . . . . .	1083
10.234.4.1 i . . . . .	1083
10.234.4.2 r . . . . .	1083
10.235 Gpx_Printer Class Reference . . . . .	1083
10.235.1 Detailed Description . . . . .	1084
10.236 Gr_Complex_Ip_Packet_Source Class Reference . . . . .	1084
10.236.1 Detailed Description . . . . .	1085
10.236.2 Member Typedef Documentation . . . . .	1085
10.236.2.1 sprtr . . . . .	1085
10.237 gtime_t Struct Reference . . . . .	1086
10.237.1 Detailed Description . . . . .	1086
10.237.2 Member Data Documentation . . . . .	1086
10.237.2.1 sec . . . . .	1086
10.237.2.2 time . . . . .	1086
10.238 half_cyc_tag Struct Reference . . . . .	1087
10.238.1 Detailed Description . . . . .	1087
10.238.2 Member Data Documentation . . . . .	1087
10.238.2.1 corr . . . . .	1087
10.238.2.2 freq . . . . .	1087
10.238.2.3 next . . . . .	1088
10.238.2.4 sat . . . . .	1088
10.238.2.5 te . . . . .	1088
10.238.2.6 ts . . . . .	1088
10.238.2.7 valid . . . . .	1088
10.239 HAS_clock_corrections Class Reference . . . . .	1088
10.239.1 Detailed Description . . . . .	1088
10.239.2 Member Data Documentation . . . . .	1089
10.239.2.1 clock_correction_m . . . . .	1089
10.239.2.2 valid_until . . . . .	1089
10.240 HAS_obs_corrections Class Reference . . . . .	1089
10.240.1 Detailed Description . . . . .	1089
10.240.2 Member Data Documentation . . . . .	1089
10.240.2.1 code_bias_m . . . . .	1089
10.240.2.2 phase_bias_cycle . . . . .	1089
10.241 HAS_orbit_corrections Class Reference . . . . .	1090
10.241.1 Detailed Description . . . . .	1090
10.241.2 Member Data Documentation . . . . .	1090
10.241.2.1 cross_track_m . . . . .	1090

10.241.2.2 in_track_m . . . . .	1090
10.241.2.3 iod . . . . .	1090
10.241.2.4 radial_m . . . . .	1090
10.241.2.5 valid_until . . . . .	1091
10.242 Has_Simple_Printer Class Reference . . . . .	1091
10.242.1 Detailed Description . . . . .	1091
10.243 hybrid_observables_gs Class Reference . . . . .	1091
10.243.1 Detailed Description . . . . .	1093
10.244 HybridObservables Class Reference . . . . .	1093
10.244.1 Detailed Description . . . . .	1096
10.244.2 Member Function Documentation . . . . .	1096
10.244.2.1 connect() . . . . .	1096
10.244.2.2 disconnect() . . . . .	1096
10.244.2.3 get_left_block() . . . . .	1096
10.244.2.4 get_right_block() . . . . .	1096
10.244.2.5 implementation() . . . . .	1096
10.244.2.6 item_size() . . . . .	1096
10.244.2.7 reset() . . . . .	1097
10.244.2.8 role() . . . . .	1097
10.245 lbyteToCbyte Class Reference . . . . .	1097
10.245.1 Detailed Description . . . . .	1100
10.245.2 Member Function Documentation . . . . .	1100
10.245.2.1 connect() . . . . .	1100
10.245.2.2 disconnect() . . . . .	1100
10.245.2.3 get_left_block() . . . . .	1100
10.245.2.4 get_right_block() . . . . .	1100
10.245.2.5 implementation() . . . . .	1100
10.245.2.6 item_size() . . . . .	1100
10.245.2.7 role() . . . . .	1100
10.246 lbyteToComplex Class Reference . . . . .	1101
10.246.1 Detailed Description . . . . .	1103
10.246.2 Member Function Documentation . . . . .	1103
10.246.2.1 connect() . . . . .	1103
10.246.2.2 disconnect() . . . . .	1103
10.246.2.3 get_left_block() . . . . .	1103
10.246.2.4 get_right_block() . . . . .	1103
10.246.2.5 implementation() . . . . .	1103
10.246.2.6 item_size() . . . . .	1103
10.246.2.7 role() . . . . .	1103
10.247 lbyteToCshort Class Reference . . . . .	1104
10.247.1 Detailed Description . . . . .	1106
10.247.2 Member Function Documentation . . . . .	1106

10.247.2.1 connect()	1106
10.247.2.2 disconnect()	1106
10.247.2.3 get_left_block()	1106
10.247.2.4 get_right_block()	1106
10.247.2.5 implementation()	1106
10.247.2.6 item_size()	1106
10.247.2.7 role()	1106
10.248 INIReader Class Reference	1107
10.248.1 Detailed Description	1107
10.248.2 Constructor & Destructor Documentation	1107
10.248.2.1 INIReader()	1107
10.248.3 Member Function Documentation	1108
10.248.3.1 Get()	1108
10.248.3.2 GetInteger()	1108
10.248.3.3 HasSection()	1108
10.248.3.4 HasValue()	1108
10.248.3.5 ParseError()	1108
10.249 InMemoryConfiguration Class Reference	1108
10.249.1 Detailed Description	1111
10.249.2 Member Function Documentation	1111
10.249.2.1 property() [1/10]	1111
10.249.2.2 property() [2/10]	1111
10.249.2.3 property() [3/10]	1111
10.249.2.4 property() [4/10]	1111
10.249.2.5 property() [5/10]	1111
10.249.2.6 property() [6/10]	1111
10.249.2.7 property() [7/10]	1112
10.249.2.8 property() [8/10]	1112
10.249.2.9 property() [9/10]	1112
10.249.2.10 property() [10/10]	1112
10.249.2.11 set_property()	1112
10.250 interleaved_byte_to_complex_byte Class Reference	1112
10.250.1 Detailed Description	1113
10.251 interleaved_byte_to_complex_short Class Reference	1114
10.251.1 Detailed Description	1115
10.252 interleaved_short_to_complex_short Class Reference	1115
10.252.1 Detailed Description	1116
10.253 IONGSMSChunkData Class Reference	1117
10.253.1 Detailed Description	1117
10.254 IONGSMSChunkUnpackingCtx< WT > Struct Template Reference	1118
10.254.1 Detailed Description	1118
10.254.2 Constructor & Destructor Documentation	1119



10.254.2.1 IONGSMSChunkUnpackingCtx()	1119
10.254.3 Member Function Documentation	1119
10.254.3.1 advance_word()	1119
10.254.3.2 shift_current_word()	1119
10.254.3.3 shift_padding()	1119
10.254.3.4 shift_sample()	1119
10.254.4 Member Data Documentation	1119
10.254.4.1 bitshift_	1119
10.254.4.2 current_word_	1119
10.254.4.3 iterator_	1119
10.254.4.4 word_bitsize_	1120
10.254.4.5 word_shift_direction_	1120
10.255 IONGSMSFileSource Class Reference	1120
10.255.1 Detailed Description	1121
10.255.2 Member Typedef Documentation	1121
10.255.2.1 sptr	1121
10.256 IONGSMSSignalSource Class Reference	1122
10.256.1 Detailed Description	1124
10.256.2 Member Function Documentation	1124
10.256.2.1 connect()	1124
10.256.2.2 disconnect()	1124
10.256.2.3 get_left_block()	1124
10.256.2.4 get_right_block() [1/2]	1124
10.256.2.5 get_right_block() [2/2]	1125
10.256.2.6 item_size()	1125
10.257 IshortToComplex Class Reference	1125
10.257.1 Detailed Description	1128
10.257.2 Member Function Documentation	1128
10.257.2.1 connect()	1128
10.257.2.2 disconnect()	1128
10.257.2.3 get_left_block()	1128
10.257.2.4 get_right_block()	1128
10.257.2.5 implementation()	1128
10.257.2.6 item_size()	1128
10.257.2.7 role()	1128
10.258 IshortToCshort Class Reference	1129
10.258.1 Detailed Description	1131
10.258.2 Member Function Documentation	1131
10.258.2.1 connect()	1131
10.258.2.2 disconnect()	1131
10.258.2.3 get_left_block()	1131
10.258.2.4 get_right_block()	1131

10.258.2.5 implementation()	1131
10.258.2.6 item_size()	1131
10.258.2.7 role()	1131
10.259 kernel_info_t Struct Reference	1132
10.259.1 Detailed Description	1132
10.259.2 Member Data Documentation	1132
10.259.2.1 dir	1132
10.259.2.2 in_place_possible	1132
10.259.2.3 kernel	1132
10.259.2.4 kernel_name	1133
10.259.2.5 lmem_size	1133
10.259.2.6 next	1133
10.259.2.7 num_workgroups	1133
10.259.2.8 num_workitems_per_workgroup	1133
10.259.2.9 num_xforms_per_workgroup	1133
10.260 Kf_Conf Class Reference	1134
10.260.1 Detailed Description	1135
10.260.2 Member Data Documentation	1135
10.260.2.1 bit_synchronization_time_limit_s	1135
10.260.2.2 carrier_disc_sd_rads	1135
10.260.2.3 carrier_freq_rate_sd_hz_s	1136
10.260.2.4 carrier_freq_sd_hz	1136
10.260.2.5 carrier_lock_test_smoother_alpha	1136
10.260.2.6 carrier_lock_test_smoother_samples	1136
10.260.2.7 carrier_lock_th	1136
10.260.2.8 carrier_phase_sd_rad	1136
10.260.2.9 cn0_min	1136
10.260.2.10 cn0_samples	1136
10.260.2.11 cn0_smoother_alpha	1136
10.260.2.12 cn0_smoother_samples	1136
10.260.2.13 code_disc_sd_chips	1136
10.260.2.14 code_phase_sd_chips	1136
10.260.2.15 dump	1137
10.260.2.16 dump_filename	1137
10.260.2.17 dump_mat	1137
10.260.2.18 early_late_space_chips	1137
10.260.2.19 early_late_space_narrow_chips	1137
10.260.2.20 enable_doppler_correction	1137
10.260.2.21 extend_correlation_symbols	1137
10.260.2.22 fs_in	1137
10.260.2.23 high_dyn	1137
10.260.2.24 init_carrier_freq_rate_sd_hz_s	1137

10.260.2.25	<a href="#">init_carrier_freq_sd_hz</a>	1137
10.260.2.26	<a href="#">init_carrier_phase_sd_rad</a>	1137
10.260.2.27	<a href="#">init_code_phase_sd_chips</a>	1138
10.260.2.28	<a href="#">item_type</a>	1138
10.260.2.29	<a href="#">max_carrier_lock_fail</a>	1138
10.260.2.30	<a href="#">max_code_lock_fail</a>	1138
10.260.2.31	<a href="#">pull_in_time_s</a>	1138
10.260.2.32	<a href="#">signal</a>	1138
10.260.2.33	<a href="#">slope</a>	1138
10.260.2.34	<a href="#">smoother_length</a>	1138
10.260.2.35	<a href="#">spc</a>	1138
10.260.2.36	<a href="#">system</a>	1138
10.260.2.37	<a href="#">track_pilot</a>	1138
10.260.2.38	<a href="#">vector_length</a>	1138
10.260.2.39	<a href="#">very_early_late_space_chips</a>	1139
10.260.2.40	<a href="#">very_early_late_space_narrow_chips</a>	1139
10.260.2.41	<a href="#">y_intercept</a>	1139
10.261	<a href="#">kf_tracking Class Reference</a>	1139
10.261.1	<a href="#">Detailed Description</a>	1140
10.262	<a href="#">Kml_Printer Class Reference</a>	1140
10.262.1	<a href="#">Detailed Description</a>	1141
10.263	<a href="#">labsat23_source Class Reference</a>	1141
10.263.1	<a href="#">Detailed Description</a>	1142
10.264	<a href="#">LabsatSignalSource Class Reference</a>	1143
10.264.1	<a href="#">Detailed Description</a>	1145
10.264.2	<a href="#">Member Function Documentation</a>	1145
10.264.2.1	<a href="#">connect()</a>	1145
10.264.2.2	<a href="#">disconnect()</a>	1145
10.264.2.3	<a href="#">get_left_block()</a>	1145
10.264.2.4	<a href="#">get_right_block()</a> [1/2]	1145
10.264.2.5	<a href="#">get_right_block()</a> [2/2]	1146
10.264.2.6	<a href="#">getRfChannels()</a>	1146
10.264.2.7	<a href="#">item_size()</a>	1146
10.265	<a href="#">lex_t Struct Reference</a>	1146
10.265.1	<a href="#">Detailed Description</a>	1147
10.265.2	<a href="#">Member Data Documentation</a>	1147
10.265.2.1	<a href="#">msgs</a>	1147
10.265.2.2	<a href="#">n</a>	1147
10.265.2.3	<a href="#">nmax</a>	1147
10.266	<a href="#">lexeph_t Struct Reference</a>	1148
10.266.1	<a href="#">Detailed Description</a>	1148
10.266.2	<a href="#">Member Data Documentation</a>	1149

10.266.2.1 acc	1149
10.266.2.2 af0	1149
10.266.2.3 af1	1149
10.266.2.4 health	1149
10.266.2.5 isc	1149
10.266.2.6 jerk	1149
10.266.2.7 pos	1149
10.266.2.8 sat	1149
10.266.2.9 tgd	1149
10.266.2.10 toe	1149
10.266.2.11 tof	1149
10.266.2.12 ura	1149
10.266.2.13 vel	1150
10.267 lexion_t Struct Reference	1150
10.267.1 Detailed Description	1150
10.267.2 Member Data Documentation	1150
10.267.2.1 coef	1150
10.267.2.2 pos0	1150
10.267.2.3 t0	1151
10.267.2.4 tspan	1151
10.268 lexmsg_t Struct Reference	1151
10.268.1 Detailed Description	1151
10.268.2 Member Data Documentation	1151
10.268.2.1 alert	1151
10.268.2.2 msg	1152
10.268.2.3 prn	1152
10.268.2.4 snr	1152
10.268.2.5 stat	1152
10.268.2.6 ttt	1152
10.268.2.7 type	1152
10.269 LimesdrSignalSource Class Reference	1152
10.269.1 Detailed Description	1155
10.269.2 Member Function Documentation	1155
10.269.2.1 connect()	1155
10.269.2.2 disconnect()	1155
10.269.2.3 get_left_block()	1155
10.269.2.4 get_right_block()	1155
10.269.2.5 item_size()	1156
10.270 MACK_header Class Reference	1156
10.270.1 Detailed Description	1156
10.270.2 Member Data Documentation	1156
10.270.2.1 cop	1156

10.270.2.2 macseq . . . . .	1156
10.270.2.3 tag0 . . . . .	1156
10.271 Mack_lookup Class Reference . . . . .	1157
10.271.1 Detailed Description . . . . .	1158
10.271.2 Constructor & Destructor Documentation . . . . .	1158
10.271.2.1 Mack_lookup() . . . . .	1158
10.271.3 Member Data Documentation . . . . .	1158
10.271.3.1 msg . . . . .	1158
10.271.3.2 nt . . . . .	1158
10.271.3.3 sequence1 . . . . .	1158
10.271.3.4 sequence2 . . . . .	1158
10.272 MACK_message Class Reference . . . . .	1159
10.272.1 Detailed Description . . . . .	1159
10.272.2 Member Data Documentation . . . . .	1160
10.272.2.1 header . . . . .	1160
10.272.2.2 key . . . . .	1160
10.272.2.3 PRNa . . . . .	1160
10.272.2.4 tag_and_info . . . . .	1160
10.272.2.5 TOW . . . . .	1160
10.272.2.6 WN . . . . .	1160
10.273 MACK_tag_and_info Class Reference . . . . .	1161
10.273.1 Detailed Description . . . . .	1161
10.273.2 Member Data Documentation . . . . .	1161
10.273.2.1 counter . . . . .	1161
10.273.2.2 tag . . . . .	1161
10.273.2.3 tag_info . . . . .	1161
10.274 MACK_tag_info Class Reference . . . . .	1162
10.274.1 Detailed Description . . . . .	1162
10.274.2 Member Data Documentation . . . . .	1162
10.274.2.1 ADKD . . . . .	1162
10.274.2.2 cop . . . . .	1162
10.274.2.3 PRN_d . . . . .	1162
10.275 MAX2771EVKITSignalSourceFPGA Class Reference . . . . .	1163
10.275.1 Detailed Description . . . . .	1165
10.275.2 Member Function Documentation . . . . .	1165
10.275.2.1 connect() . . . . .	1165
10.275.2.2 disconnect() . . . . .	1165
10.275.2.3 get_left_block() . . . . .	1165
10.275.2.4 get_right_block() . . . . .	1165
10.275.2.5 item_size() . . . . .	1166
10.276 MmseResamplerConditioner Class Reference . . . . .	1166
10.276.1 Detailed Description . . . . .	1168

10.276.2 Member Function Documentation	1168
10.276.2.1 connect()	1168
10.276.2.2 disconnect()	1168
10.276.2.3 get_left_block()	1168
10.276.2.4 get_right_block()	1168
10.276.2.5 implementation()	1168
10.276.2.6 item_size()	1168
10.276.2.7 role()	1168
10.277 ModelFunction Class Reference	1169
10.277.1 Detailed Description	1169
10.277.2 Constructor & Destructor Documentation	1169
10.277.2.1 ModelFunction()	1169
10.278 Monitor_Ephemeris_Udp_Sink Class Reference	1169
10.278.1 Detailed Description	1170
10.279 Monitor_Pvt Class Reference	1170
10.279.1 Detailed Description	1172
10.279.2 Member Function Documentation	1172
10.279.2.1 serialize()	1172
10.279.3 Member Data Documentation	1172
10.279.3.1 AR_ratio_factor	1172
10.279.3.2 AR_ratio_threshold	1172
10.279.3.3 cog	1173
10.279.3.4 cov_xx	1173
10.279.3.5 cov_xy	1173
10.279.3.6 cov_yy	1173
10.279.3.7 cov_yz	1173
10.279.3.8 cov_zx	1173
10.279.3.9 cov_zz	1173
10.279.3.10 galhas_status	1173
10.279.3.11 gdop	1173
10.279.3.12 geohash	1173
10.279.3.13 hdop	1173
10.279.3.14 height	1173
10.279.3.15 latitude	1174
10.279.3.16 longitude	1174
10.279.3.17 pdop	1174
10.279.3.18 pos_x	1174
10.279.3.19 pos_y	1174
10.279.3.20 pos_z	1174
10.279.3.21 RX_time	1174
10.279.3.22 solution_status	1174
10.279.3.23 solution_type	1174

10.279.3.24 TOW_at_current_symbol_ms . . . . .	1174
10.279.3.25 user_clk_drift_ppm . . . . .	1174
10.279.3.26 user_clk_offset . . . . .	1174
10.279.3.27 utc_time . . . . .	1175
10.279.3.28 valid_sats . . . . .	1175
10.279.3.29 vdop . . . . .	1175
10.279.3.30 vel_e . . . . .	1175
10.279.3.31 vel_n . . . . .	1175
10.279.3.32 vel_u . . . . .	1175
10.279.3.33 vel_x . . . . .	1175
10.279.3.34 vel_y . . . . .	1175
10.279.3.35 vel_z . . . . .	1175
10.279.3.36 week . . . . .	1175
10.280 Monitor_Pvt_Udp_Sink Class Reference . . . . .	1176
10.280.1 Detailed Description . . . . .	1176
10.281 msm_h_t Struct Reference . . . . .	1177
10.281.1 Detailed Description . . . . .	1177
10.281.2 Member Data Documentation . . . . .	1177
10.281.2.1 cellmask . . . . .	1177
10.281.2.2 clk_ext . . . . .	1177
10.281.2.3 clk_str . . . . .	1178
10.281.2.4 iod . . . . .	1178
10.281.2.5 nsat . . . . .	1178
10.281.2.6 nsig . . . . .	1178
10.281.2.7 sats . . . . .	1178
10.281.2.8 sigs . . . . .	1178
10.281.2.9 smooth . . . . .	1178
10.281.2.10 time_s . . . . .	1178
10.281.2.11 tint_s . . . . .	1178
10.282 mt1_header Struct Reference . . . . .	1179
10.282.1 Detailed Description . . . . .	1179
10.282.2 Member Data Documentation . . . . .	1179
10.282.2.1 clock_fullset_flag . . . . .	1179
10.282.2.2 clock_subset_flag . . . . .	1179
10.282.2.3 code_bias_flag . . . . .	1180
10.282.2.4 iod_set_id . . . . .	1180
10.282.2.5 mask_flag . . . . .	1180
10.282.2.6 mask_id . . . . .	1180
10.282.2.7 orbit_correction_flag . . . . .	1180
10.282.2.8 phase_bias_flag . . . . .	1180
10.282.2.9 reserved . . . . .	1180
10.282.2.10 toh . . . . .	1180

10.283 MultichannelFileSignalSource Class Reference . . . . .	1180
10.283.1 Detailed Description . . . . .	1183
10.283.2 Member Function Documentation . . . . .	1183
10.283.2.1 connect() . . . . .	1183
10.283.2.2 disconnect() . . . . .	1183
10.283.2.3 filename() . . . . .	1183
10.283.2.4 get_left_block() . . . . .	1183
10.283.2.5 get_right_block() . . . . .	1184
10.283.2.6 item_size() . . . . .	1184
10.283.2.7 item_type() . . . . .	1184
10.283.2.8 repeat() . . . . .	1184
10.283.2.9 samples() . . . . .	1184
10.283.2.10 sampling_frequency() . . . . .	1184
10.284 nav_message_monitor Class Reference . . . . .	1184
10.284.1 Detailed Description . . . . .	1185
10.284.2 Constructor & Destructor Documentation . . . . .	1185
10.284.2.1 ~nav_message_monitor() . . . . .	1185
10.285 Nav_Message_Packet Class Reference . . . . .	1186
10.285.1 Detailed Description . . . . .	1187
10.285.2 Constructor & Destructor Documentation . . . . .	1187
10.285.2.1 Nav_Message_Packet() [1/3] . . . . .	1187
10.285.2.2 ~Nav_Message_Packet() . . . . .	1187
10.285.2.3 Nav_Message_Packet() [2/3] . . . . .	1187
10.285.2.4 Nav_Message_Packet() [3/3] . . . . .	1187
10.285.3 Member Function Documentation . . . . .	1188
10.285.3.1 operator=() [1/2] . . . . .	1188
10.285.3.2 operator=() [2/2] . . . . .	1188
10.285.4 Member Data Documentation . . . . .	1189
10.285.4.1 nav_message . . . . .	1189
10.285.4.2 prn . . . . .	1189
10.285.4.3 signal . . . . .	1189
10.285.4.4 system . . . . .	1189
10.285.4.5 tow_at_current_symbol_ms . . . . .	1189
10.286 Nav_Message_Udp_Sink Class Reference . . . . .	1189
10.286.1 Detailed Description . . . . .	1190
10.287 nav_t Struct Reference . . . . .	1190
10.287.1 Detailed Description . . . . .	1191
10.287.2 Member Data Documentation . . . . .	1192
10.287.2.1 alm . . . . .	1192
10.287.2.2 cbias . . . . .	1192
10.287.2.3 dgps . . . . .	1192
10.287.2.4 eph . . . . .	1192



10.287.2.5 erp	1192
10.287.2.6 fcb	1192
10.287.2.7 geph	1192
10.287.2.8 glo_cpbias	1192
10.287.2.9 glo_fcn	1192
10.287.2.10 ion_cmp	1192
10.287.2.11 ion_gal	1192
10.287.2.12 ion_gps	1192
10.287.2.13 ion_irn	1193
10.287.2.14 ion_qzs	1193
10.287.2.15 lam	1193
10.287.2.16 leaps	1193
10.287.2.17 lexeph	1193
10.287.2.18 lexion	1193
10.287.2.19 n	1193
10.287.2.20 na	1193
10.287.2.21 namax	1193
10.287.2.22 nc	1193
10.287.2.23 ncmax	1193
10.287.2.24 ne	1193
10.287.2.25 nemax	1194
10.287.2.26 nf	1194
10.287.2.27 nfmex	1194
10.287.2.28 ng	1194
10.287.2.29 ngmax	1194
10.287.2.30 nmax	1194
10.287.2.31 ns	1194
10.287.2.32 nsmax	1194
10.287.2.33 nt	1194
10.287.2.34 ntmax	1194
10.287.2.35 pclk	1194
10.287.2.36 pcvs	1194
10.287.2.37 pepb	1195
10.287.2.38 pppcorr	1195
10.287.2.39 rbias	1195
10.287.2.40 sbsion	1195
10.287.2.41 sbssat	1195
10.287.2.42 sepb	1195
10.287.2.43 ssr	1195
10.287.2.44 tec	1195
10.287.2.45 utc_cmp	1195
10.287.2.46 utc_gal	1195

10.287.2.47 utc_glo . . . . .	1195
10.287.2.48 utc_gps . . . . .	1195
10.287.2.49 utc_irn . . . . .	1196
10.287.2.50 utc_qzs . . . . .	1196
10.287.2.51 utc_sbs . . . . .	1196
10.287.2.52 wlbias . . . . .	1196
10.288 Nmea_Printer Class Reference . . . . .	1196
10.288.1 Detailed Description . . . . .	1197
10.288.2 Constructor & Destructor Documentation . . . . .	1197
10.288.2.1 Nmea_Printer() . . . . .	1197
10.288.2.2 ~Nmea_Printer() . . . . .	1197
10.288.3 Member Function Documentation . . . . .	1197
10.288.3.1 Print_Nmea_Line() . . . . .	1197
10.289 Notch Class Reference . . . . .	1197
10.289.1 Detailed Description . . . . .	1198
10.290 NotchFilter Class Reference . . . . .	1199
10.290.1 Detailed Description . . . . .	1201
10.290.2 Member Function Documentation . . . . .	1201
10.290.2.1 connect() . . . . .	1201
10.290.2.2 disconnect() . . . . .	1201
10.290.2.3 get_left_block() . . . . .	1201
10.290.2.4 get_right_block() . . . . .	1201
10.290.2.5 implementation() . . . . .	1201
10.290.2.6 item_size() . . . . .	1201
10.290.2.7 role() . . . . .	1201
10.291 NotchFilterLite Class Reference . . . . .	1202
10.291.1 Detailed Description . . . . .	1204
10.291.2 Member Function Documentation . . . . .	1204
10.291.2.1 connect() . . . . .	1204
10.291.2.2 disconnect() . . . . .	1204
10.291.2.3 get_left_block() . . . . .	1204
10.291.2.4 get_right_block() . . . . .	1204
10.291.2.5 implementation() . . . . .	1204
10.291.2.6 item_size() . . . . .	1204
10.291.2.7 role() . . . . .	1204
10.292 NotchLite Class Reference . . . . .	1205
10.292.1 Detailed Description . . . . .	1206
10.293 NsrFileSignalSource Class Reference . . . . .	1206
10.293.1 Detailed Description . . . . .	1210
10.293.2 Member Function Documentation . . . . .	1210
10.293.2.1 create_file_source_hook() . . . . .	1210
10.293.2.2 itemTypeToSize() . . . . .	1210

10.293.2.3 packetsPerSample()	1210
10.293.2.4 pre_connect_hook()	1210
10.293.2.5 pre_disconnect_hook()	1211
10.293.2.6 source()	1211
10.294 ntrip_t Struct Reference	1212
10.294.1 Detailed Description	1213
10.294.2 Member Data Documentation	1213
10.294.2.1 buff	1213
10.294.2.2 mntpnt	1213
10.294.2.3 nb	1213
10.294.2.4 passwd	1213
10.294.2.5 state	1213
10.294.2.6 str	1213
10.294.2.7 tcp	1213
10.294.2.8 type	1213
10.294.2.9 url	1213
10.294.2.10 user	1214
10.295 Obs_Conf Class Reference	1214
10.295.1 Detailed Description	1215
10.295.2 Member Data Documentation	1215
10.295.2.1 always_output_gs	1215
10.295.2.2 dump	1215
10.295.2.3 dump_filename	1215
10.295.2.4 dump_mat	1215
10.295.2.5 enable_carrier_smoothing	1215
10.295.2.6 enable_E6	1215
10.295.2.7 nchannels_in	1215
10.295.2.8 nchannels_out	1215
10.295.2.9 observable_interval_ms	1215
10.295.2.10 smoothing_factor	1216
10.296 obs_t Struct Reference	1216
10.296.1 Detailed Description	1217
10.296.2 Member Data Documentation	1217
10.296.2.1 data	1217
10.296.2.2 n	1217
10.296.2.3 nmax	1217
10.297 obsd_t Struct Reference	1217
10.297.1 Detailed Description	1218
10.297.2 Member Data Documentation	1218
10.297.2.1 code	1218
10.297.2.2 D	1218
10.297.2.3 L	1218

10.297.2.4 LLI . . . . .	1218
10.297.2.5 P . . . . .	1218
10.297.2.6 rcv . . . . .	1218
10.297.2.7 sat . . . . .	1218
10.297.2.8 SNR . . . . .	1218
10.297.2.9 time . . . . .	1219
10.298 ObservablesInterface Class Reference . . . . .	1219
10.298.1 Detailed Description . . . . .	1221
10.299 opt_t Struct Reference . . . . .	1222
10.299.1 Detailed Description . . . . .	1222
10.299.2 Member Data Documentation . . . . .	1222
10.299.2.1 comment . . . . .	1222
10.299.2.2 format . . . . .	1222
10.299.2.3 name . . . . .	1222
10.299.2.4 var . . . . .	1222
10.300 OsmosdrSignalSource Class Reference . . . . .	1223
10.300.1 Detailed Description . . . . .	1225
10.300.2 Member Function Documentation . . . . .	1225
10.300.2.1 connect() . . . . .	1225
10.300.2.2 disconnect() . . . . .	1225
10.300.2.3 get_left_block() . . . . .	1225
10.300.2.4 get_right_block() . . . . .	1225
10.300.2.5 item_size() . . . . .	1226
10.301 OSNMA_data Class Reference . . . . .	1226
10.301.1 Detailed Description . . . . .	1227
10.301.2 Member Data Documentation . . . . .	1227
10.301.2.1 d_dsm_header . . . . .	1227
10.301.2.2 d_dsm_kroot_message . . . . .	1227
10.301.2.3 d_dsm_kroot_new_message . . . . .	1227
10.301.2.4 d_dsm_pkr_message . . . . .	1227
10.301.2.5 d_mack_message . . . . .	1227
10.301.2.6 d_nav_data . . . . .	1227
10.301.2.7 d_nma_header . . . . .	1227
10.302 OSNMA_DSM_Reader Class Reference . . . . .	1228
10.302.1 Detailed Description . . . . .	1229
10.303 Osnma_Helper Class Reference . . . . .	1229
10.303.1 Detailed Description . . . . .	1230
10.303.2 Member Data Documentation . . . . .	1230
10.303.2.1 GST_START_EPOCH . . . . .	1230
10.304 OSNMA_msg Class Reference . . . . .	1230
10.304.1 Detailed Description . . . . .	1231
10.304.2 Member Data Documentation . . . . .	1231

10.304.2.1 hkroot . . . . .	1231
10.304.2.2 mack . . . . .	1231
10.304.2.3 PRN . . . . .	1231
10.304.2.4 TOW_sf0 . . . . .	1231
10.304.2.5 WN_sf0 . . . . .	1231
10.305 osnma_msg_receiver Class Reference . . . . .	1231
10.305.1 Detailed Description . . . . .	1233
10.305.2 Constructor & Destructor Documentation . . . . .	1233
10.305.2.1 ~osnma_msg_receiver() . . . . .	1233
10.305.3 Member Function Documentation . . . . .	1233
10.305.3.1 msg_handler_osnma() . . . . .	1233
10.305.3.2 read_merkle_xml() . . . . .	1233
10.305.3.3 set_merkle_root() . . . . .	1233
10.305.3.4 verify_dsm_pkr() . . . . .	1233
10.306 OSNMA_NavData Class Reference . . . . .	1234
10.306.1 Detailed Description . . . . .	1234
10.306.2 Constructor & Destructor Documentation . . . . .	1235
10.306.2.1 OSNMA_NavData() . . . . .	1235
10.306.3 Member Function Documentation . . . . .	1235
10.306.3.1 get_IOD_nav() . . . . .	1235
10.306.3.2 get_last_received_TOW() . . . . .	1235
10.306.3.3 get_prn_d() . . . . .	1235
10.306.3.4 get_tow_sf0() . . . . .	1235
10.306.3.5 get_verified_bits() . . . . .	1235
10.306.3.6 get_verified_status() . . . . .	1235
10.306.3.7 set_ephemeris_data() . . . . .	1235
10.306.3.8 set_IOD_nav() . . . . .	1235
10.306.3.9 set_last_received_TOW() . . . . .	1235
10.306.3.10 set_prn_d() . . . . .	1236
10.306.3.11 set_tow_sf0() . . . . .	1236
10.306.3.12 set_update_verified_bits() . . . . .	1236
10.306.3.13 set_utc_data() . . . . .	1236
10.306.3.14 set_verified_status() . . . . .	1236
10.306.4 Member Data Documentation . . . . .	1236
10.306.4.1 nav_data_id . . . . .	1236
10.307 OSNMA_NavDataManager Class Reference . . . . .	1236
10.307.1 Detailed Description . . . . .	1237
10.308 Pass_Through Class Reference . . . . .	1237
10.308.1 Detailed Description . . . . .	1240
10.308.2 Member Function Documentation . . . . .	1240
10.308.2.1 connect() . . . . .	1240
10.308.2.2 disconnect() . . . . .	1240

10.308.2.3	<a href="#">get_left_block()</a>	1240
10.308.2.4	<a href="#">get_right_block()</a>	1240
10.308.2.5	<a href="#">implementation()</a>	1240
10.308.2.6	<a href="#">item_size()</a>	1240
10.308.2.7	<a href="#">item_type()</a>	1240
10.308.2.8	<a href="#">role()</a>	1240
10.309	<a href="#">pcl_k_t Struct Reference</a>	1241
10.309.1	<a href="#">Detailed Description</a>	1241
10.309.2	<a href="#">Member Data Documentation</a>	1241
10.309.2.1	<a href="#">clk</a>	1241
10.309.2.2	<a href="#">index</a>	1241
10.309.2.3	<a href="#">std</a>	1241
10.309.2.4	<a href="#">time</a>	1242
10.310	<a href="#">pcps_acquisition Class Reference</a>	1242
10.310.1	<a href="#">Detailed Description</a>	1244
10.310.2	<a href="#">Member Function Documentation</a>	1244
10.310.2.1	<a href="#">general_work()</a>	1244
10.310.2.2	<a href="#">init()</a>	1244
10.310.2.3	<a href="#">mag()</a>	1244
10.310.2.4	<a href="#">set_active()</a>	1244
10.310.2.5	<a href="#">set_channel()</a>	1245
10.310.2.6	<a href="#">set_channel_fsm()</a>	1245
10.310.2.7	<a href="#">set_doppler_center()</a>	1245
10.310.2.8	<a href="#">set_doppler_max()</a>	1245
10.310.2.9	<a href="#">set_doppler_step()</a>	1245
10.310.2.10	<a href="#">set_gnss_synchro()</a>	1245
10.310.2.11	<a href="#">set_local_code()</a>	1246
10.310.2.12	<a href="#">set_state()</a>	1246
10.310.2.13	<a href="#">set_threshold()</a>	1246
10.311	<a href="#">pcps_acquisition_fine_doppler_cc Class Reference</a>	1246
10.311.1	<a href="#">Detailed Description</a>	1249
10.311.2	<a href="#">Constructor &amp; Destructor Documentation</a>	1249
10.311.2.1	<a href="#">~pcps_acquisition_fine_doppler_cc()</a>	1249
10.311.3	<a href="#">Member Function Documentation</a>	1249
10.311.3.1	<a href="#">general_work()</a>	1249
10.311.3.2	<a href="#">init()</a>	1249
10.311.3.3	<a href="#">mag()</a>	1249
10.311.3.4	<a href="#">nextPowerOf2()</a>	1250
10.311.3.5	<a href="#">set_active()</a>	1251
10.311.3.6	<a href="#">set_channel()</a>	1251
10.311.3.7	<a href="#">set_channel_fsm()</a>	1251
10.311.3.8	<a href="#">set_doppler_max()</a>	1251

10.311.3.9 set_doppler_step()	1251
10.311.3.10 set_gnss_synchro()	1252
10.311.3.11 set_local_code()	1252
10.311.3.12 set_state()	1252
10.311.3.13 set_threshold()	1252
10.312 pcps_acquisition_fpga Class Reference	1252
10.312.1 Detailed Description	1254
10.312.2 Constructor & Destructor Documentation	1254
10.312.2.1 ~pcps_acquisition_fpga()	1254
10.312.3 Member Function Documentation	1254
10.312.3.1 init()	1254
10.312.3.2 mag()	1254
10.312.3.3 reset_acquisition()	1254
10.312.3.4 set_active()	1254
10.312.3.5 set_channel()	1255
10.312.3.6 set_channel_fsm()	1255
10.312.3.7 set_doppler_center()	1255
10.312.3.8 set_doppler_max()	1255
10.312.3.9 set_doppler_step()	1255
10.312.3.10 set_gnss_synchro()	1255
10.312.3.11 set_local_code()	1256
10.312.3.12 set_state()	1256
10.312.3.13 set_threshold()	1256
10.312.3.14 stop_acquisition()	1256
10.313 pcps_assisted_acquisition_cc Class Reference	1256
10.313.1 Detailed Description	1259
10.313.2 Constructor & Destructor Documentation	1259
10.313.2.1 ~pcps_assisted_acquisition_cc()	1259
10.313.3 Member Function Documentation	1259
10.313.3.1 general_work()	1259
10.313.3.2 init()	1259
10.313.3.3 mag()	1259
10.313.3.4 set_active()	1260
10.313.3.5 set_channel()	1260
10.313.3.6 set_channel_fsm()	1260
10.313.3.7 set_doppler_max()	1260
10.313.3.8 set_doppler_step()	1260
10.313.3.9 set_gnss_synchro()	1260
10.313.3.10 set_local_code()	1261
10.313.3.11 set_state()	1261
10.313.3.12 set_threshold()	1261
10.314 pcps_cccwsr_acquisition_cc Class Reference	1261

10.314.1 Detailed Description . . . . .	1264
10.314.2 Constructor & Destructor Documentation . . . . .	1264
10.314.2.1 ~pcps_ccwsr_acquisition_cc() . . . . .	1264
10.314.3 Member Function Documentation . . . . .	1264
10.314.3.1 general_work() . . . . .	1264
10.314.3.2 init() . . . . .	1264
10.314.3.3 mag() . . . . .	1264
10.314.3.4 set_active() . . . . .	1264
10.314.3.5 set_channel() . . . . .	1265
10.314.3.6 set_channel_fsm() . . . . .	1265
10.314.3.7 set_doppler_max() . . . . .	1265
10.314.3.8 set_doppler_step() . . . . .	1265
10.314.3.9 set_gnss_synchro() . . . . .	1265
10.314.3.10 set_local_code() . . . . .	1266
10.314.3.11 set_state() . . . . .	1266
10.314.3.12 set_threshold() . . . . .	1266
10.315 pcps_openc1_acquisition_cc Class Reference . . . . .	1266
10.315.1 Detailed Description . . . . .	1269
10.315.2 Constructor & Destructor Documentation . . . . .	1269
10.315.2.1 ~pcps_openc1_acquisition_cc() . . . . .	1269
10.315.3 Member Function Documentation . . . . .	1269
10.315.3.1 general_work() . . . . .	1269
10.315.3.2 init() . . . . .	1269
10.315.3.3 mag() . . . . .	1269
10.315.3.4 openc1_ready() . . . . .	1269
10.315.3.5 set_active() . . . . .	1270
10.315.3.6 set_channel() . . . . .	1270
10.315.3.7 set_channel_fsm() . . . . .	1270
10.315.3.8 set_doppler_max() . . . . .	1270
10.315.3.9 set_doppler_step() . . . . .	1270
10.315.3.10 set_gnss_synchro() . . . . .	1271
10.315.3.11 set_local_code() . . . . .	1272
10.315.3.12 set_state() . . . . .	1272
10.315.3.13 set_threshold() . . . . .	1272
10.316 pcps_quicksync_acquisition_cc Class Reference . . . . .	1272
10.316.1 Detailed Description . . . . .	1275
10.316.2 Constructor & Destructor Documentation . . . . .	1275
10.316.2.1 ~pcps_quicksync_acquisition_cc() . . . . .	1275
10.316.3 Member Function Documentation . . . . .	1275
10.316.3.1 general_work() . . . . .	1275
10.316.3.2 init() . . . . .	1275
10.316.3.3 mag() . . . . .	1275



10.316.3.4 set_active()	1276
10.316.3.5 set_channel()	1276
10.316.3.6 set_channel_fsm()	1276
10.316.3.7 set_doppler_max()	1276
10.316.3.8 set_doppler_step()	1276
10.316.3.9 set_gnss_synchro()	1277
10.316.3.10 set_local_code()	1278
10.316.3.11 set_state()	1278
10.316.3.12 set_threshold()	1278
10.317 pcps_tong_acquisition_cc Class Reference	1278
10.317.1 Detailed Description	1281
10.317.2 Constructor & Destructor Documentation	1281
10.317.2.1 ~pcps_tong_acquisition_cc()	1281
10.317.3 Member Function Documentation	1281
10.317.3.1 general_work()	1281
10.317.3.2 init()	1281
10.317.3.3 mag()	1281
10.317.3.4 set_active()	1281
10.317.3.5 set_channel()	1282
10.317.3.6 set_channel_fsm()	1282
10.317.3.7 set_doppler_max()	1282
10.317.3.8 set_doppler_step()	1282
10.317.3.9 set_gnss_synchro()	1282
10.317.3.10 set_local_code()	1283
10.317.3.11 set_state()	1283
10.317.3.12 set_threshold()	1283
10.318 pcv_t Struct Reference	1284
10.318.1 Detailed Description	1284
10.318.2 Member Data Documentation	1284
10.318.2.1 code	1284
10.318.2.2 off	1284
10.318.2.3 sat	1285
10.318.2.4 te	1285
10.318.2.5 ts	1285
10.318.2.6 type	1285
10.318.2.7 var	1285
10.319 pcvs_t Struct Reference	1286
10.319.1 Detailed Description	1286
10.319.2 Member Data Documentation	1286
10.319.2.1 n	1286
10.319.2.2 nmax	1287
10.319.2.3 pcv	1287

10.320 peph_t Struct Reference . . . . .	1287
10.320.1 Detailed Description . . . . .	1288
10.320.2 Member Data Documentation . . . . .	1288
10.320.2.1 cov . . . . .	1288
10.320.2.2 index . . . . .	1288
10.320.2.3 pos . . . . .	1288
10.320.2.4 std . . . . .	1288
10.320.2.5 time . . . . .	1288
10.320.2.6 vco . . . . .	1288
10.320.2.7 vel . . . . .	1288
10.320.2.8 vst . . . . .	1288
10.321 PlutosdrSignalSource Class Reference . . . . .	1289
10.321.1 Detailed Description . . . . .	1291
10.321.2 Member Function Documentation . . . . .	1291
10.321.2.1 connect() . . . . .	1291
10.321.2.2 disconnect() . . . . .	1291
10.321.2.3 get_left_block() . . . . .	1291
10.321.2.4 get_right_block() . . . . .	1291
10.321.2.5 item_size() . . . . .	1291
10.322 pppcorr_t Struct Reference . . . . .	1292
10.322.1 Detailed Description . . . . .	1293
10.322.2 Member Data Documentation . . . . .	1293
10.322.2.1 ns . . . . .	1293
10.322.2.2 nsmax . . . . .	1293
10.322.2.3 nsta . . . . .	1293
10.322.2.4 nt . . . . .	1293
10.322.2.5 ntmax . . . . .	1293
10.322.2.6 rr . . . . .	1293
10.322.2.7 stas . . . . .	1293
10.322.2.8 stec . . . . .	1293
10.322.2.9 trop . . . . .	1293
10.323 pps_tcp_rx Class Reference . . . . .	1294
10.323.1 Detailed Description . . . . .	1294
10.323.2 Member Data Documentation . . . . .	1294
10.323.2.1 is_connected . . . . .	1294
10.324 PpsSamplestamp Class Reference . . . . .	1295
10.324.1 Detailed Description . . . . .	1295
10.324.2 Member Data Documentation . . . . .	1295
10.324.2.1 overflow_reg . . . . .	1295
10.324.2.2 samplestamp . . . . .	1295
10.325 prcopt_t Struct Reference . . . . .	1296
10.325.1 Detailed Description . . . . .	1297

10.325.2 Member Data Documentation	1298
10.325.2.1 antdel	1298
10.325.2.2 anttype	1298
10.325.2.3 armaxiter	1298
10.325.2.4 bancroft_init	1298
10.325.2.5 baseline	1298
10.325.2.6 bdsmodear	1298
10.325.2.7 codesmooth	1298
10.325.2.8 dynamics	1298
10.325.2.9 elmaskar	1298
10.325.2.10 elmaskhold	1298
10.325.2.11 elmin	1298
10.325.2.12 eratio	1298
10.325.2.13 err	1299
10.325.2.14 exsats	1299
10.325.2.15 exterr	1299
10.325.2.16 freqopt	1299
10.325.2.17 glomodear	1299
10.325.2.18 initrst	1299
10.325.2.19 intpref	1299
10.325.2.20 ionoopt	1299
10.325.2.21 maxaveep	1299
10.325.2.22 maxgdop	1299
10.325.2.23 maxinno	1299
10.325.2.24 maxout	1299
10.325.2.25 maxtdiff	1300
10.325.2.26 minfix	1300
10.325.2.27 minlock	1300
10.325.2.28 mode	1300
10.325.2.29 modear	1300
10.325.2.30 navsys	1300
10.325.2.31 nf	1300
10.325.2.32 niter	1300
10.325.2.33 odisp	1300
10.325.2.34 outsingle	1300
10.325.2.35 pcvr	1300
10.325.2.36 posopt	1300
10.325.2.37 pppopt	1301
10.325.2.38 prn	1301
10.325.2.39 rb	1301
10.325.2.40 refpos	1301
10.325.2.41 rnsopt	1301

10.325.2.42	rovpos	1301
10.325.2.43	ru	1301
10.325.2.44	sateph	1301
10.325.2.45	sbascorr	1301
10.325.2.46	sbassatsel	1301
10.325.2.47	sclkstab	1301
10.325.2.48	snrmask	1301
10.325.2.49	soltype	1302
10.325.2.50	std	1302
10.325.2.51	syncsol	1302
10.325.2.52	thresar	1302
10.325.2.53	thresslip	1302
10.325.2.54	tidecorr	1302
10.325.2.55	tropopt	1302
10.326	pulse_blanking_cc Class Reference	1303
10.326.1	Detailed Description	1304
10.327	PulseBlankingFilter Class Reference	1304
10.327.1	Detailed Description	1306
10.327.2	Member Function Documentation	1306
10.327.2.1	connect()	1306
10.327.2.2	disconnect()	1306
10.327.2.3	get_left_block()	1306
10.327.2.4	get_right_block()	1306
10.327.2.5	implementation()	1306
10.327.2.6	item_size()	1306
10.327.2.7	role()	1306
10.328	Pvt_Conf Class Reference	1307
10.328.1	Detailed Description	1309
10.328.2	Member Data Documentation	1309
10.328.2.1	an_dump_devname	1309
10.328.2.2	an_output_enabled	1309
10.328.2.3	an_rate_ms	1309
10.328.2.4	display_rate_ms	1309
10.328.2.5	dump	1309
10.328.2.6	dump_filename	1309
10.328.2.7	dump_mat	1309
10.328.2.8	enable_pvt_kf	1309
10.328.2.9	enable_rx_clock_correction	1309
10.328.2.10	flag_nmea_tty_port	1310
10.328.2.11	flag_rtcn_server	1310
10.328.2.12	flag_rtcn_tty_port	1310
10.328.2.13	geojson_output_enabled	1310

10.328.2.14 geojson_output_path . . . . .	1310
10.328.2.15 geojson_rate_ms . . . . .	1310
10.328.2.16 gpx_output_enabled . . . . .	1310
10.328.2.17 gpx_output_path . . . . .	1310
10.328.2.18 gpx_rate_ms . . . . .	1310
10.328.2.19 has_output_file_path . . . . .	1310
10.328.2.20 kml_output_enabled . . . . .	1310
10.328.2.21 kml_output_path . . . . .	1310
10.328.2.22 kml_rate_ms . . . . .	1311
10.328.2.23 log_source_timetag . . . . .	1311
10.328.2.24 log_source_timetag_file . . . . .	1311
10.328.2.25 max_obs_block_rx_clock_offset_ms . . . . .	1311
10.328.2.26 measures_ecef_pos_sd_m . . . . .	1311
10.328.2.27 measures_ecef_vel_sd_ms . . . . .	1311
10.328.2.28 monitor_enabled . . . . .	1311
10.328.2.29 monitor_ephemeris_enabled . . . . .	1311
10.328.2.30 nmea_dump_devname . . . . .	1311
10.328.2.31 nmea_dump_filename . . . . .	1311
10.328.2.32 nmea_output_file_enabled . . . . .	1311
10.328.2.33 nmea_output_file_path . . . . .	1311
10.328.2.34 nmea_rate_ms . . . . .	1312
10.328.2.35 observable_interval_ms . . . . .	1312
10.328.2.36 osnma_strict . . . . .	1312
10.328.2.37 output_enabled . . . . .	1312
10.328.2.38 output_path . . . . .	1312
10.328.2.39 output_rate_ms . . . . .	1312
10.328.2.40 pre_2009_file . . . . .	1312
10.328.2.41 protobuf_enabled . . . . .	1312
10.328.2.42 rinex_name . . . . .	1312
10.328.2.43 rinex_output_enabled . . . . .	1312
10.328.2.44 rinex_output_path . . . . .	1312
10.328.2.45 rinex_version . . . . .	1312
10.328.2.46 rinexobs_rate_ms . . . . .	1313
10.328.2.47 rtdcm_dump_devname . . . . .	1313
10.328.2.48 rtdcm_msg_rate_ms . . . . .	1313
10.328.2.49 rtdcm_output_file_enabled . . . . .	1313
10.328.2.50 rtdcm_output_file_path . . . . .	1313
10.328.2.51 rtdcm_station_id . . . . .	1313
10.328.2.52 rtdcm_tcp_port . . . . .	1313
10.328.2.53 rtk_trace_level . . . . .	1313
10.328.2.54 show_local_time_zone . . . . .	1313
10.328.2.55 system_ecef_pos_sd_m . . . . .	1313

10.328.2.56	<a href="#">system_ecef_vel_sd_ms</a>	1313
10.328.2.57	<a href="#">type_of_receiver</a>	1313
10.328.2.58	<a href="#">udp_addresses</a>	1314
10.328.2.59	<a href="#">udp_eph_addresses</a>	1314
10.328.2.60	<a href="#">udp_eph_port</a>	1314
10.328.2.61	<a href="#">udp_ports</a>	1314
10.328.2.62	<a href="#">use_e6_for_pvt</a>	1314
10.328.2.63	<a href="#">use_has_corrections</a>	1314
10.328.2.64	<a href="#">use_unhealthy_sats</a>	1314
10.328.2.65	<a href="#">xml_output_enabled</a>	1314
10.328.2.66	<a href="#">xml_output_path</a>	1314
10.329	<a href="#">Pvt_Kf Class Reference</a>	1314
10.329.1	<a href="#">Detailed Description</a>	1315
10.330	<a href="#">Pvt_Solution Class Reference</a>	1315
10.330.1	<a href="#">Detailed Description</a>	1318
10.330.2	<a href="#">Member Function Documentation</a>	1318
10.330.2.1	<a href="#">get_clock_drift_ppm()</a>	1318
10.330.2.2	<a href="#">get_course_over_ground()</a>	1318
10.330.2.3	<a href="#">get_height()</a>	1318
10.330.2.4	<a href="#">get_latitude()</a>	1318
10.330.2.5	<a href="#">get_longitude()</a>	1318
10.330.2.6	<a href="#">get_num_valid_observations()</a>	1318
10.330.2.7	<a href="#">get_speed_over_ground()</a>	1319
10.330.2.8	<a href="#">get_time_offset_s()</a>	1319
10.330.2.9	<a href="#">set_clock_drift_ppm()</a>	1319
10.330.2.10	<a href="#">set_course_over_ground()</a>	1319
10.330.2.11	<a href="#">set_num_valid_observations()</a>	1319
10.330.2.12	<a href="#">set_pre_2009_file()</a>	1319
10.330.2.13	<a href="#">set_rx_pos()</a>	1319
10.330.2.14	<a href="#">set_rx_vel()</a>	1319
10.330.2.15	<a href="#">set_speed_over_ground()</a>	1319
10.330.2.16	<a href="#">set_time_offset_s()</a>	1319
10.331	<a href="#">PvtInterface Class Reference</a>	1320
10.331.1	<a href="#">Detailed Description</a>	1322
10.332	<a href="#">raw_t Struct Reference</a>	1323
10.332.1	<a href="#">Detailed Description</a>	1324
10.332.2	<a href="#">Member Data Documentation</a>	1324
10.332.2.1	<a href="#">buff</a>	1324
10.332.2.2	<a href="#">dpCA</a>	1324
10.332.2.3	<a href="#">ephsat</a>	1324
10.332.2.4	<a href="#">flag</a>	1324
10.332.2.5	<a href="#">freqn</a>	1325

10.332.2.6 halfc	1325
10.332.2.7 icpc	1325
10.332.2.8 icpp	1325
10.332.2.9 iod	1325
10.332.2.10 len	1325
10.332.2.11 lexmsg	1325
10.332.2.12 lockt	1325
10.332.2.13 msgtype	1325
10.332.2.14 nav	1325
10.332.2.15 nbyte	1325
10.332.2.16 obs	1325
10.332.2.17 obuf	1326
10.332.2.18 off	1326
10.332.2.19 opt	1326
10.332.2.20 outtype	1326
10.332.2.21 page	1326
10.332.2.22 pbuff	1326
10.332.2.23 pbyte	1326
10.332.2.24 plen	1326
10.332.2.25 prCA	1326
10.332.2.26 receive_time	1326
10.332.2.27 reply	1326
10.332.2.28 sbsmsg	1326
10.332.2.29 sta	1327
10.332.2.30 subfrm	1327
10.332.2.31 tbase	1327
10.332.2.32 time	1327
10.332.2.33 tobs	1327
10.332.2.34 tod	1327
10.332.2.35 week	1327
10.333 RawArraySignalSource Class Reference	1327
10.333.1 Detailed Description	1330
10.333.2 Member Function Documentation	1330
10.333.2.1 connect()	1330
10.333.2.2 disconnect()	1330
10.333.2.3 get_left_block()	1330
10.333.2.4 get_right_block()	1330
10.333.2.5 item_size()	1330
10.334 ReedSolomon Class Reference	1331
10.334.1 Detailed Description	1331
10.334.2 Constructor & Destructor Documentation	1331
10.334.2.1 ReedSolomon() [1/2]	1331

10.334.2.2 ReedSolomon() [2/2]	1332
10.334.3 Member Function Documentation	1332
10.334.3.1 decode()	1332
10.334.3.2 encode_with_generator_matrix()	1332
10.334.3.3 encode_with_generator_poly()	1332
10.335 Rinex_Printer Class Reference	1333
10.335.1 Detailed Description	1334
10.335.2 Constructor & Destructor Documentation	1334
10.335.2.1 Rinex_Printer()	1334
10.335.2.2 ~Rinex_Printer()	1334
10.335.3 Member Function Documentation	1334
10.335.3.1 get_navfilename()	1334
10.335.3.2 get_obsfilename()	1334
10.335.3.3 is_rinex_header_written()	1334
10.335.3.4 log_rinex_nav_bds_dnav()	1334
10.335.3.5 log_rinex_nav_gal_nav()	1335
10.335.3.6 log_rinex_nav_glo_gnav()	1335
10.335.3.7 log_rinex_nav_gps_cnav()	1335
10.335.3.8 log_rinex_nav_gps_nav()	1335
10.335.3.9 print_rinex_annotation()	1335
10.335.3.10 set_pre_2009_file()	1336
10.336 RtcM Class Reference	1337
10.336.1 Detailed Description	1340
10.336.2 Constructor & Destructor Documentation	1340
10.336.2.1 RtcM()	1340
10.336.3 Member Function Documentation	1341
10.336.3.1 bin_to_binary_data()	1341
10.336.3.2 bin_to_double()	1341
10.336.3.3 bin_to_hex()	1341
10.336.3.4 bin_to_uint()	1341
10.336.3.5 binary_data_to_bin()	1341
10.336.3.6 check_CRC()	1341
10.336.3.7 hex_to_bin()	1341
10.336.3.8 hex_to_int()	1341
10.336.3.9 hex_to_uint()	1341
10.336.3.10 is_server_running()	1341
10.336.3.11 lock_time() [1/4]	1342
10.336.3.12 lock_time() [2/4]	1342
10.336.3.13 lock_time() [3/4]	1342
10.336.3.14 lock_time() [4/4]	1342
10.336.3.15 print_IGM01()	1342
10.336.3.16 print_IGM02()	1342



10.336.3.17 print_IGM03()	1343
10.336.3.18 print_IGM05()	1343
10.336.3.19 print_MSM_1()	1343
10.336.3.20 print_MSM_2()	1343
10.336.3.21 print_MSM_3()	1343
10.336.3.22 print_MSM_4()	1344
10.336.3.23 print_MSM_5()	1344
10.336.3.24 print_MSM_6()	1344
10.336.3.25 print_MSM_7()	1344
10.336.3.26 print_MT1001()	1345
10.336.3.27 print_MT1002()	1345
10.336.3.28 print_MT1003()	1345
10.336.3.29 print_MT1004()	1345
10.336.3.30 print_MT1005()	1345
10.336.3.31 print_MT1005_test()	1346
10.336.3.32 print_MT1006()	1346
10.336.3.33 print_MT1008()	1346
10.336.3.34 print_MT1009()	1346
10.336.3.35 print_MT1010()	1347
10.336.3.36 print_MT1011()	1347
10.336.3.37 print_MT1012()	1348
10.336.3.38 print_MT1019()	1348
10.336.3.39 print_MT1020()	1348
10.336.3.40 print_MT1029()	1348
10.336.3.41 print_MT1045()	1349
10.336.3.42 read_MT1005()	1349
10.336.3.43 read_MT1019()	1349
10.336.3.44 read_MT1020()	1349
10.336.3.45 read_MT1045()	1349
10.336.3.46 run_server()	1349
10.336.3.47 send_message()	1350
10.336.3.48 stop_server()	1350
10.337 RtcM_Printer Class Reference	1350
10.337.1 Detailed Description	1351
10.337.2 Constructor & Destructor Documentation	1351
10.337.2.1 RtcM_Printer()	1351
10.337.2.2 ~RtcM_Printer()	1351
10.337.3 Member Function Documentation	1351
10.337.3.1 lock_time()	1351
10.337.3.2 print_MT1005_test()	1351
10.337.3.3 Print_RtcM_Messages()	1352
10.338 rtcM_t Struct Reference	1353

10.338.1 Detailed Description . . . . .	1354
10.338.2 Member Data Documentation . . . . .	1354
10.338.2.1 buff . . . . .	1354
10.338.2.2 cp . . . . .	1354
10.338.2.3 dgps . . . . .	1354
10.338.2.4 ephsat . . . . .	1354
10.338.2.5 len . . . . .	1354
10.338.2.6 lltime . . . . .	1354
10.338.2.7 lock . . . . .	1355
10.338.2.8 loss . . . . .	1355
10.338.2.9 msg . . . . .	1355
10.338.2.10 msgtype . . . . .	1355
10.338.2.11 msmttype . . . . .	1355
10.338.2.12 nav . . . . .	1355
10.338.2.13 nbit . . . . .	1355
10.338.2.14 nbyte . . . . .	1355
10.338.2.15 nmsg2 . . . . .	1355
10.338.2.16 nmsg3 . . . . .	1355
10.338.2.17 obs . . . . .	1355
10.338.2.18 obsflag . . . . .	1355
10.338.2.19 opt . . . . .	1356
10.338.2.20 outtype . . . . .	1356
10.338.2.21 seqno . . . . .	1356
10.338.2.22 ssr . . . . .	1356
10.338.2.23 sta . . . . .	1356
10.338.2.24 stah . . . . .	1356
10.338.2.25 staid . . . . .	1356
10.338.2.26 time . . . . .	1356
10.338.2.27 time_s . . . . .	1356
10.338.2.28 word . . . . .	1356
10.339 rtk_t Struct Reference . . . . .	1357
10.339.1 Detailed Description . . . . .	1358
10.339.2 Member Data Documentation . . . . .	1358
10.339.2.1 ambc . . . . .	1358
10.339.2.2 errbuf . . . . .	1358
10.339.2.3 na . . . . .	1358
10.339.2.4 neb . . . . .	1358
10.339.2.5 nfix . . . . .	1358
10.339.2.6 nx . . . . .	1358
10.339.2.7 opt . . . . .	1358
10.339.2.8 P . . . . .	1358
10.339.2.9 Pa . . . . .	1359

10.339.2.10 rb . . . . .	1359
10.339.2.11 sol . . . . .	1359
10.339.2.12 ssat . . . . .	1359
10.339.2.13 tt . . . . .	1359
10.339.2.14 x . . . . .	1359
10.339.2.15 xa . . . . .	1359
10.340 Rtklib_Pvt Class Reference . . . . .	1359
10.340.1 Detailed Description . . . . .	1362
10.340.2 Member Function Documentation . . . . .	1363
10.340.2.1 clear_ephemeris() . . . . .	1363
10.340.2.2 connect() . . . . .	1363
10.340.2.3 disconnect() . . . . .	1363
10.340.2.4 get_galileo_almanac() . . . . .	1363
10.340.2.5 get_galileo_ephemeris() . . . . .	1363
10.340.2.6 get_gps_almanac() . . . . .	1363
10.340.2.7 get_gps_ephemeris() . . . . .	1364
10.340.2.8 get_latest_PVT() . . . . .	1364
10.340.2.9 get_left_block() . . . . .	1364
10.340.2.10 get_right_block() . . . . .	1364
10.340.2.11 implementation() . . . . .	1364
10.340.2.12 item_size() . . . . .	1364
10.340.2.13 reset() . . . . .	1364
10.340.2.14 role() . . . . .	1364
10.341 rtklib_pvt_gs Class Reference . . . . .	1365
10.341.1 Detailed Description . . . . .	1367
10.341.2 Constructor & Destructor Documentation . . . . .	1367
10.341.2.1 ~rtklib_pvt_gs() . . . . .	1367
10.341.3 Member Function Documentation . . . . .	1367
10.341.3.1 clear_ephemeris() . . . . .	1367
10.341.3.2 get_beidou_dnav_almanac_map() . . . . .	1367
10.341.3.3 get_beidou_dnav_ephemeris_map() . . . . .	1367
10.341.3.4 get_galileo_almanac_map() . . . . .	1367
10.341.3.5 get_galileo_ephemeris_map() . . . . .	1367
10.341.3.6 get_gps_almanac_map() . . . . .	1367
10.341.3.7 get_gps_ephemeris_map() . . . . .	1367
10.341.3.8 get_latest_PVT() . . . . .	1368
10.341.3.9 work() . . . . .	1368
10.342 Rtklib_Solver Class Reference . . . . .	1368
10.342.1 Detailed Description . . . . .	1372
10.342.2 Member Function Documentation . . . . .	1372
10.342.2.1 get_gdop() . . . . .	1372
10.342.2.2 get_hdop() . . . . .	1372

10.342.2.3	<a href="#">get_pdop()</a>	1372
10.342.2.4	<a href="#">get_vdop()</a>	1372
10.342.3	Member Data Documentation	1372
10.342.3.1	<a href="#">beidou_dnav_almanac_map</a>	1372
10.342.3.2	<a href="#">beidou_dnav_ephemeris_map</a>	1372
10.342.3.3	<a href="#">beidou_dnav_iono</a>	1372
10.342.3.4	<a href="#">beidou_dnav_utc_model</a>	1372
10.342.3.5	<a href="#">galileo_almanac_map</a>	1372
10.342.3.6	<a href="#">galileo_ephemeris_map</a>	1373
10.342.3.7	<a href="#">galileo_iono</a>	1373
10.342.3.8	<a href="#">galileo_utc_model</a>	1373
10.342.3.9	<a href="#">glonass_gnav_almanac</a>	1373
10.342.3.10	<a href="#">glonass_gnav_ephemeris_map</a>	1373
10.342.3.11	<a href="#">glonass_gnav_utc_model</a>	1373
10.342.3.12	<a href="#">gps_almanac_map</a>	1373
10.342.3.13	<a href="#">gps_cnav_ephemeris_map</a>	1373
10.342.3.14	<a href="#">gps_cnav_iono</a>	1373
10.342.3.15	<a href="#">gps_cnav_utc_model</a>	1373
10.342.3.16	<a href="#">gps_ephemeris_map</a>	1373
10.342.3.17	<a href="#">gps_iono</a>	1374
10.342.3.18	<a href="#">gps_utc_model</a>	1374
10.342.3.19	<a href="#">pvt_sol</a>	1374
10.342.3.20	<a href="#">pvt_ssat</a>	1374
10.343	<a href="#">rtksvr_t</a> Struct Reference	1375
10.343.1	Detailed Description	1376
10.343.2	Member Data Documentation	1376
10.343.2.1	<a href="#">buff</a>	1376
10.343.2.2	<a href="#">buffsize</a>	1376
10.343.2.3	<a href="#">cputime</a>	1376
10.343.2.4	<a href="#">cycle</a>	1376
10.343.2.5	<a href="#">files</a>	1376
10.343.2.6	<a href="#">format</a>	1376
10.343.2.7	<a href="#">ftime</a>	1377
10.343.2.8	<a href="#">lock</a>	1377
10.343.2.9	<a href="#">moni</a>	1377
10.343.2.10	<a href="#">nav</a>	1377
10.343.2.11	<a href="#">navsel</a>	1377
10.343.2.12	<a href="#">nb</a>	1377
10.343.2.13	<a href="#">nmeacycle</a>	1377
10.343.2.14	<a href="#">nmeapos</a>	1377
10.343.2.15	<a href="#">nmeareq</a>	1377
10.343.2.16	<a href="#">nmsg</a>	1377

10.343.2.17 npb	1377
10.343.2.18 nsb	1377
10.343.2.19 nsbs	1378
10.343.2.20 nsol	1378
10.343.2.21 obs	1378
10.343.2.22 pbuf	1378
10.343.2.23 prcout	1378
10.343.2.24 raw	1378
10.343.2.25 rtcn	1378
10.343.2.26 rtk	1378
10.343.2.27 sbsmsg	1378
10.343.2.28 sbuf	1378
10.343.2.29 solbuf	1378
10.343.2.30 solopt	1378
10.343.2.31 state	1379
10.343.2.32 stream	1379
10.343.2.33 thread	1379
10.343.2.34 tick	1379
10.344 Rtl_Tcp_Dongle_Info Class Reference	1379
10.344.1 Detailed Description	1380
10.344.2 Member Enumeration Documentation	1380
10.344.2.1 anonymous enum	1380
10.344.3 Member Function Documentation	1380
10.344.3.1 get_tuner_gain_count()	1380
10.344.3.2 get_tuner_type()	1380
10.345 rtl_tcp_signal_source_c Class Reference	1380
10.345.1 Detailed Description	1382
10.346 RtlTcpSignalSource Class Reference	1382
10.346.1 Detailed Description	1385
10.346.2 Member Function Documentation	1385
10.346.2.1 connect()	1385
10.346.2.2 disconnect()	1385
10.346.2.3 get_left_block()	1385
10.346.2.4 get_right_block()	1385
10.346.2.5 item_size()	1386
10.347 Sbas_Ephemeris Class Reference	1386
10.347.1 Detailed Description	1387
10.347.2 Member Data Documentation	1387
10.347.2.1 b_sv_do_not_use	1387
10.347.2.2 d_acc	1387
10.347.2.3 d_af0	1387
10.347.2.4 d_af1	1387

10.347.2.5 d_pos . . . . .	1387
10.347.2.6 d_tof . . . . .	1387
10.347.2.7 d_vel . . . . .	1387
10.347.2.8 i_prn . . . . .	1388
10.347.2.9 i_sv_ura . . . . .	1388
10.347.2.10 i_t0 . . . . .	1388
10.348 sbas_l1_telemetry_decoder_gs Class Reference . . . . .	1388
10.348.1 Detailed Description . . . . .	1389
10.348.2 Member Function Documentation . . . . .	1389
10.348.2.1 general_work() . . . . .	1389
10.348.2.2 reset() . . . . .	1390
10.348.2.3 set_channel() . . . . .	1390
10.348.2.4 set_satellite() . . . . .	1390
10.349 SbasL1TelemetryDecoder Class Reference . . . . .	1390
10.349.1 Detailed Description . . . . .	1393
10.349.2 Member Function Documentation . . . . .	1393
10.349.2.1 connect() . . . . .	1393
10.349.2.2 disconnect() . . . . .	1393
10.349.2.3 get_left_block() . . . . .	1393
10.349.2.4 get_right_block() . . . . .	1393
10.349.2.5 implementation() . . . . .	1393
10.349.2.6 item_size() . . . . .	1393
10.349.2.7 reset() . . . . .	1394
10.349.2.8 role() . . . . .	1394
10.349.2.9 set_channel() . . . . .	1394
10.349.2.10 set_satellite() . . . . .	1394
10.350 sbs_t Struct Reference . . . . .	1394
10.350.1 Detailed Description . . . . .	1395
10.350.2 Member Data Documentation . . . . .	1395
10.350.2.1 msgs . . . . .	1395
10.350.2.2 n . . . . .	1395
10.350.2.3 nmax . . . . .	1395
10.351 sbscorr_t Struct Reference . . . . .	1396
10.351.1 Detailed Description . . . . .	1396
10.351.2 Member Data Documentation . . . . .	1396
10.351.2.1 ai . . . . .	1396
10.351.2.2 dt . . . . .	1396
10.351.2.3 iodf . . . . .	1397
10.351.2.4 prc . . . . .	1397
10.351.2.5 rrc . . . . .	1397
10.351.2.6 t0 . . . . .	1397
10.351.2.7 udre . . . . .	1397

10.352 sbsigp_t Struct Reference . . . . .	1397
10.352.1 Detailed Description . . . . .	1398
10.352.2 Member Data Documentation . . . . .	1398
10.352.2.1 delay . . . . .	1398
10.352.2.2 give . . . . .	1398
10.352.2.3 lat . . . . .	1398
10.352.2.4 lon . . . . .	1398
10.352.2.5 to . . . . .	1398
10.353 sbsigband_t Struct Reference . . . . .	1399
10.353.1 Detailed Description . . . . .	1399
10.353.2 Member Data Documentation . . . . .	1399
10.353.2.1 bite . . . . .	1399
10.353.2.2 bits . . . . .	1399
10.353.2.3 x . . . . .	1399
10.353.2.4 y . . . . .	1399
10.354 sbSION_t Struct Reference . . . . .	1400
10.354.1 Detailed Description . . . . .	1400
10.354.2 Member Data Documentation . . . . .	1400
10.354.2.1 igp . . . . .	1400
10.354.2.2 iodi . . . . .	1400
10.354.2.3 nigp . . . . .	1401
10.355 sbSlcorr_t Struct Reference . . . . .	1401
10.355.1 Detailed Description . . . . .	1401
10.355.2 Member Data Documentation . . . . .	1401
10.355.2.1 daf0 . . . . .	1401
10.355.2.2 daf1 . . . . .	1402
10.355.2.3 dpos . . . . .	1402
10.355.2.4 dvel . . . . .	1402
10.355.2.5 iode . . . . .	1402
10.355.2.6 to . . . . .	1402
10.356 sbSmsg_t Struct Reference . . . . .	1402
10.356.1 Detailed Description . . . . .	1403
10.356.2 Member Data Documentation . . . . .	1403
10.356.2.1 msg . . . . .	1403
10.356.2.2 prn . . . . .	1403
10.356.2.3 tow . . . . .	1403
10.356.2.4 week . . . . .	1403
10.357 sbSSat_t Struct Reference . . . . .	1404
10.357.1 Detailed Description . . . . .	1405
10.357.2 Member Data Documentation . . . . .	1405
10.357.2.1 iodp . . . . .	1405
10.357.2.2 nsat . . . . .	1405

10.357.2.3 sat	1405
10.357.2.4 tlat	1405
10.358 sbssatp_t Struct Reference	1406
10.358.1 Detailed Description	1406
10.358.2 Member Data Documentation	1406
10.358.2.1 fcorr	1406
10.358.2.2 lcorr	1407
10.358.2.3 sat	1407
10.359 sdr_gnss_packet_t Struct Reference	1407
10.359.1 Detailed Description	1408
10.359.2 Member Data Documentation	1408
10.359.2.1 doppler	1408
10.359.2.2 galileo_satellites	1408
10.359.2.3 gps_satellites	1408
10.359.2.4 height	1408
10.359.2.5 latitude	1408
10.359.2.6 longitude	1408
10.359.2.7 microseconds	1408
10.359.2.8 nsvfix	1408
10.359.2.9 prn	1408
10.359.2.10 reserved	1409
10.359.2.11 snr	1409
10.359.2.12 status	1409
10.359.2.13 velocity	1409
10.360 seph_t Struct Reference	1409
10.360.1 Detailed Description	1410
10.360.2 Member Data Documentation	1410
10.360.2.1 acc	1410
10.360.2.2 af0	1410
10.360.2.3 af1	1410
10.360.2.4 pos	1410
10.360.2.5 sat	1410
10.360.2.6 sva	1410
10.360.2.7 svh	1410
10.360.2.8 t0	1410
10.360.2.9 tof	1411
10.360.2.10 vel	1411
10.361 Serdes_Galileo_Eph Class Reference	1411
10.361.1 Detailed Description	1411
10.361.2 Constructor & Destructor Documentation	1412
10.361.2.1 Serdes_Galileo_Eph() [1/3]	1412
10.361.2.2 ~Serdes_Galileo_Eph()	1412



10.361.2.3 Serdes_Galileo_Eph() [2/3]	1412
10.361.2.4 Serdes_Galileo_Eph() [3/3]	1412
10.361.3 Member Function Documentation	1412
10.361.3.1 createProtobuffer()	1412
10.361.3.2 operator=() [1/2]	1412
10.361.3.3 operator=() [2/2]	1412
10.361.3.4 readProtobuffer()	1412
10.362 Serdes_Gnss_Synchro Class Reference	1413
10.362.1 Detailed Description	1413
10.362.2 Constructor & Destructor Documentation	1414
10.362.2.1 Serdes_Gnss_Synchro() [1/3]	1414
10.362.2.2 ~Serdes_Gnss_Synchro()	1414
10.362.2.3 Serdes_Gnss_Synchro() [2/3]	1414
10.362.2.4 Serdes_Gnss_Synchro() [3/3]	1414
10.362.3 Member Function Documentation	1414
10.362.3.1 createProtobuffer()	1414
10.362.3.2 operator=() [1/2]	1414
10.362.3.3 operator=() [2/2]	1414
10.362.3.4 readProtobuffer()	1414
10.363 Serdes_Gps_Eph Class Reference	1415
10.363.1 Detailed Description	1415
10.363.2 Constructor & Destructor Documentation	1416
10.363.2.1 Serdes_Gps_Eph() [1/3]	1416
10.363.2.2 ~Serdes_Gps_Eph()	1416
10.363.2.3 Serdes_Gps_Eph() [2/3]	1416
10.363.2.4 Serdes_Gps_Eph() [3/3]	1416
10.363.3 Member Function Documentation	1416
10.363.3.1 createProtobuffer()	1416
10.363.3.2 operator=() [1/2]	1416
10.363.3.3 operator=() [2/2]	1416
10.363.3.4 readProtobuffer()	1416
10.364 Serdes_Monitor_Pvt Class Reference	1417
10.364.1 Detailed Description	1417
10.364.2 Constructor & Destructor Documentation	1418
10.364.2.1 Serdes_Monitor_Pvt() [1/3]	1418
10.364.2.2 ~Serdes_Monitor_Pvt()	1418
10.364.2.3 Serdes_Monitor_Pvt() [2/3]	1418
10.364.2.4 Serdes_Monitor_Pvt() [3/3]	1418
10.364.3 Member Function Documentation	1418
10.364.3.1 createProtobuffer()	1418
10.364.3.2 operator=() [1/2]	1418
10.364.3.3 operator=() [2/2]	1418

10.364.3.4 readProtobuffer()	1418
10.365 Serdes_Nav_Message Class Reference	1419
10.365.1 Detailed Description	1419
10.365.2 Constructor & Destructor Documentation	1419
10.365.2.1 Serdes_Nav_Message() [1/3]	1419
10.365.2.2 ~Serdes_Nav_Message()	1419
10.365.2.3 Serdes_Nav_Message() [2/3]	1420
10.365.2.4 Serdes_Nav_Message() [3/3]	1420
10.365.3 Member Function Documentation	1420
10.365.3.1 createProtobuffer()	1420
10.365.3.2 operator=() [1/2]	1420
10.365.3.3 operator=() [2/2]	1420
10.365.3.4 readProtobuffer()	1420
10.366 serial_t Struct Reference	1421
10.366.1 Detailed Description	1421
10.366.2 Member Data Documentation	1421
10.366.2.1 dev	1421
10.366.2.2 error	1421
10.367 short_x2_to_cshort Class Reference	1421
10.367.1 Detailed Description	1422
10.368 signal_generator_c Class Reference	1423
10.368.1 Detailed Description	1424
10.368.2 Friends And Related Symbol Documentation	1424
10.368.2.1 signal_make_generator_c	1424
10.369 SignalConditioner Class Reference	1424
10.369.1 Detailed Description	1427
10.369.2 Constructor & Destructor Documentation	1427
10.369.2.1 SignalConditioner()	1427
10.369.2.2 ~SignalConditioner()	1427
10.369.3 Member Function Documentation	1427
10.369.3.1 connect()	1427
10.369.3.2 data_type_adapter()	1427
10.369.3.3 disconnect()	1427
10.369.3.4 get_left_block()	1428
10.369.3.5 get_right_block()	1428
10.369.3.6 implementation()	1428
10.369.3.7 input_filter()	1428
10.369.3.8 item_size()	1428
10.369.3.9 resampler()	1428
10.369.3.10 role()	1428
10.370 SignalGenerator Class Reference	1428
10.370.1 Detailed Description	1431

10.370.2 Member Function Documentation . . . . .	1431
10.370.2.1 connect() . . . . .	1431
10.370.2.2 disconnect() . . . . .	1431
10.370.2.3 get_left_block() . . . . .	1431
10.370.2.4 get_right_block() . . . . .	1431
10.370.2.5 implementation() . . . . .	1431
10.370.2.6 item_size() . . . . .	1431
10.370.2.7 role() . . . . .	1431
10.371 SignalSourceBase Class Reference . . . . .	1432
10.371.1 Detailed Description . . . . .	1434
10.371.2 Constructor & Destructor Documentation . . . . .	1434
10.371.2.1 SignalSourceBase() . . . . .	1434
10.371.3 Member Function Documentation . . . . .	1434
10.371.3.1 decode_item_type() . . . . .	1434
10.371.3.2 get_left_block() . . . . .	1434
10.371.3.3 getRfChannels() . . . . .	1435
10.371.3.4 implementation() . . . . .	1435
10.371.3.5 role() . . . . .	1435
10.372 SignalSourceInterface Class Reference . . . . .	1435
10.372.1 Detailed Description . . . . .	1437
10.373 snrmask_t Struct Reference . . . . .	1437
10.373.1 Detailed Description . . . . .	1437
10.373.2 Member Data Documentation . . . . .	1437
10.373.2.1 ena . . . . .	1437
10.373.2.2 mask . . . . .	1437
10.374 sol_t Struct Reference . . . . .	1438
10.374.1 Detailed Description . . . . .	1438
10.374.2 Member Data Documentation . . . . .	1438
10.374.2.1 age . . . . .	1438
10.374.2.2 dtr . . . . .	1439
10.374.2.3 ns . . . . .	1439
10.374.2.4 qr . . . . .	1439
10.374.2.5 ratio . . . . .	1439
10.374.2.6 rr . . . . .	1439
10.374.2.7 stat . . . . .	1439
10.374.2.8 thres . . . . .	1439
10.374.2.9 time . . . . .	1439
10.374.2.10 type . . . . .	1439
10.375 solbuf_t Struct Reference . . . . .	1440
10.375.1 Detailed Description . . . . .	1441
10.375.2 Member Data Documentation . . . . .	1441
10.375.2.1 buff . . . . .	1441

10.375.2.2 cyclic . . . . .	1441
10.375.2.3 data . . . . .	1441
10.375.2.4 end . . . . .	1441
10.375.2.5 n . . . . .	1441
10.375.2.6 nb . . . . .	1441
10.375.2.7 nmax . . . . .	1441
10.375.2.8 rb . . . . .	1441
10.375.2.9 start . . . . .	1441
10.375.2.10 time . . . . .	1442
10.376 solopt_t Struct Reference . . . . .	1442
10.376.1 Detailed Description . . . . .	1443
10.376.2 Member Data Documentation . . . . .	1443
10.376.2.1 datum . . . . .	1443
10.376.2.2 degf . . . . .	1443
10.376.2.3 geoid . . . . .	1443
10.376.2.4 height . . . . .	1443
10.376.2.5 maxsolstd . . . . .	1443
10.376.2.6 nmeaintv . . . . .	1443
10.376.2.7 outhead . . . . .	1443
10.376.2.8 outopt . . . . .	1443
10.376.2.9 posf . . . . .	1443
10.376.2.10 prog . . . . .	1443
10.376.2.11 sep . . . . .	1443
10.376.2.12 solstatic . . . . .	1444
10.376.2.13 sstat . . . . .	1444
10.376.2.14 timef . . . . .	1444
10.376.2.15 times . . . . .	1444
10.376.2.16 timeu . . . . .	1444
10.376.2.17 trace . . . . .	1444
10.377 solstat_t Struct Reference . . . . .	1445
10.377.1 Detailed Description . . . . .	1445
10.377.2 Member Data Documentation . . . . .	1446
10.377.2.1 az . . . . .	1446
10.377.2.2 el . . . . .	1446
10.377.2.3 flag . . . . .	1446
10.377.2.4 frq . . . . .	1446
10.377.2.5 lock . . . . .	1446
10.377.2.6 outc . . . . .	1446
10.377.2.7 rejc . . . . .	1446
10.377.2.8 resc . . . . .	1446
10.377.2.9 resp . . . . .	1446
10.377.2.10 sat . . . . .	1446

10.377.2.11 slipc . . . . .	1446
10.377.2.12 snr . . . . .	1446
10.377.2.13 time . . . . .	1447
10.378 solstatbuf_t Struct Reference . . . . .	1447
10.378.1 Detailed Description . . . . .	1448
10.378.2 Member Data Documentation . . . . .	1448
10.378.2.1 data . . . . .	1448
10.378.2.2 n . . . . .	1448
10.378.2.3 nmax . . . . .	1448
10.379 SpirFileSignalSource Class Reference . . . . .	1448
10.379.1 Detailed Description . . . . .	1452
10.379.2 Member Function Documentation . . . . .	1452
10.379.2.1 create_file_source_hook() . . . . .	1452
10.379.2.2 itemTypeToSize() . . . . .	1452
10.379.2.3 post_disconnect_hook() . . . . .	1452
10.379.2.4 pre_connect_hook() . . . . .	1452
10.379.2.5 source() . . . . .	1453
10.380 SpirGSS6450FileSignalSource Class Reference . . . . .	1453
10.380.1 Detailed Description . . . . .	1456
10.380.2 Member Function Documentation . . . . .	1456
10.380.2.1 connect() . . . . .	1456
10.380.2.2 disconnect() . . . . .	1456
10.380.2.3 filename() . . . . .	1456
10.380.2.4 get_left_block() . . . . .	1456
10.380.2.5 get_right_block() [1/2] . . . . .	1457
10.380.2.6 get_right_block() [2/2] . . . . .	1457
10.380.2.7 item_size() . . . . .	1457
10.380.2.8 item_type() . . . . .	1457
10.380.2.9 repeat() . . . . .	1457
10.380.2.10 samples() . . . . .	1457
10.380.2.11 sampling_frequency() . . . . .	1457
10.381 ssat_t Struct Reference . . . . .	1458
10.381.1 Detailed Description . . . . .	1459
10.381.2 Member Data Documentation . . . . .	1459
10.381.2.1 azel . . . . .	1459
10.381.2.2 fix . . . . .	1459
10.381.2.3 gf . . . . .	1459
10.381.2.4 gf2 . . . . .	1459
10.381.2.5 half . . . . .	1459
10.381.2.6 lock . . . . .	1459
10.381.2.7 mw . . . . .	1459
10.381.2.8 outc . . . . .	1459

10.381.2.9 ph . . . . .	1459
10.381.2.10 phw . . . . .	1459
10.381.2.11 pt . . . . .	1460
10.381.2.12 rejc . . . . .	1460
10.381.2.13 resc . . . . .	1460
10.381.2.14 resp . . . . .	1460
10.381.2.15 slip . . . . .	1460
10.381.2.16 slipc . . . . .	1460
10.381.2.17 snr . . . . .	1460
10.381.2.18 sys . . . . .	1460
10.381.2.19 vs . . . . .	1460
10.381.2.20 vsat . . . . .	1460
10.382 ssr_t Struct Reference . . . . .	1461
10.382.1 Detailed Description . . . . .	1462
10.382.2 Member Data Documentation . . . . .	1462
10.382.2.1 cbias . . . . .	1462
10.382.2.2 dclk . . . . .	1462
10.382.2.3 ddeph . . . . .	1462
10.382.2.4 deph . . . . .	1462
10.382.2.5 hrcclk . . . . .	1462
10.382.2.6 iod . . . . .	1462
10.382.2.7 iodcrc . . . . .	1462
10.382.2.8 iode . . . . .	1462
10.382.2.9 pbias . . . . .	1462
10.382.2.10 refd . . . . .	1462
10.382.2.11 stdpb . . . . .	1462
10.382.2.12 t0 . . . . .	1463
10.382.2.13 udi . . . . .	1463
10.382.2.14 update . . . . .	1463
10.382.2.15 ura . . . . .	1463
10.382.2.16 yaw_ang . . . . .	1463
10.382.2.17 yaw_rate . . . . .	1463
10.383 sta_t Struct Reference . . . . .	1464
10.383.1 Detailed Description . . . . .	1464
10.383.2 Member Data Documentation . . . . .	1464
10.383.2.1 antdes . . . . .	1464
10.383.2.2 antsetup . . . . .	1465
10.383.2.3 antsno . . . . .	1465
10.383.2.4 del . . . . .	1465
10.383.2.5 deltype . . . . .	1465
10.383.2.6 hgt . . . . .	1465
10.383.2.7 itrfr . . . . .	1465

10.383.2.8 marker	1465
10.383.2.9 name	1465
10.383.2.10 pos	1465
10.383.2.11 recsno	1465
10.383.2.12 rectype	1465
10.383.2.13 recver	1465
10.384 stec_t Struct Reference	1466
10.384.1 Detailed Description	1466
10.384.2 Member Data Documentation	1466
10.384.2.1 azel	1466
10.384.2.2 flag	1466
10.384.2.3 ion	1466
10.384.2.4 sat	1467
10.384.2.5 std	1467
10.384.2.6 time	1467
10.385 stream_cfg Struct Reference	1467
10.385.1 Detailed Description	1467
10.385.2 Member Data Documentation	1467
10.385.2.1 bw_hz	1467
10.385.2.2 fs_hz	1467
10.385.2.3 lo_hz	1468
10.385.2.4 rfport	1468
10.386 stream_t Struct Reference	1468
10.386.1 Detailed Description	1469
10.386.2 Member Data Documentation	1469
10.386.2.1 inb	1469
10.386.2.2 inbt	1469
10.386.2.3 inr	1469
10.386.2.4 lock	1469
10.386.2.5 mode	1469
10.386.2.6 msg	1469
10.386.2.7 outb	1469
10.386.2.8 outbt	1469
10.386.2.9 outr	1469
10.386.2.10 path	1470
10.386.2.11 port	1470
10.386.2.12 state	1470
10.386.2.13 tact	1470
10.386.2.14 tick	1470
10.386.2.15 type	1470
10.387 StringConverter Class Reference	1470
10.387.1 Detailed Description	1471

10.388 Tag Class Reference	1472
10.388.1 Detailed Description	1473
10.388.2 Member Enumeration Documentation	1473
10.388.2.1 e_verification_status	1473
10.388.3 Constructor & Destructor Documentation	1473
10.388.3.1 Tag() [1/2]	1473
10.388.3.2 Tag() [2/2]	1473
10.388.4 Member Data Documentation	1473
10.388.4.1 ADKD	1473
10.388.4.2 computed_tag	1473
10.388.4.3 cop	1474
10.388.4.4 CTR	1474
10.388.4.5 id_counter	1474
10.388.4.6 nav_data	1474
10.388.4.7 PRN_d	1474
10.388.4.8 PRNa	1474
10.388.4.9 received_tag	1474
10.388.4.10 skipped	1474
10.388.4.11 status	1474
10.388.4.12 tag_id	1474
10.388.4.13 TOW	1474
10.388.4.14 WN	1474
10.389 Tcp_Communication Class Reference	1475
10.389.1 Detailed Description	1475
10.390 Tcp_Packet_Data Class Reference	1475
10.390.1 Detailed Description	1476
10.390.2 Member Data Documentation	1476
10.390.2.1 proc_pack_carr_error	1476
10.390.2.2 proc_pack_carrier_doppler_hz	1476
10.390.2.3 proc_pack_code_error	1476
10.391 tcp_t Struct Reference	1477
10.391.1 Detailed Description	1477
10.391.2 Member Data Documentation	1477
10.391.2.1 addr	1477
10.391.2.2 port	1477
10.391.2.3 saddr	1477
10.391.2.4 sock	1478
10.391.2.5 state	1478
10.391.2.6 tact	1478
10.391.2.7 tcon	1478
10.391.2.8 tdis	1478
10.392 tcpcli_t Struct Reference	1479



10.392.1 Detailed Description . . . . .	1479
10.392.2 Member Data Documentation . . . . .	1479
10.392.2.1 svr . . . . .	1479
10.392.2.2 tirecon . . . . .	1479
10.392.2.3 toinact . . . . .	1480
10.393 TcpCmdInterface Class Reference . . . . .	1480
10.393.1 Detailed Description . . . . .	1480
10.393.2 Member Function Documentation . . . . .	1480
10.393.2.1 get_LLH() . . . . .	1480
10.393.2.2 get_utc_time() . . . . .	1480
10.394 tcpsvr_t Struct Reference . . . . .	1481
10.394.1 Detailed Description . . . . .	1481
10.394.2 Member Data Documentation . . . . .	1481
10.394.2.1 cli . . . . .	1481
10.394.2.2 svr . . . . .	1481
10.395 tec_t Struct Reference . . . . .	1482
10.395.1 Detailed Description . . . . .	1482
10.395.2 Member Data Documentation . . . . .	1482
10.395.2.1 data . . . . .	1482
10.395.2.2 hgts . . . . .	1482
10.395.2.3 lats . . . . .	1483
10.395.2.4 lons . . . . .	1483
10.395.2.5 ndata . . . . .	1483
10.395.2.6 rb . . . . .	1483
10.395.2.7 rms . . . . .	1483
10.395.2.8 time . . . . .	1483
10.396 TelemetryDecoderInterface Class Reference . . . . .	1483
10.396.1 Detailed Description . . . . .	1485
10.397 tle_t Struct Reference . . . . .	1485
10.397.1 Detailed Description . . . . .	1486
10.397.2 Member Data Documentation . . . . .	1486
10.397.2.1 data . . . . .	1486
10.397.2.2 n . . . . .	1486
10.397.2.3 nmax . . . . .	1486
10.398 tied_t Struct Reference . . . . .	1487
10.398.1 Detailed Description . . . . .	1488
10.398.2 Member Data Documentation . . . . .	1488
10.398.2.1 alias . . . . .	1488
10.398.2.2 bstar . . . . .	1488
10.398.2.3 desig . . . . .	1488
10.398.2.4 ecc . . . . .	1488
10.398.2.5 eleno . . . . .	1488

10.398.2.6 epoch	1488
10.398.2.7 etype	1488
10.398.2.8 inc	1488
10.398.2.9 M	1488
10.398.2.10 n	1488
10.398.2.11 name	1488
10.398.2.12 nddot	1489
10.398.2.13 ndot	1489
10.398.2.14 OMG	1489
10.398.2.15 omg	1489
10.398.2.16 rev	1489
10.398.2.17 satclass	1489
10.398.2.18 satno	1489
10.399 Tlm_Conf Class Reference	1490
10.399.1 Detailed Description	1491
10.399.2 Member Data Documentation	1491
10.399.2.1 dump	1491
10.399.2.2 dump_crc_stats	1491
10.399.2.3 dump_crc_stats_filename	1491
10.399.2.4 dump_filename	1491
10.399.2.5 dump_mat	1491
10.399.2.6 enable_navdata_monitor	1491
10.399.2.7 enable_reed_solomon	1491
10.399.2.8 remove_dat	1491
10.399.2.9 there_are_e1_channels	1491
10.399.2.10 there_are_e6_channels	1491
10.399.2.11 use_ced	1492
10.400 Tlm_CRC_Stats Class Reference	1492
10.400.1 Detailed Description	1492
10.400.2 Member Function Documentation	1492
10.400.2.1 initialize()	1492
10.400.2.2 set_channel()	1492
10.400.2.3 update_CRC_stats()	1493
10.401 Tracking_2nd_DLL_filter Class Reference	1493
10.401.1 Detailed Description	1493
10.401.2 Member Function Documentation	1494
10.401.2.1 get_code_nco()	1494
10.401.2.2 initialize()	1494
10.401.2.3 set_DLL_BW()	1494
10.401.2.4 set_pdi()	1494
10.402 Tracking_2nd_PLL_filter Class Reference	1494
10.402.1 Detailed Description	1495

10.402.2 Member Function Documentation . . . . .	1495
10.402.2.1 set_pdi() . . . . .	1495
10.402.2.2 set_PLL_BW() . . . . .	1495
10.403 Tracking_FLL_PLL_filter Class Reference . . . . .	1495
10.403.1 Detailed Description . . . . .	1496
10.404 Tracking_loop_filter Class Reference . . . . .	1496
10.404.1 Detailed Description . . . . .	1497
10.404.2 Constructor & Destructor Documentation . . . . .	1497
10.404.2.1 Tracking_loop_filter() . . . . .	1497
10.404.3 Member Function Documentation . . . . .	1497
10.404.3.1 operator=() . . . . .	1497
10.405 TrackingCmd Class Reference . . . . .	1497
10.405.1 Detailed Description . . . . .	1498
10.405.2 Member Data Documentation . . . . .	1498
10.405.2.1 carrier_freq_hz . . . . .	1498
10.405.2.2 carrier_freq_rate_hz_s . . . . .	1498
10.405.2.3 code_freq_chips . . . . .	1498
10.405.2.4 enable_carrier_nco_cmd . . . . .	1498
10.405.2.5 enable_code_nco_cmd . . . . .	1498
10.405.2.6 sample_counter . . . . .	1498
10.406 TrackingInterface Class Reference . . . . .	1498
10.406.1 Detailed Description . . . . .	1500
10.406.2 Member Function Documentation . . . . .	1500
10.406.2.1 set_channel() . . . . .	1500
10.406.2.2 set_gnss_synchro() . . . . .	1500
10.406.2.3 start_tracking() . . . . .	1500
10.406.2.4 stop_tracking() . . . . .	1500
10.407 trop_t Struct Reference . . . . .	1501
10.407.1 Detailed Description . . . . .	1501
10.407.2 Member Data Documentation . . . . .	1501
10.407.2.1 std . . . . .	1501
10.407.2.2 time . . . . .	1501
10.407.2.3 trp . . . . .	1501
10.408 TwoBitCpxFileSignalSource Class Reference . . . . .	1502
10.408.1 Detailed Description . . . . .	1505
10.408.2 Member Function Documentation . . . . .	1505
10.408.2.1 create_file_source_hook() . . . . .	1505
10.408.2.2 itemTypeToSize() . . . . .	1505
10.408.2.3 packetsPerSample() . . . . .	1505
10.408.2.4 pre_connect_hook() . . . . .	1505
10.408.2.5 pre_disconnect_hook() . . . . .	1506
10.408.2.6 source() . . . . .	1506

10.409 TwoBitPackedFileSignalSource Class Reference . . . . .	1506
10.409.1 Detailed Description . . . . .	1510
10.409.2 Member Function Documentation . . . . .	1510
10.409.2.1 create_file_source_hook() . . . . .	1510
10.409.2.2 itemTypeToSize() . . . . .	1510
10.409.2.3 packetsPerSample() . . . . .	1510
10.409.2.4 pre_connect_hook() . . . . .	1510
10.409.2.5 pre_disconnect_hook() . . . . .	1511
10.409.2.6 source() . . . . .	1511
10.410 UhdSignalSource Class Reference . . . . .	1511
10.410.1 Detailed Description . . . . .	1514
10.410.2 Member Function Documentation . . . . .	1514
10.410.2.1 connect() . . . . .	1514
10.410.2.2 disconnect() . . . . .	1514
10.410.2.3 get_left_block() . . . . .	1514
10.410.2.4 get_right_block() [1/2] . . . . .	1514
10.410.2.5 get_right_block() [2/2] . . . . .	1514
10.410.2.6 item_size() . . . . .	1515
10.411 unpack_2bit_samples Class Reference . . . . .	1515
10.411.1 Detailed Description . . . . .	1516
10.412 unpack_byte_2bit_cpx_samples Class Reference . . . . .	1516
10.412.1 Detailed Description . . . . .	1518
10.413 unpack_byte_2bit_samples Class Reference . . . . .	1518
10.413.1 Detailed Description . . . . .	1519
10.414 unpack_byte_4bit_samples Class Reference . . . . .	1519
10.414.1 Detailed Description . . . . .	1521
10.415 unpack_intspir_1bit_samples Class Reference . . . . .	1521
10.415.1 Detailed Description . . . . .	1522
10.416 unpack_short_byte_samples Class Reference . . . . .	1522
10.416.1 Detailed Description . . . . .	1524
10.417 unpack_spir_gss6450_samples Class Reference . . . . .	1524
10.417.1 Detailed Description . . . . .	1525
10.418 UnscentedFilter Class Reference . . . . .	1526
10.418.1 Detailed Description . . . . .	1526
10.419 url_t Struct Reference . . . . .	1527
10.419.1 Detailed Description . . . . .	1527
10.419.2 Member Data Documentation . . . . .	1527
10.419.2.1 dir . . . . .	1527
10.419.2.2 path . . . . .	1527
10.419.2.3 tint . . . . .	1527
10.419.2.4 type . . . . .	1527
10.420 v27_decision_t Struct Reference . . . . .	1528

10.420.1 Detailed Description . . . . .	1528
10.420.2 Member Data Documentation . . . . .	1528
10.420.2.1 w . . . . .	1528
10.421 v27_poly_t Struct Reference . . . . .	1528
10.421.1 Detailed Description . . . . .	1529
10.421.2 Member Data Documentation . . . . .	1529
10.421.2.1 c0 . . . . .	1529
10.421.2.2 c1 . . . . .	1529
10.422 v27_t Struct Reference . . . . .	1529
10.422.1 Detailed Description . . . . .	1530
10.422.2 Member Data Documentation . . . . .	1530
10.422.2.1 decisions . . . . .	1530
10.422.2.2 decisions_count . . . . .	1530
10.422.2.3 decisions_index . . . . .	1530
10.422.2.4 metrics1 . . . . .	1530
10.422.2.5 metrics2 . . . . .	1530
10.422.2.6 new_metrics . . . . .	1530
10.422.2.7 old_metrics . . . . .	1530
10.422.2.8 poly . . . . .	1530
10.423 Viterbi_Decoder Class Reference . . . . .	1530
10.423.1 Detailed Description . . . . .	1531
10.423.2 Constructor & Destructor Documentation . . . . .	1531
10.423.2.1 Viterbi_Decoder() . . . . .	1531
10.423.3 Member Function Documentation . . . . .	1531
10.423.3.1 decode() . . . . .	1531
10.423.3.2 reset() . . . . .	1532
10.424 Viterbi_Decoder_Sbas Class Reference . . . . .	1532
10.424.1 Detailed Description . . . . .	1532
10.424.2 Member Function Documentation . . . . .	1532
10.424.2.1 decode_block() . . . . .	1532
10.425 ZmqSignalSource Class Reference . . . . .	1533
10.425.1 Detailed Description . . . . .	1536
10.425.2 Member Function Documentation . . . . .	1536
10.425.2.1 connect() . . . . .	1536
10.425.2.2 disconnect() . . . . .	1536
10.425.2.3 get_right_block() . . . . .	1536
10.425.2.4 item_size() . . . . .	1537
<b>11 File Documentation</b>	<b>1539</b>
11.1 beidou_b1i_pcps_acquisition.h File Reference . . . . .	1539
11.1.1 Detailed Description . . . . .	1539
11.2 beidou_b1i_pcps_acquisition.h . . . . .	1539

11.3 beidou_b3i_pcps_acquisition.h File Reference	1542
11.3.1 Detailed Description	1542
11.4 beidou_b3i_pcps_acquisition.h	1542
11.5 galileo_e1_pcps_8ms_ambiguous_acquisition.h File Reference	1544
11.5.1 Detailed Description	1545
11.6 galileo_e1_pcps_8ms_ambiguous_acquisition.h	1545
11.7 galileo_e1_pcps_ambiguous_acquisition.h File Reference	1547
11.7.1 Detailed Description	1548
11.8 galileo_e1_pcps_ambiguous_acquisition.h	1548
11.9 galileo_e1_pcps_ambiguous_acquisition_fpga.h File Reference	1550
11.9.1 Detailed Description	1551
11.10 galileo_e1_pcps_ambiguous_acquisition_fpga.h	1551
11.11 galileo_e1_pcps_cccwsr_ambiguous_acquisition.h File Reference	1554
11.11.1 Detailed Description	1554
11.12 galileo_e1_pcps_cccwsr_ambiguous_acquisition.h	1554
11.13 galileo_e1_pcps_quicksync_ambiguous_acquisition.h File Reference	1556
11.13.1 Detailed Description	1557
11.14 galileo_e1_pcps_quicksync_ambiguous_acquisition.h	1557
11.15 galileo_e1_pcps_tong_ambiguous_acquisition.h File Reference	1559
11.15.1 Detailed Description	1560
11.16 galileo_e1_pcps_tong_ambiguous_acquisition.h	1560
11.17 galileo_e5a_noncoherent_iq_acquisition_caf.h File Reference	1562
11.17.1 Detailed Description	1563
11.18 galileo_e5a_noncoherent_iq_acquisition_caf.h	1563
11.19 galileo_e5a_pcps_acquisition.h File Reference	1565
11.19.1 Detailed Description	1565
11.20 galileo_e5a_pcps_acquisition.h	1566
11.21 galileo_e5a_pcps_acquisition_fpga.h File Reference	1568
11.21.1 Detailed Description	1568
11.22 galileo_e5a_pcps_acquisition_fpga.h	1568
11.23 galileo_e5b_pcps_acquisition.h File Reference	1571
11.23.1 Detailed Description	1571
11.24 galileo_e5b_pcps_acquisition.h	1572
11.25 galileo_e5b_pcps_acquisition_fpga.h File Reference	1574
11.25.1 Detailed Description	1575
11.26 galileo_e5b_pcps_acquisition_fpga.h	1575
11.27 galileo_e6_pcps_acquisition.h File Reference	1578
11.27.1 Detailed Description	1578
11.28 galileo_e6_pcps_acquisition.h	1578
11.29 glonass_l1_ca_pcps_acquisition.h File Reference	1581
11.29.1 Detailed Description	1581
11.30 glonass_l1_ca_pcps_acquisition.h	1581

11.31	<a href="#">glonass_l2_ca_pcps_acquisition.h File Reference</a>	1583
11.31.1	Detailed Description	1584
11.32	<a href="#">glonass_l2_ca_pcps_acquisition.h</a>	1584
11.33	<a href="#">gps_l1_ca_pcps_acquisition.h File Reference</a>	1586
11.33.1	Detailed Description	1587
11.34	<a href="#">gps_l1_ca_pcps_acquisition.h</a>	1587
11.35	<a href="#">gps_l1_ca_pcps_acquisition_fine_doppler.h File Reference</a>	1589
11.35.1	Detailed Description	1590
11.36	<a href="#">gps_l1_ca_pcps_acquisition_fine_doppler.h</a>	1590
11.37	<a href="#">gps_l1_ca_pcps_acquisition_fpga.h File Reference</a>	1592
11.37.1	Detailed Description	1593
11.38	<a href="#">gps_l1_ca_pcps_acquisition_fpga.h</a>	1593
11.39	<a href="#">gps_l1_ca_pcps_assisted_acquisition.h File Reference</a>	1595
11.39.1	Detailed Description	1596
11.40	<a href="#">gps_l1_ca_pcps_assisted_acquisition.h</a>	1596
11.41	<a href="#">gps_l1_ca_pcps_openc1_acquisition.h File Reference</a>	1598
11.41.1	Detailed Description	1599
11.42	<a href="#">gps_l1_ca_pcps_openc1_acquisition.h</a>	1599
11.43	<a href="#">gps_l1_ca_pcps_quicksync_acquisition.h File Reference</a>	1601
11.43.1	Detailed Description	1601
11.44	<a href="#">gps_l1_ca_pcps_quicksync_acquisition.h</a>	1602
11.45	<a href="#">gps_l1_ca_pcps_tong_acquisition.h File Reference</a>	1604
11.45.1	Detailed Description	1604
11.46	<a href="#">gps_l1_ca_pcps_tong_acquisition.h</a>	1605
11.47	<a href="#">gps_l2_m_pcps_acquisition.h File Reference</a>	1607
11.47.1	Detailed Description	1607
11.48	<a href="#">gps_l2_m_pcps_acquisition.h</a>	1607
11.49	<a href="#">gps_l2_m_pcps_acquisition_fpga.h File Reference</a>	1610
11.49.1	Detailed Description	1610
11.50	<a href="#">gps_l2_m_pcps_acquisition_fpga.h</a>	1610
11.51	<a href="#">gps_l5i_pcps_acquisition.h File Reference</a>	1613
11.51.1	Detailed Description	1613
11.52	<a href="#">gps_l5i_pcps_acquisition.h</a>	1613
11.53	<a href="#">gps_l5i_pcps_acquisition_fpga.h File Reference</a>	1615
11.53.1	Detailed Description	1616
11.54	<a href="#">gps_l5i_pcps_acquisition_fpga.h</a>	1616
11.55	<a href="#">galileo_e5a_noncoherent_iq_acquisition_caf_cc.h File Reference</a>	1619
11.55.1	Detailed Description	1620
11.56	<a href="#">galileo_e5a_noncoherent_iq_acquisition_caf_cc.h</a>	1620
11.57	<a href="#">galileo_pcps_8ms_acquisition_cc.h File Reference</a>	1623
11.57.1	Detailed Description	1624
11.58	<a href="#">galileo_pcps_8ms_acquisition_cc.h</a>	1625

11.59 pcps_acquisition.h File Reference	1627
11.59.1 Detailed Description	1628
11.60 pcps_acquisition.h	1629
11.61 pcps_acquisition_fine_doppler_cc.h File Reference	1632
11.61.1 Detailed Description	1633
11.62 pcps_acquisition_fine_doppler_cc.h	1634
11.63 pcps_acquisition_fpga.h File Reference	1637
11.63.1 Detailed Description	1637
11.64 pcps_acquisition_fpga.h	1638
11.65 pcps_assisted_acquisition_cc.h File Reference	1640
11.65.1 Detailed Description	1641
11.66 pcps_assisted_acquisition_cc.h	1642
11.67 pcps_cccwsr_acquisition_cc.h File Reference	1645
11.67.1 Detailed Description	1646
11.68 pcps_cccwsr_acquisition_cc.h	1646
11.69 pcps_openc1_acquisition_cc.h File Reference	1649
11.69.1 Detailed Description	1650
11.69.2 Macro Definition Documentation	1650
11.69.2.1 CL_SILENCE_DEPRECATED	1650
11.70 pcps_openc1_acquisition_cc.h	1650
11.71 pcps_quicksync_acquisition_cc.h File Reference	1654
11.71.1 Detailed Description	1655
11.72 pcps_quicksync_acquisition_cc.h	1655
11.73 pcps_tong_acquisition_cc.h File Reference	1658
11.73.1 Detailed Description	1659
11.74 pcps_tong_acquisition_cc.h	1660
11.75 acq_conf.h File Reference	1663
11.75.1 Detailed Description	1664
11.76 acq_conf.h	1664
11.77 acq_conf_fpga.h File Reference	1665
11.77.1 Detailed Description	1666
11.78 acq_conf_fpga.h	1666
11.79 fpga_acquisition.h File Reference	1667
11.79.1 Detailed Description	1667
11.80 fpga_acquisition.h	1668
11.81 channel.h File Reference	1671
11.81.1 Detailed Description	1671
11.82 channel.h	1672
11.83 channel_fsm.h File Reference	1673
11.83.1 Detailed Description	1674
11.84 channel_fsm.h	1674
11.85 channel_msg_receiver_cc.h File Reference	1675



11.85.1 Detailed Description . . . . .	1676
11.86 channel_msg_receiver_cc.h . . . . .	1676
11.87 array_signal_conditioner.h File Reference . . . . .	1677
11.87.1 Detailed Description . . . . .	1677
11.88 array_signal_conditioner.h . . . . .	1677
11.89 signal_conditioner.h File Reference . . . . .	1678
11.89.1 Detailed Description . . . . .	1679
11.90 signal_conditioner.h . . . . .	1679
11.91 byte_to_short.h File Reference . . . . .	1680
11.91.1 Detailed Description . . . . .	1681
11.92 byte_to_short.h . . . . .	1681
11.93 cshort_to_grcomplex.h File Reference . . . . .	1682
11.93.1 Detailed Description . . . . .	1682
11.94 cshort_to_grcomplex.h . . . . .	1682
11.95 ibyte_to_cbyte.h File Reference . . . . .	1683
11.95.1 Detailed Description . . . . .	1684
11.96 ibyte_to_cbyte.h . . . . .	1684
11.97 ibyte_to_complex.h File Reference . . . . .	1685
11.97.1 Detailed Description . . . . .	1686
11.98 ibyte_to_complex.h . . . . .	1686
11.99 ibyte_to_cshort.h File Reference . . . . .	1687
11.99.1 Detailed Description . . . . .	1688
11.100 ibyte_to_cshort.h . . . . .	1688
11.101 ishort_to_complex.h File Reference . . . . .	1689
11.101.1 Detailed Description . . . . .	1689
11.102 ishort_to_complex.h . . . . .	1689
11.103 ishort_to_cshort.h File Reference . . . . .	1691
11.103.1 Detailed Description . . . . .	1691
11.104 ishort_to_cshort.h . . . . .	1691
11.105 cshort_to_gr_complex.h File Reference . . . . .	1692
11.105.1 Detailed Description . . . . .	1693
11.106 cshort_to_gr_complex.h . . . . .	1693
11.107 interleaved_byte_to_complex_byte.h File Reference . . . . .	1694
11.107.1 Detailed Description . . . . .	1695
11.108 interleaved_byte_to_complex_byte.h . . . . .	1695
11.109 interleaved_byte_to_complex_short.h File Reference . . . . .	1696
11.109.1 Detailed Description . . . . .	1697
11.110 interleaved_byte_to_complex_short.h . . . . .	1697
11.111 interleaved_short_to_complex_short.h File Reference . . . . .	1698
11.111.1 Detailed Description . . . . .	1699
11.112 interleaved_short_to_complex_short.h . . . . .	1699
11.113 beamformer_filter.h File Reference . . . . .	1700

11.113.1 Detailed Description . . . . .	1700
11.114 beamformer_filter.h . . . . .	1701
11.115 fir_filter.h File Reference . . . . .	1702
11.115.1 Detailed Description . . . . .	1702
11.116 fir_filter.h . . . . .	1702
11.117 freq_xlating_fir_filter.h File Reference . . . . .	1704
11.117.1 Detailed Description . . . . .	1704
11.118 freq_xlating_fir_filter.h . . . . .	1704
11.119 notch_filter.h File Reference . . . . .	1706
11.119.1 Detailed Description . . . . .	1706
11.120 notch_filter.h . . . . .	1707
11.121 notch_filter_lite.h File Reference . . . . .	1708
11.121.1 Detailed Description . . . . .	1708
11.122 notch_filter_lite.h . . . . .	1708
11.123 pulse_blanking_filter.h File Reference . . . . .	1709
11.123.1 Detailed Description . . . . .	1710
11.124 pulse_blanking_filter.h . . . . .	1710
11.125 beamformer.h File Reference . . . . .	1711
11.125.1 Detailed Description . . . . .	1711
11.126 beamformer.h . . . . .	1712
11.127 notch_cc.h File Reference . . . . .	1712
11.127.1 Detailed Description . . . . .	1713
11.128 notch_cc.h . . . . .	1713
11.129 notch_lite_cc.h File Reference . . . . .	1715
11.129.1 Detailed Description . . . . .	1715
11.130 notch_lite_cc.h . . . . .	1716
11.131 pulse_blanking_cc.h File Reference . . . . .	1717
11.131.1 Detailed Description . . . . .	1717
11.132 pulse_blanking_cc.h . . . . .	1718
11.133 beidou_b1i_signal_replica.h File Reference . . . . .	1719
11.133.1 Detailed Description . . . . .	1719
11.134 beidou_b1i_signal_replica.h . . . . .	1719
11.135 beidou_b3i_signal_replica.h File Reference . . . . .	1720
11.135.1 Detailed Description . . . . .	1721
11.136 beidou_b3i_signal_replica.h . . . . .	1721
11.137 byte_x2_to_complex_byte.h File Reference . . . . .	1722
11.137.1 Detailed Description . . . . .	1723
11.138 byte_x2_to_complex_byte.h . . . . .	1723
11.139 complex_byte_to_float_x2.h File Reference . . . . .	1724
11.139.1 Detailed Description . . . . .	1724
11.140 complex_byte_to_float_x2.h . . . . .	1724
11.141 complex_float_to_complex_byte.h File Reference . . . . .	1725

11.141.1 Detailed Description . . . . .	1726
11.142 complex_float_to_complex_byte.h . . . . .	1726
11.143 conjugate_cc.h File Reference . . . . .	1727
11.143.1 Detailed Description . . . . .	1728
11.144 conjugate_cc.h . . . . .	1728
11.145 conjugate_ic.h File Reference . . . . .	1729
11.145.1 Detailed Description . . . . .	1730
11.146 conjugate_ic.h . . . . .	1730
11.147 conjugate_sc.h File Reference . . . . .	1730
11.147.1 Detailed Description . . . . .	1731
11.148 conjugate_sc.h . . . . .	1731
11.149 cshort_to_float_x2.h File Reference . . . . .	1732
11.149.1 Detailed Description . . . . .	1733
11.150 cshort_to_float_x2.h . . . . .	1733
11.151 galileo_e1_signal_replica.h File Reference . . . . .	1734
11.151.1 Detailed Description . . . . .	1735
11.152 galileo_e1_signal_replica.h . . . . .	1735
11.153 galileo_e5_signal_replica.h File Reference . . . . .	1736
11.153.1 Detailed Description . . . . .	1737
11.154 galileo_e5_signal_replica.h . . . . .	1737
11.155 galileo_e6_signal_replica.h File Reference . . . . .	1738
11.155.1 Detailed Description . . . . .	1739
11.156 galileo_e6_signal_replica.h . . . . .	1739
11.157 geofunctions.h File Reference . . . . .	1740
11.157.1 Detailed Description . . . . .	1742
11.158 geofunctions.h . . . . .	1742
11.159 glonass_l1_signal_replica.h . . . . .	1744
11.160 glonass_l2_signal_replica.h File Reference . . . . .	1745
11.160.1 Detailed Description . . . . .	1746
11.161 glonass_l2_signal_replica.h . . . . .	1746
11.162 gnss_circular_deque.h File Reference . . . . .	1747
11.162.1 Detailed Description . . . . .	1747
11.163 gnss_circular_deque.h . . . . .	1748
11.164 gnss_sdr_create_directory.h File Reference . . . . .	1749
11.164.1 Detailed Description . . . . .	1750
11.165 gnss_sdr_create_directory.h . . . . .	1750
11.166 gnss_sdr_fft.h File Reference . . . . .	1751
11.166.1 Detailed Description . . . . .	1751
11.166.2 Typedef Documentation . . . . .	1751
11.166.2.1 gnss_fft_complex_fwd . . . . .	1751
11.166.2.2 gnss_fft_complex_rev . . . . .	1752
11.166.2.3 gnss_fft_fwd_unique_ptr . . . . .	1752

11.166.2.4 gnss_fft_rev_unique_ptr . . . . .	1752
11.166.3 Function Documentation . . . . .	1752
11.166.3.1 gnss_fft_fwd_make_unique() . . . . .	1752
11.166.3.2 gnss_fft_rev_make_unique() . . . . .	1752
11.167 gnss_sdr_fft.h . . . . .	1752
11.168 gnss_sdr_filesystem.h File Reference . . . . .	1753
11.168.1 Detailed Description . . . . .	1754
11.169 gnss_sdr_filesystem.h . . . . .	1754
11.170 gnss_sdr_flags.h File Reference . . . . .	1755
11.170.1 Detailed Description . . . . .	1756
11.171 gnss_sdr_flags.h . . . . .	1756
11.172 gnss_sdr_make_unique.h File Reference . . . . .	1759
11.172.1 Detailed Description . . . . .	1759
11.173 gnss_sdr_make_unique.h . . . . .	1759
11.174 gnss_sdr_string_literals.h File Reference . . . . .	1760
11.174.1 Detailed Description . . . . .	1760
11.175 gnss_sdr_string_literals.h . . . . .	1760
11.176 gnss_signal_replica.h File Reference . . . . .	1761
11.176.1 Detailed Description . . . . .	1762
11.177 gnss_signal_replica.h . . . . .	1762
11.178 gnss_time.h File Reference . . . . .	1763
11.178.1 Detailed Description . . . . .	1764
11.179 gnss_time.h . . . . .	1764
11.180 gps_l2c_signal_replica.h File Reference . . . . .	1764
11.180.1 Detailed Description . . . . .	1765
11.181 gps_l2c_signal_replica.h . . . . .	1765
11.182 gps_l5_signal_replica.h File Reference . . . . .	1766
11.182.1 Detailed Description . . . . .	1766
11.183 gps_l5_signal_replica.h . . . . .	1766
11.184 gps_sdr_signal_replica.h File Reference . . . . .	1767
11.184.1 Detailed Description . . . . .	1768
11.185 gps_sdr_signal_replica.h . . . . .	1768
11.186 item_type_helpers.h File Reference . . . . .	1769
11.186.1 Detailed Description . . . . .	1770
11.187 item_type_helpers.h . . . . .	1770
11.188 cIFFT.h File Reference . . . . .	1771
11.188.1 Detailed Description . . . . .	1773
11.188.2 Typedef Documentation . . . . .	1773
11.188.2.1 cIFFT_Plan . . . . .	1773
11.188.3 Enumeration Type Documentation . . . . .	1773
11.188.3.1 cIFFT_DataFormat . . . . .	1773
11.188.3.2 cIFFT_Dimension . . . . .	1773

11.188.3.3 <code>clFFT_Direction</code> . . . . .	1773
11.189 <code>clFFT.h</code> . . . . .	1773
11.190 <code>fft_base_kernels.h</code> File Reference . . . . .	1774
11.190.1 Detailed Description . . . . .	1775
11.191 <code>fft_base_kernels.h</code> . . . . .	1775
11.192 <code>fft_internal.h</code> File Reference . . . . .	1778
11.192.1 Detailed Description . . . . .	1779
11.192.2 Enumeration Type Documentation . . . . .	1779
11.192.2.1 <code>kernel_dir_t</code> . . . . .	1779
11.193 <code>fft_internal.h</code> . . . . .	1779
11.194 <code>pass_through.h</code> File Reference . . . . .	1781
11.194.1 Detailed Description . . . . .	1782
11.195 <code>pass_through.h</code> . . . . .	1782
11.196 <code>rtklib.h</code> File Reference . . . . .	1783
11.196.1 Detailed Description . . . . .	1792
11.197 <code>rtklib.h</code> . . . . .	1792
11.198 <code>rtklib_conversions.h</code> File Reference . . . . .	1808
11.198.1 Detailed Description . . . . .	1809
11.199 <code>rtklib_conversions.h</code> . . . . .	1809
11.200 <code>rtklib_ephemeris.h</code> File Reference . . . . .	1810
11.200.1 Detailed Description . . . . .	1811
11.201 <code>rtklib_ephemeris.h</code> . . . . .	1812
11.202 <code>rtklib_ionex.h</code> File Reference . . . . .	1813
11.202.1 Detailed Description . . . . .	1813
11.202.2 Variable Documentation . . . . .	1814
11.202.2.1 <code>MIN_EL</code> . . . . .	1814
11.202.2.2 <code>MIN_HGT</code> . . . . .	1814
11.202.2.3 <code>VAR_NOTEC</code> . . . . .	1814
11.203 <code>rtklib_ionex.h</code> . . . . .	1814
11.204 <code>rtklib_lambda.h</code> File Reference . . . . .	1815
11.204.1 Detailed Description . . . . .	1815
11.204.2 Macro Definition Documentation . . . . .	1816
11.204.2.1 <code>ROUND_LAMBDA</code> . . . . .	1816
11.204.2.2 <code>SGN_LAMBDA</code> . . . . .	1816
11.204.2.3 <code>SWAP_LAMBDA</code> . . . . .	1816
11.204.3 Variable Documentation . . . . .	1816
11.204.3.1 <code>LOOPMAX</code> . . . . .	1816
11.205 <code>rtklib_lambda.h</code> . . . . .	1817
11.206 <code>rtklib_pntpos.h</code> File Reference . . . . .	1817
11.206.1 Detailed Description . . . . .	1819
11.206.2 Function Documentation . . . . .	1819
11.206.2.1 <code>pntpos()</code> . . . . .	1819

11.206.3 Variable Documentation	1819
11.206.3.1 ERR_ION	1819
11.206.3.2 ERR_TROP	1819
11.206.3.3 MAXITR	1820
11.206.3.4 NX	1820
11.206.4 of estimated parameters	1820
11.207 rtklib_pntpos.h	1820
11.208 rtklib_ppp.h File Reference	1822
11.208.1 Detailed Description	1823
11.208.2 Macro Definition Documentation	1823
11.208.2.1 MIN_PPP	1823
11.208.2.2 ROUND_PPP	1824
11.208.2.3 SWAP_D	1824
11.208.2.4 SWAP_I	1824
11.208.3 Variable Documentation	1824
11.208.3.1 CONST_AMB	1824
11.208.3.2 LOG_PI	1824
11.208.3.3 MIN_ARC_GAP	1824
11.208.3.4 SQRT2	1824
11.208.3.5 THRES_RES	1824
11.208.3.6 VAR_BIAS	1825
11.208.3.7 VAR_CLK	1825
11.208.3.8 VAR_GRA_PPP	1825
11.208.3.9 VAR_IONO_OFF	1825
11.208.3.10 VAR_POS_PPP	1825
11.208.3.11 VAR_ZTD	1825
11.209 rtklib_ppp.h	1825
11.210 rtklib_preceph.h File Reference	1827
11.210.1 Detailed Description	1828
11.210.2 Variable Documentation	1828
11.210.2.1 EXTERR_CLK	1828
11.210.2.2 EXTERR_EPH	1828
11.210.2.3 MAXDTE	1828
11.210.2.4 NMAX	1828
11.211 rtklib_preceph.h	1829
11.212 rtklib_rtcn.h File Reference	1830
11.212.1 Detailed Description	1830
11.212.2 Macro Definition Documentation	1831
11.212.2.1 RTCM2PREAMB	1831
11.212.2.2 RTCM3PREAMB	1831
11.213 rtklib_rtcn.h	1831
11.214 rtklib_rtcn2.h File Reference	1832

11.214.1 Detailed Description . . . . .	1833
11.215 rtklib_rtc2m2.h . . . . .	1833
11.216 rtklib_rtc3m3.h File Reference . . . . .	1834
11.216.1 Detailed Description . . . . .	1836
11.216.2 Variable Documentation . . . . .	1836
11.216.2.1 CODES_BDS . . . . .	1836
11.216.2.2 CODES_GAL . . . . .	1836
11.216.2.3 CODES_GLO . . . . .	1836
11.216.2.4 CODES_GPS . . . . .	1837
11.216.2.5 CODES_QZS . . . . .	1837
11.216.2.6 CODES_SBS . . . . .	1837
11.216.2.7 PRUNIT_GLO . . . . .	1837
11.216.2.8 PRUNIT_GPS . . . . .	1837
11.216.2.9 RANGE_MS . . . . .	1837
11.216.2.10 SSRUDINT . . . . .	1837
11.217 rtklib_rtc3m3.h . . . . .	1837
11.218 rtklib_rtkcmn.h File Reference . . . . .	1840
11.218.1 Detailed Description . . . . .	1844
11.218.2 Macro Definition Documentation . . . . .	1844
11.218.2.1 Rx . . . . .	1844
11.218.2.2 Ry . . . . .	1844
11.218.2.3 Rz . . . . .	1845
11.219 rtklib_rtkcmn.h . . . . .	1845
11.220 rtklib_rtkpos.h File Reference . . . . .	1848
11.220.1 Detailed Description . . . . .	1850
11.220.1.1 SPDX-License-Identifier: BSD-2-Clause . . . . .	1850
11.221 rtklib_rtkpos.h . . . . .	1850
11.222 rtklib_rtksvr.h File Reference . . . . .	1852
11.222.1 Detailed Description . . . . .	1853
11.222.2 Variable Documentation . . . . .	1854
11.222.2.1 PRCOPT_DEFAULT . . . . .	1854
11.222.2.2 SOLOPT_DEFAULT . . . . .	1854
11.223 rtklib_rtksvr.h . . . . .	1854
11.224 rtklib_sbass.h File Reference . . . . .	1856
11.224.1 Detailed Description . . . . .	1857
11.224.2 Variable Documentation . . . . .	1857
11.224.2.1 IGPBAND1 . . . . .	1857
11.224.2.2 IGPBAND2 . . . . .	1857
11.224.2.3 WEEKOFFSET . . . . .	1858
11.225 rtklib_sbass.h . . . . .	1858
11.226 rtklib_solution.h File Reference . . . . .	1859
11.226.1 Detailed Description . . . . .	1861

11.226.1.1 SPDX-License-Identifier: BSD-2-Clause . . . . .	1861
11.227 rtklib_solution.h . . . . .	1861
11.228 rtklib_stream.h File Reference . . . . .	1863
11.228.1 Detailed Description . . . . .	1866
11.228.2 Macro Definition Documentation . . . . .	1866
11.228.2.1 FTP_CMD . . . . .	1866
11.228.2.2 FTP_TIMEOUT . . . . .	1866
11.228.2.3 MAXCLI . . . . .	1866
11.228.2.4 MAXSTATMSG . . . . .	1866
11.228.2.5 NTRIP_AGENT . . . . .	1866
11.228.2.6 NTRIP_CLI_PORT . . . . .	1866
11.228.2.7 NTRIP_MAXRSP . . . . .	1866
11.228.2.8 NTRIP_MAXSTR . . . . .	1866
11.228.2.9 NTRIP_RSP_ERROR . . . . .	1867
11.228.2.10 NTRIP_RSP_HTTP . . . . .	1867
11.228.2.11 NTRIP_RSP_OK_CLI . . . . .	1867
11.228.2.12 NTRIP_RSP_OK_SVR . . . . .	1867
11.228.2.13 NTRIP_RSP_SRCTBL . . . . .	1867
11.228.2.14 NTRIP_RSP_TBLEND . . . . .	1867
11.228.2.15 NTRIP_SVR_PORT . . . . .	1867
11.228.2.16 SERIBUFFSIZE . . . . .	1867
11.228.2.17 TIMETAGH_LEN . . . . .	1867
11.228.2.18 TINTACT . . . . .	1867
11.228.2.19 VER_RTKLIB . . . . .	1867
11.229 rtklib_stream.h . . . . .	1868
11.230 rtklib_tides.h File Reference . . . . .	1870
11.230.1 Detailed Description . . . . .	1871
11.230.1.1 Conventions (2010), 2010 . . . . .	1871
11.230.2 Variable Documentation . . . . .	1871
11.230.2.1 GME . . . . .	1871
11.230.2.2 GMM . . . . .	1871
11.230.2.3 GMS . . . . .	1871
11.231 rtklib_tides.h . . . . .	1871
11.232 short_x2_to_cshort.h File Reference . . . . .	1872
11.232.1 Detailed Description . . . . .	1873
11.233 short_x2_to_cshort.h . . . . .	1873
11.234 trackingcmd.h File Reference . . . . .	1874
11.234.1 Detailed Description . . . . .	1875
11.235 trackingcmd.h . . . . .	1875
11.236 hybrid_observables.h File Reference . . . . .	1875
11.236.1 Detailed Description . . . . .	1876
11.237 hybrid_observables.h . . . . .	1876



11.238 hybrid_observables_gs.h File Reference . . . . .	1877
11.238.1 Detailed Description . . . . .	1878
11.239 hybrid_observables_gs.h . . . . .	1878
11.240 obs_conf.h File Reference . . . . .	1880
11.240.1 Detailed Description . . . . .	1881
11.241 obs_conf.h . . . . .	1881
11.242 rtklib_pvt.h File Reference . . . . .	1882
11.242.1 Detailed Description . . . . .	1882
11.243 rtklib_pvt.h . . . . .	1882
11.244 rtklib_pvt_gs.h File Reference . . . . .	1885
11.244.1 Detailed Description . . . . .	1886
11.245 rtklib_pvt_gs.h . . . . .	1887
11.246 an_packet_printer.h File Reference . . . . .	1890
11.246.1 Detailed Description . . . . .	1891
11.247 an_packet_printer.h . . . . .	1891
11.248 geohash.h File Reference . . . . .	1893
11.248.1 Detailed Description . . . . .	1893
11.249 geohash.h . . . . .	1893
11.250 geojson_printer.h File Reference . . . . .	1894
11.250.1 Detailed Description . . . . .	1895
11.251 geojson_printer.h . . . . .	1895
11.252 gpx_printer.h File Reference . . . . .	1896
11.252.1 Detailed Description . . . . .	1896
11.253 gpx_printer.h . . . . .	1897
11.254 has_simple_printer.h File Reference . . . . .	1897
11.254.1 Detailed Description . . . . .	1898
11.255 has_simple_printer.h . . . . .	1898
11.256 kml_printer.h File Reference . . . . .	1899
11.256.1 Detailed Description . . . . .	1899
11.257 kml_printer.h . . . . .	1900
11.258 monitor_ephemeris_udp_sink.h File Reference . . . . .	1900
11.258.1 Detailed Description . . . . .	1901
11.259 monitor_ephemeris_udp_sink.h . . . . .	1901
11.260 monitor_pvt.h File Reference . . . . .	1902
11.260.1 Detailed Description . . . . .	1903
11.260.1.1 Álvaro Cebrián Juan, 2019. <a href="mailto:acebrianjuan(at)gmail.com">acebrianjuan(at)gmail.com</a> . . . . .	1903
11.261 monitor_pvt.h . . . . .	1903
11.262 monitor_pvt_udp_sink.h File Reference . . . . .	1905
11.262.1 Detailed Description . . . . .	1906
11.263 monitor_pvt_udp_sink.h . . . . .	1906
11.264 nmea_printer.h File Reference . . . . .	1907
11.264.1 Detailed Description . . . . .	1907

11.265 nmea_printer.h . . . . .	1908
11.266 pvt_conf.h File Reference . . . . .	1909
11.266.1 Detailed Description . . . . .	1910
11.267 pvt_conf.h . . . . .	1910
11.268 pvt_kf.h File Reference . . . . .	1911
11.268.1 Detailed Description . . . . .	1912
11.269 pvt_kf.h . . . . .	1912
11.270 pvt_solution.h File Reference . . . . .	1913
11.270.1 Detailed Description . . . . .	1914
11.271 pvt_solution.h . . . . .	1914
11.272 rinex_printer.h File Reference . . . . .	1916
11.272.1 Detailed Description . . . . .	1916
11.273 rinex_printer.h . . . . .	1916
11.274 rtm.h File Reference . . . . .	1931
11.274.1 Detailed Description . . . . .	1932
11.275 rtm.h . . . . .	1932
11.276 rtm_printer.h File Reference . . . . .	1952
11.276.1 Detailed Description . . . . .	1953
11.277 rtm_printer.h . . . . .	1953
11.278 rtklib_solver.h File Reference . . . . .	1956
11.278.1 Detailed Description . . . . .	1957
11.279 rtklib_solver.h . . . . .	1957
11.280 serdes_galileo_eph.h File Reference . . . . .	1959
11.280.1 Detailed Description . . . . .	1960
11.281 serdes_galileo_eph.h . . . . .	1960
11.282 serdes_gps_eph.h File Reference . . . . .	1962
11.282.1 Detailed Description . . . . .	1963
11.283 serdes_gps_eph.h . . . . .	1963
11.284 serdes_monitor_pvt.h File Reference . . . . .	1966
11.284.1 Detailed Description . . . . .	1966
11.285 serdes_monitor_pvt.h . . . . .	1967
11.286 direct_resampler_conditioner.h File Reference . . . . .	1969
11.286.1 Detailed Description . . . . .	1969
11.287 direct_resampler_conditioner.h . . . . .	1969
11.288 mmse_resampler_conditioner.h File Reference . . . . .	1971
11.288.1 Detailed Description . . . . .	1971
11.289 mmse_resampler_conditioner.h . . . . .	1971
11.290 direct_resampler_conditioner_cb.h File Reference . . . . .	1972
11.290.1 Detailed Description . . . . .	1973
11.291 direct_resampler_conditioner_cb.h . . . . .	1973
11.292 direct_resampler_conditioner_cc.h File Reference . . . . .	1974
11.292.1 Detailed Description . . . . .	1975

11.293 direct_resampler_conditioner_cc.h . . . . .	1975
11.294 direct_resampler_conditioner_cs.h File Reference . . . . .	1976
11.294.1 Detailed Description . . . . .	1977
11.295 direct_resampler_conditioner_cs.h . . . . .	1977
11.296 signal_generator.h File Reference . . . . .	1978
11.296.1 Detailed Description . . . . .	1979
11.297 signal_generator.h . . . . .	1979
11.298 signal_generator_c.h File Reference . . . . .	1980
11.298.1 Detailed Description . . . . .	1981
11.298.2 Typedef Documentation . . . . .	1981
11.298.2.1 signal_generator_c_sptr . . . . .	1981
11.298.3 Function Documentation . . . . .	1981
11.298.3.1 signal_make_generator_c() . . . . .	1981
11.299 signal_generator_c.h . . . . .	1982
11.300 ad936x_custom_signal_source.h File Reference . . . . .	1983
11.300.1 Detailed Description . . . . .	1984
11.301 ad936x_custom_signal_source.h . . . . .	1984
11.302 adrv9361_z7035_signal_source_fpga.h File Reference . . . . .	1986
11.302.1 Detailed Description . . . . .	1986
11.303 adrv9361_z7035_signal_source_fpga.h . . . . .	1987
11.304 custom_udp_signal_source.h File Reference . . . . .	1988
11.304.1 Detailed Description . . . . .	1989
11.304.1.1 autotoc_md299 . . . . .	1989
11.305 custom_udp_signal_source.h . . . . .	1989
11.306 dma_signal_source_fpga.h File Reference . . . . .	1990
11.306.1 Detailed Description . . . . .	1991
11.307 dma_signal_source_fpga.h . . . . .	1991
11.308 fifo_signal_source.h File Reference . . . . .	1992
11.308.1 Detailed Description . . . . .	1993
11.309 fifo_signal_source.h . . . . .	1993
11.310 file_signal_source.h File Reference . . . . .	1994
11.310.1 Detailed Description . . . . .	1995
11.311 file_signal_source.h . . . . .	1995
11.312 file_source_base.h File Reference . . . . .	1995
11.312.1 Detailed Description . . . . .	1996
11.313 file_source_base.h . . . . .	1996
11.314 file_timestamp_signal_source.h File Reference . . . . .	1999
11.314.1 Detailed Description . . . . .	1999
11.315 file_timestamp_signal_source.h . . . . .	1999
11.316 flexiband_signal_source.h File Reference . . . . .	2000
11.316.1 Detailed Description . . . . .	2001
11.317 flexiband_signal_source.h . . . . .	2001

11.318 fmcomms2_signal_source.h File Reference	2002
11.318.1 Detailed Description	2002
11.318.1.1 This class represent a fmcomms2 signal source. It use the gr_iio block	2003
11.319 fmcomms2_signal_source.h	2003
11.320 fmcomms5_signal_source_fpga.h File Reference	2004
11.320.1 Detailed Description	2005
11.321 fmcomms5_signal_source_fpga.h	2005
11.322 four_bit_cpx_file_signal_source.h File Reference	2007
11.322.1 Detailed Description	2007
11.323 four_bit_cpx_file_signal_source.h	2007
11.324 gen_signal_source.h File Reference	2008
11.324.1 Detailed Description	2009
11.325 gen_signal_source.h	2009
11.326 ion_gsms_signal_source.h File Reference	2010
11.326.1 Detailed Description	2010
11.327 ion_gsms_signal_source.h	2010
11.328 labsat_signal_source.h File Reference	2011
11.328.1 Detailed Description	2012
11.329 labsat_signal_source.h	2012
11.330 limesdr_signal_source.h	2013
11.331 max2771_evkit_signal_source_fpga.h File Reference	2014
11.331.1 Detailed Description	2015
11.332 max2771_evkit_signal_source_fpga.h	2015
11.333 multichannel_file_signal_source.h File Reference	2017
11.333.1 Detailed Description	2018
11.334 multichannel_file_signal_source.h	2018
11.335 nsr_file_signal_source.h File Reference	2019
11.335.1 Detailed Description	2020
11.336 nsr_file_signal_source.h	2020
11.337 osmosdr_signal_source.h File Reference	2021
11.337.1 Detailed Description	2022
11.338 osmosdr_signal_source.h	2022
11.339 plutosdr_signal_source.h File Reference	2023
11.339.1 Detailed Description	2023
11.340 plutosdr_signal_source.h	2024
11.341 raw_array_signal_source.h File Reference	2025
11.341.1 Detailed Description	2025
11.342 raw_array_signal_source.h	2026
11.343 rtl_tcp_signal_source.h File Reference	2027
11.343.1 Detailed Description	2027
11.344 rtl_tcp_signal_source.h	2027
11.345 signal_source_base.h File Reference	2029

11.345.1 Detailed Description . . . . .	2029
11.346 <a href="#">signal_source_base.h</a> . . . . .	2029
11.347 <a href="#">spir_file_signal_source.h</a> File Reference . . . . .	2030
11.347.1 Detailed Description . . . . .	2030
11.348 <a href="#">spir_file_signal_source.h</a> . . . . .	2031
11.349 <a href="#">spir_gss6450_file_signal_source.h</a> File Reference . . . . .	2031
11.349.1 Detailed Description . . . . .	2032
11.350 <a href="#">spir_gss6450_file_signal_source.h</a> . . . . .	2032
11.351 <a href="#">two_bit_cpx_file_signal_source.h</a> File Reference . . . . .	2034
11.351.1 Detailed Description . . . . .	2034
11.352 <a href="#">two_bit_cpx_file_signal_source.h</a> . . . . .	2034
11.353 <a href="#">two_bit_packed_file_signal_source.h</a> File Reference . . . . .	2035
11.353.1 Detailed Description . . . . .	2036
11.354 <a href="#">two_bit_packed_file_signal_source.h</a> . . . . .	2036
11.355 <a href="#">uhd_signal_source.h</a> File Reference . . . . .	2037
11.355.1 Detailed Description . . . . .	2038
11.356 <a href="#">uhd_signal_source.h</a> . . . . .	2038
11.357 <a href="#">zmq_signal_source.h</a> File Reference . . . . .	2039
11.357.1 Detailed Description . . . . .	2040
11.358 <a href="#">zmq_signal_source.h</a> . . . . .	2040
11.359 <a href="#">ad936x_jio_source.h</a> File Reference . . . . .	2041
11.359.1 Detailed Description . . . . .	2042
11.360 <a href="#">ad936x_jio_source.h</a> . . . . .	2042
11.361 <a href="#">fifo_reader.h</a> File Reference . . . . .	2044
11.361.1 Detailed Description . . . . .	2045
11.362 <a href="#">fifo_reader.h</a> . . . . .	2045
11.363 <a href="#">gr_complex_ip_packet_source.h</a> File Reference . . . . .	2046
11.363.1 Detailed Description . . . . .	2047
11.364 <a href="#">gr_complex_ip_packet_source.h</a> . . . . .	2047
11.365 <a href="#">ion_gsms.h</a> File Reference . . . . .	2049
11.365.1 Detailed Description . . . . .	2049
11.366 <a href="#">ion_gsms.h</a> . . . . .	2050
11.367 <a href="#">labsat23_source.h</a> File Reference . . . . .	2050
11.367.1 Detailed Description . . . . .	2051
11.368 <a href="#">labsat23_source.h</a> . . . . .	2051
11.369 <a href="#">rtl_tcp_signal_source_c.h</a> File Reference . . . . .	2053
11.369.1 Detailed Description . . . . .	2054
11.370 <a href="#">rtl_tcp_signal_source_c.h</a> . . . . .	2054
11.371 <a href="#">unpack_2bit_samples.h</a> File Reference . . . . .	2055
11.371.1 Detailed Description . . . . .	2056
11.372 <a href="#">unpack_2bit_samples.h</a> . . . . .	2057
11.373 <a href="#">unpack_byte_2bit_cpx_samples.h</a> File Reference . . . . .	2058

11.373.1 Detailed Description . . . . .	2059
11.374 unpack_byte_2bit_cpx_samples.h . . . . .	2060
11.375 unpack_byte_2bit_samples.h File Reference . . . . .	2060
11.375.1 Detailed Description . . . . .	2061
11.376 unpack_byte_2bit_samples.h . . . . .	2061
11.377 unpack_byte_4bit_samples.h File Reference . . . . .	2062
11.377.1 Detailed Description . . . . .	2063
11.378 unpack_byte_4bit_samples.h . . . . .	2063
11.379 unpack_intspir_1bit_samples.h File Reference . . . . .	2064
11.379.1 Detailed Description . . . . .	2065
11.380 unpack_intspir_1bit_samples.h . . . . .	2065
11.381 unpack_short_byte_samples.h . . . . .	2066
11.382 unpack_spir_gss6450_samples.h File Reference . . . . .	2067
11.382.1 Detailed Description . . . . .	2068
11.383 unpack_spir_gss6450_samples.h . . . . .	2068
11.384 ad9361_manager.h File Reference . . . . .	2069
11.384.1 Detailed Description . . . . .	2070
11.385 ad9361_manager.h . . . . .	2070
11.386 ad936x_jio_custom.h File Reference . . . . .	2072
11.386.1 Detailed Description . . . . .	2073
11.386.1.1 autotoc_md383 . . . . .	2073
11.387 ad936x_jio_custom.h . . . . .	2073
11.388 ad936x_jio_samples.h File Reference . . . . .	2075
11.388.1 Detailed Description . . . . .	2076
11.388.2 Macro Definition Documentation . . . . .	2076
11.388.2.1 IIO_DEFAULTAD936XAPIFIFOSIZE_SAMPLES . . . . .	2076
11.388.2.2 IIO_INPUTTRAMFIFOSIZE . . . . .	2076
11.389 ad936x_jio_samples.h . . . . .	2077
11.390 fpga_buffer_monitor.h File Reference . . . . .	2077
11.390.1 Detailed Description . . . . .	2078
11.391 fpga_buffer_monitor.h . . . . .	2078
11.392 fpga_dma-proxy.h File Reference . . . . .	2079
11.392.1 Detailed Description . . . . .	2080
11.392.2 Enumeration Type Documentation . . . . .	2081
11.392.2.1 proxy_status . . . . .	2081
11.392.3 Variable Documentation . . . . .	2081
11.392.3.1 buffer . . . . .	2081
11.392.3.2 length . . . . .	2081
11.393 fpga_dma-proxy.h . . . . .	2081
11.394 fpga_dynamic_bit_selection.h File Reference . . . . .	2082
11.394.1 Detailed Description . . . . .	2083
11.395 fpga_dynamic_bit_selection.h . . . . .	2083

11.396 fpga_spidev.h File Reference	2084
11.396.1 Detailed Description	2085
11.397 fpga_spidev.h	2085
11.398 fpga_switch.h File Reference	2086
11.398.1 Detailed Description	2087
11.399 fpga_switch.h	2087
11.400 gnss_sdr_timestamp.h File Reference	2088
11.400.1 Detailed Description	2089
11.401 gnss_sdr_timestamp.h	2089
11.402 gnss_sdr_valve.h File Reference	2090
11.402.1 Detailed Description	2091
11.403 gnss_sdr_valve.h	2091
11.404 ion_gsms_chunk_data.h File Reference	2092
11.404.1 Detailed Description	2093
11.404.2 Function Documentation	2094
11.404.2.1 bits_to_item_size()	2094
11.404.2.2 with_word_type()	2094
11.405 ion_gsms_chunk_data.h	2094
11.406 ion_gsms_chunk_unpacking_ctx.h File Reference	2096
11.406.1 Detailed Description	2097
11.407 ion_gsms_chunk_unpacking_ctx.h	2097
11.408 ion_gsms_stream_encodings.h File Reference	2099
11.408.1 Detailed Description	2101
11.408.2 Typedef Documentation	2101
11.408.2.1 StreamEncoding	2101
11.408.3 Function Documentation	2101
11.408.3.1 encoding_from_string()	2101
11.408.4 Variable Documentation	2101
11.408.4.1 five_bit_look_up	2101
11.408.4.2 four_bit_look_up	2102
11.408.4.3 FP	2102
11.408.4.4 MS	2102
11.408.4.5 MSA	2102
11.408.4.6 OB	2102
11.408.4.7 OBA	2102
11.408.4.8 OG	2102
11.408.4.9 OGA	2102
11.408.4.10 SIGN	2103
11.408.4.11 SM	2103
11.408.4.12 SMA	2103
11.408.4.13 TC	2103
11.408.4.14 TCA	2103

11.408.4.15 three_bit_look_up . . . . .	2103
11.408.4.16 two_bit_look_up . . . . .	2103
11.409 ion_gsms_stream_encodings.h . . . . .	2104
11.410 pps_samplestamp.h File Reference . . . . .	2106
11.410.1 Detailed Description . . . . .	2107
11.410.1.1 autotoc_md407 . . . . .	2107
11.411 pps_samplestamp.h . . . . .	2107
11.412 ppstcprx.h File Reference . . . . .	2107
11.412.1 Detailed Description . . . . .	2108
11.412.1.1 autotoc_md409 . . . . .	2108
11.413 ppstcprx.h . . . . .	2108
11.414 rtl_tcp_commands.h File Reference . . . . .	2109
11.414.1 Detailed Description . . . . .	2110
11.414.1.1 <a href="https://git.osmocom.org/rtl-sdr">https://git.osmocom.org/rtl-sdr</a> . . . . .	2110
11.415 rtl_tcp_commands.h . . . . .	2110
11.416 rtl_tcp_dongle_info.h File Reference . . . . .	2111
11.416.1 Detailed Description . . . . .	2111
11.416.1.1 <a href="https://git.osmocom.org/rtl-sdr">https://git.osmocom.org/rtl-sdr</a> . . . . .	2111
11.417 rtl_tcp_dongle_info.h . . . . .	2112
11.418 beidou_b1i_telemetry_decoder.h File Reference . . . . .	2113
11.418.1 Detailed Description . . . . .	2113
11.419 beidou_b1i_telemetry_decoder.h . . . . .	2113
11.420 beidou_b3i_telemetry_decoder.h File Reference . . . . .	2114
11.420.1 Detailed Description . . . . .	2115
11.421 beidou_b3i_telemetry_decoder.h . . . . .	2115
11.422 galileo_e1b_telemetry_decoder.h File Reference . . . . .	2116
11.422.1 Detailed Description . . . . .	2117
11.423 galileo_e1b_telemetry_decoder.h . . . . .	2117
11.424 galileo_e5a_telemetry_decoder.h File Reference . . . . .	2118
11.424.1 Detailed Description . . . . .	2118
11.425 galileo_e5a_telemetry_decoder.h . . . . .	2119
11.426 galileo_e5b_telemetry_decoder.h File Reference . . . . .	2120
11.426.1 Detailed Description . . . . .	2120
11.427 galileo_e5b_telemetry_decoder.h . . . . .	2120
11.428 galileo_e6_telemetry_decoder.h File Reference . . . . .	2122
11.428.1 Detailed Description . . . . .	2122
11.429 galileo_e6_telemetry_decoder.h . . . . .	2122
11.430 glonass_l1_ca_telemetry_decoder.h File Reference . . . . .	2124
11.430.1 Detailed Description . . . . .	2124
11.431 glonass_l1_ca_telemetry_decoder.h . . . . .	2124
11.432 glonass_l2_ca_telemetry_decoder.h File Reference . . . . .	2126
11.432.1 Detailed Description . . . . .	2126



11.433 glonass_l2_ca_telemetry_decoder.h . . . . .	2126
11.434 gps_l1_ca_telemetry_decoder.h File Reference . . . . .	2127
11.434.1 Detailed Description . . . . .	2128
11.435 gps_l1_ca_telemetry_decoder.h . . . . .	2128
11.436 gps_l2c_telemetry_decoder.h File Reference . . . . .	2129
11.436.1 Detailed Description . . . . .	2130
11.437 gps_l2c_telemetry_decoder.h . . . . .	2130
11.438 gps_l5_telemetry_decoder.h File Reference . . . . .	2131
11.438.1 Detailed Description . . . . .	2131
11.439 gps_l5_telemetry_decoder.h . . . . .	2132
11.440 sbas_l1_telemetry_decoder.h File Reference . . . . .	2133
11.440.1 Detailed Description . . . . .	2133
11.441 sbas_l1_telemetry_decoder.h . . . . .	2133
11.442 beidou_b1i_telemetry_decoder_gs.h File Reference . . . . .	2135
11.442.1 Detailed Description . . . . .	2136
11.443 beidou_b1i_telemetry_decoder_gs.h . . . . .	2136
11.444 beidou_b3i_telemetry_decoder_gs.h File Reference . . . . .	2137
11.444.1 Detailed Description . . . . .	2138
11.445 beidou_b3i_telemetry_decoder_gs.h . . . . .	2139
11.446 galileo_telemetry_decoder_gs.h File Reference . . . . .	2140
11.446.1 Detailed Description . . . . .	2141
11.447 galileo_telemetry_decoder_gs.h . . . . .	2141
11.448 glonass_l1_ca_telemetry_decoder_gs.h File Reference . . . . .	2143
11.448.1 Detailed Description . . . . .	2144
11.449 glonass_l1_ca_telemetry_decoder_gs.h . . . . .	2145
11.450 glonass_l2_ca_telemetry_decoder_gs.h File Reference . . . . .	2146
11.450.1 Detailed Description . . . . .	2147
11.451 glonass_l2_ca_telemetry_decoder_gs.h . . . . .	2147
11.452 gps_l1_ca_telemetry_decoder_gs.h File Reference . . . . .	2149
11.452.1 Detailed Description . . . . .	2150
11.452.1.1 autotoc_md449 . . . . .	2150
11.453 gps_l1_ca_telemetry_decoder_gs.h . . . . .	2150
11.454 gps_l2c_telemetry_decoder_gs.h File Reference . . . . .	2152
11.454.1 Detailed Description . . . . .	2153
11.455 gps_l2c_telemetry_decoder_gs.h . . . . .	2153
11.456 gps_l5_telemetry_decoder_gs.h File Reference . . . . .	2154
11.456.1 Detailed Description . . . . .	2155
11.456.1.1 autotoc_md453 . . . . .	2155
11.457 gps_l5_telemetry_decoder_gs.h . . . . .	2156
11.458 sbas_l1_telemetry_decoder_gs.h File Reference . . . . .	2157
11.458.1 Detailed Description . . . . .	2158
11.459 sbas_l1_telemetry_decoder_gs.h . . . . .	2158

11.460 bits.h File Reference . . . . .	2160
11.460.1 Detailed Description . . . . .	2161
11.461 bits.h . . . . .	2161
11.462 cnav_msg.h File Reference . . . . .	2162
11.462.1 Detailed Description . . . . .	2163
11.463 cnav_msg.h . . . . .	2163
11.464 edc.h File Reference . . . . .	2165
11.464.1 Detailed Description . . . . .	2165
11.465 edc.h . . . . .	2166
11.466 fec.h File Reference . . . . .	2166
11.466.1 Detailed Description . . . . .	2167
11.466.2 Macro Definition Documentation . . . . .	2168
11.466.2.1 V27POLYA . . . . .	2168
11.466.2.2 V27POLYB . . . . .	2168
11.467 fec.h . . . . .	2168
11.468 swift_common.h File Reference . . . . .	2169
11.468.1 Detailed Description . . . . .	2169
11.469 swift_common.h . . . . .	2170
11.470 tlm_conf.h File Reference . . . . .	2170
11.470.1 Detailed Description . . . . .	2171
11.471 tlm_conf.h . . . . .	2171
11.472 tlm_crc_stats.h File Reference . . . . .	2172
11.472.1 Detailed Description . . . . .	2173
11.473 tlm_crc_stats.h . . . . .	2173
11.474 tlm_utils.h File Reference . . . . .	2174
11.474.1 Detailed Description . . . . .	2174
11.475 tlm_utils.h . . . . .	2174
11.476 viterbi_decoder.h File Reference . . . . .	2175
11.476.1 Detailed Description . . . . .	2175
11.477 viterbi_decoder.h . . . . .	2175
11.478 viterbi_decoder_sbass.h File Reference . . . . .	2177
11.478.1 Detailed Description . . . . .	2177
11.479 viterbi_decoder_sbass.h . . . . .	2178
11.480 beidou_b1i_dll_pll_tracking.h File Reference . . . . .	2179
11.480.1 Detailed Description . . . . .	2180
11.481 beidou_b1i_dll_pll_tracking.h . . . . .	2180
11.482 beidou_b3i_dll_pll_tracking.h File Reference . . . . .	2181
11.482.1 Detailed Description . . . . .	2181
11.483 beidou_b3i_dll_pll_tracking.h . . . . .	2182
11.484 galileo_e1_dll_pll_veml_tracking.h File Reference . . . . .	2183
11.484.1 Detailed Description . . . . .	2183
11.485 galileo_e1_dll_pll_veml_tracking.h . . . . .	2183

11.486 <a href="#">galileo_e1_dll_pll_veml_tracking_fpga.h</a> File Reference	2185
11.486.1 Detailed Description	2185
11.487 <a href="#">galileo_e1_dll_pll_veml_tracking_fpga.h</a>	2185
11.488 <a href="#">galileo_e1_tcp_connector_tracking.h</a> File Reference	2187
11.488.1 Detailed Description	2188
11.489 <a href="#">galileo_e1_tcp_connector_tracking.h</a>	2188
11.490 <a href="#">galileo_e5a_dll_pll_tracking.h</a> File Reference	2189
11.490.1 Detailed Description	2189
11.491 <a href="#">galileo_e5a_dll_pll_tracking.h</a>	2190
11.492 <a href="#">galileo_e5a_dll_pll_tracking_fpga.h</a> File Reference	2191
11.492.1 Detailed Description	2191
11.493 <a href="#">galileo_e5a_dll_pll_tracking_fpga.h</a>	2191
11.494 <a href="#">galileo_e5b_dll_pll_tracking.h</a> File Reference	2193
11.494.1 Detailed Description	2194
11.495 <a href="#">galileo_e5b_dll_pll_tracking.h</a>	2194
11.496 <a href="#">galileo_e6_dll_pll_tracking.h</a> File Reference	2195
11.496.1 Detailed Description	2196
11.497 <a href="#">galileo_e6_dll_pll_tracking.h</a>	2196
11.498 <a href="#">glonass_l1_ca_dll_pll_c_aid_tracking.h</a> File Reference	2197
11.498.1 Detailed Description	2198
11.499 <a href="#">glonass_l1_ca_dll_pll_c_aid_tracking.h</a>	2198
11.500 <a href="#">glonass_l1_ca_dll_pll_tracking.h</a> File Reference	2199
11.500.1 Detailed Description	2200
11.501 <a href="#">glonass_l1_ca_dll_pll_tracking.h</a>	2200
11.502 <a href="#">glonass_l2_ca_dll_pll_c_aid_tracking.h</a> File Reference	2201
11.502.1 Detailed Description	2202
11.503 <a href="#">glonass_l2_ca_dll_pll_c_aid_tracking.h</a>	2202
11.504 <a href="#">glonass_l2_ca_dll_pll_tracking.h</a> File Reference	2203
11.504.1 Detailed Description	2204
11.505 <a href="#">glonass_l2_ca_dll_pll_tracking.h</a>	2204
11.506 <a href="#">gps_l1_ca_dll_pll_tracking.h</a> File Reference	2205
11.506.1 Detailed Description	2206
11.507 <a href="#">gps_l1_ca_dll_pll_tracking.h</a>	2206
11.508 <a href="#">gps_l1_ca_dll_pll_tracking_fpga.h</a> File Reference	2207
11.508.1 Detailed Description	2208
11.509 <a href="#">gps_l1_ca_dll_pll_tracking_fpga.h</a>	2208
11.510 <a href="#">gps_l1_ca_dll_pll_tracking_gpu.h</a> File Reference	2210
11.510.1 Detailed Description	2210
11.511 <a href="#">gps_l1_ca_dll_pll_tracking_gpu.h</a>	2210
11.512 <a href="#">gps_l1_ca_gaussian_tracking.h</a> File Reference	2212
11.512.1 Detailed Description	2212
11.513 <a href="#">gps_l1_ca_gaussian_tracking.h</a>	2212

11.514 <a href="#">gps_l1_ca_kf_tracking.h File Reference</a> . . . . .	2214
11.514.1 Detailed Description . . . . .	2214
11.515 <a href="#">gps_l1_ca_kf_tracking.h</a> . . . . .	2214
11.516 <a href="#">gps_l1_ca_tcp_connector_tracking.h File Reference</a> . . . . .	2215
11.516.1 Detailed Description . . . . .	2216
11.517 <a href="#">gps_l1_ca_tcp_connector_tracking.h</a> . . . . .	2216
11.518 <a href="#">gps_l2_m_dll_pll_tracking.h File Reference</a> . . . . .	2217
11.518.1 Detailed Description . . . . .	2218
11.519 <a href="#">gps_l2_m_dll_pll_tracking.h</a> . . . . .	2218
11.520 <a href="#">gps_l2_m_dll_pll_tracking_fpga.h File Reference</a> . . . . .	2219
11.520.1 Detailed Description . . . . .	2220
11.521 <a href="#">gps_l2_m_dll_pll_tracking_fpga.h</a> . . . . .	2220
11.522 <a href="#">gps_l5_dll_pll_tracking.h File Reference</a> . . . . .	2221
11.522.1 Detailed Description . . . . .	2222
11.523 <a href="#">gps_l5_dll_pll_tracking.h</a> . . . . .	2222
11.524 <a href="#">gps_l5_dll_pll_tracking_fpga.h File Reference</a> . . . . .	2223
11.524.1 Detailed Description . . . . .	2224
11.525 <a href="#">gps_l5_dll_pll_tracking_fpga.h</a> . . . . .	2224
11.526 <a href="#">dll_pll_veml_tracking.h File Reference</a> . . . . .	2226
11.526.1 Detailed Description . . . . .	2227
11.527 <a href="#">dll_pll_veml_tracking.h</a> . . . . .	2227
11.528 <a href="#">dll_pll_veml_tracking_fpga.h File Reference</a> . . . . .	2229
11.528.1 Detailed Description . . . . .	2230
11.529 <a href="#">dll_pll_veml_tracking_fpga.h</a> . . . . .	2230
11.530 <a href="#">galileo_e1_tcp_connector_tracking_cc.h File Reference</a> . . . . .	2233
11.530.1 Detailed Description . . . . .	2234
11.531 <a href="#">galileo_e1_tcp_connector_tracking_cc.h</a> . . . . .	2235
11.532 <a href="#">glonass_l1_ca_dll_pll_c_aid_tracking_cc.h File Reference</a> . . . . .	2237
11.532.1 Detailed Description . . . . .	2238
11.533 <a href="#">glonass_l1_ca_dll_pll_c_aid_tracking_cc.h</a> . . . . .	2238
11.534 <a href="#">glonass_l1_ca_dll_pll_c_aid_tracking_sc.h File Reference</a> . . . . .	2240
11.534.1 Detailed Description . . . . .	2241
11.535 <a href="#">glonass_l1_ca_dll_pll_c_aid_tracking_sc.h</a> . . . . .	2242
11.536 <a href="#">glonass_l1_ca_dll_pll_tracking_cc.h File Reference</a> . . . . .	2244
11.536.1 Detailed Description . . . . .	2245
11.537 <a href="#">glonass_l1_ca_dll_pll_tracking_cc.h</a> . . . . .	2246
11.538 <a href="#">glonass_l2_ca_dll_pll_c_aid_tracking_cc.h File Reference</a> . . . . .	2248
11.538.1 Detailed Description . . . . .	2249
11.539 <a href="#">glonass_l2_ca_dll_pll_c_aid_tracking_cc.h</a> . . . . .	2249
11.540 <a href="#">glonass_l2_ca_dll_pll_c_aid_tracking_sc.h File Reference</a> . . . . .	2251
11.540.1 Detailed Description . . . . .	2252
11.541 <a href="#">glonass_l2_ca_dll_pll_c_aid_tracking_sc.h</a> . . . . .	2253

11.542 glonass_l2_ca_dll_pll_tracking_cc.h File Reference	2255
11.542.1 Detailed Description	2256
11.543 glonass_l2_ca_dll_pll_tracking_cc.h	2256
11.544 gps_l1_ca_dll_pll_tracking_gpu_cc.h File Reference	2258
11.544.1 Detailed Description	2259
11.545 gps_l1_ca_dll_pll_tracking_gpu_cc.h	2260
11.546 gps_l1_ca_gaussian_tracking_cc.h File Reference	2262
11.546.1 Detailed Description	2263
11.547 gps_l1_ca_gaussian_tracking_cc.h	2263
11.548 gps_l1_ca_tcp_connector_tracking_cc.h File Reference	2266
11.548.1 Detailed Description	2266
11.549 gps_l1_ca_tcp_connector_tracking_cc.h	2267
11.550 kf_tracking.h	2269
11.551 bayesian_estimation.h File Reference	2271
11.551.1 Detailed Description	2272
11.551.1.1	2273
11.552 bayesian_estimation.h	2273
11.553 cpu_multicorrelator.h File Reference	2274
11.553.1 Detailed Description	2274
11.554 cpu_multicorrelator.h	2275
11.555 cpu_multicorrelator_16sc.h File Reference	2276
11.555.1 Detailed Description	2276
11.556 cpu_multicorrelator_16sc.h	2277
11.557 cpu_multicorrelator_real_codes.h File Reference	2277
11.557.1 Detailed Description	2278
11.558 cpu_multicorrelator_real_codes.h	2278
11.559 cuda_multicorrelator.h File Reference	2279
11.559.1 Detailed Description	2280
11.559.2 Macro Definition Documentation	2281
11.559.2.1 CUDA_CALLABLE_MEMBER_DEVICE	2281
11.559.2.2 CUDA_CALLABLE_MEMBER_GLOBAL	2281
11.560 cuda_multicorrelator.h	2281
11.561 dll_pll_conf.h File Reference	2283
11.561.1 Detailed Description	2283
11.562 dll_pll_conf.h	2284
11.563 dll_pll_conf_fpga.h File Reference	2285
11.563.1 Detailed Description	2286
11.564 dll_pll_conf_fpga.h	2286
11.565 exponential_smoother.h File Reference	2287
11.565.1 Detailed Description	2288
11.566 exponential_smoother.h	2288
11.567 fpga_multicorrelator.h File Reference	2289

11.567.1 Detailed Description . . . . .	2289
11.568 fpga_multicorrelator.h . . . . .	2290
11.569 kf_conf.h . . . . .	2293
11.570 lock_detectors.h File Reference . . . . .	2294
11.570.1 Detailed Description . . . . .	2294
11.571 lock_detectors.h . . . . .	2295
11.572 nonlinear_tracking.h File Reference . . . . .	2296
11.572.1 Detailed Description . . . . .	2297
11.572.1.1 . . . . .	2297
11.573 nonlinear_tracking.h . . . . .	2297
11.574 tcp_communication.h File Reference . . . . .	2299
11.574.1 Detailed Description . . . . .	2300
11.575 tcp_communication.h . . . . .	2300
11.576 tcp_packet_data.h File Reference . . . . .	2301
11.576.1 Detailed Description . . . . .	2301
11.577 tcp_packet_data.h . . . . .	2302
11.578 tracking_2nd_DLL_filter.h File Reference . . . . .	2302
11.578.1 Detailed Description . . . . .	2302
11.579 tracking_2nd_DLL_filter.h . . . . .	2303
11.580 tracking_2nd_PLL_filter.h File Reference . . . . .	2304
11.580.1 Detailed Description . . . . .	2304
11.581 tracking_2nd_PLL_filter.h . . . . .	2304
11.582 tracking_discriminators.h File Reference . . . . .	2305
11.582.1 Detailed Description . . . . .	2306
11.583 tracking_discriminators.h . . . . .	2306
11.584 tracking_FLL_PLL_filter.h File Reference . . . . .	2309
11.584.1 Detailed Description . . . . .	2309
11.585 tracking_FLL_PLL_filter.h . . . . .	2309
11.586 tracking_loop_filter.h File Reference . . . . .	2310
11.586.1 Detailed Description . . . . .	2311
11.587 tracking_loop_filter.h . . . . .	2311
11.588 acquisition_interface.h File Reference . . . . .	2312
11.588.1 Detailed Description . . . . .	2312
11.589 acquisition_interface.h . . . . .	2313
11.590 channel_interface.h File Reference . . . . .	2314
11.590.1 Detailed Description . . . . .	2314
11.591 channel_interface.h . . . . .	2315
11.592 configuration_interface.h File Reference . . . . .	2315
11.592.1 Detailed Description . . . . .	2316
11.593 configuration_interface.h . . . . .	2316
11.594 gnss_block_interface.h File Reference . . . . .	2317
11.594.1 Detailed Description . . . . .	2318

11.595 gnss_block_interface.h . . . . .	2318
11.596 observables_interface.h File Reference . . . . .	2319
11.596.1 Detailed Description . . . . .	2320
11.597 observables_interface.h . . . . .	2320
11.598 pvt_interface.h File Reference . . . . .	2321
11.598.1 Detailed Description . . . . .	2322
11.599 pvt_interface.h . . . . .	2322
11.600 signal_source_interface.h . . . . .	2323
11.601 telemetry_decoder_interface.h File Reference . . . . .	2324
11.601.1 Detailed Description . . . . .	2324
11.602 telemetry_decoder_interface.h . . . . .	2324
11.603 tracking_interface.h File Reference . . . . .	2325
11.603.1 Detailed Description . . . . .	2325
11.604 tracking_interface.h . . . . .	2326
11.605 channel_event.h File Reference . . . . .	2326
11.605.1 Detailed Description . . . . .	2327
11.606 channel_event.h . . . . .	2328
11.607 channel_status_msg_receiver.h File Reference . . . . .	2328
11.607.1 Detailed Description . . . . .	2329
11.608 channel_status_msg_receiver.h . . . . .	2329
11.609 command_event.h File Reference . . . . .	2330
11.609.1 Detailed Description . . . . .	2331
11.610 command_event.h . . . . .	2331
11.611 galileo_e6_has_msg_receiver.h File Reference . . . . .	2332
11.611.1 Detailed Description . . . . .	2333
11.612 galileo_e6_has_msg_receiver.h . . . . .	2333
11.613 galileo_tow_map.h File Reference . . . . .	2335
11.613.1 Detailed Description . . . . .	2336
11.614 galileo_tow_map.h . . . . .	2336
11.615 gnss_crypto.h File Reference . . . . .	2337
11.615.1 Detailed Description . . . . .	2337
11.616 gnss_crypto.h . . . . .	2337
11.617 gnss_sdr_fpga_sample_counter.h File Reference . . . . .	2339
11.617.1 Detailed Description . . . . .	2339
11.618 gnss_sdr_fpga_sample_counter.h . . . . .	2340
11.619 gnss_sdr_sample_counter.h File Reference . . . . .	2341
11.619.1 Detailed Description . . . . .	2342
11.620 gnss_sdr_sample_counter.h . . . . .	2342
11.621 gnss_sdr_supl_client.h File Reference . . . . .	2343
11.621.1 Detailed Description . . . . .	2344
11.622 gnss_sdr_supl_client.h . . . . .	2344
11.623 gnss_sdr_time_counter.h File Reference . . . . .	2347

11.623.1 Detailed Description	2348
11.624 gnss_sdr_time_counter.h	2348
11.625 ini.h File Reference	2349
11.625.1 Detailed Description	2349
11.625.1.1 <a href="https://github.com/benhoyt/inih">https://github.com/benhoyt/inih</a>	2350
11.626 ini.h	2350
11.627 INIReader.h File Reference	2351
11.627.1 Detailed Description	2351
11.627.1.1 <a href="https://github.com/benhoyt/inih">https://github.com/benhoyt/inih</a>	2352
11.628 INIReader.h	2352
11.629 nav_message_monitor.h File Reference	2353
11.629.1 Detailed Description	2353
11.630 nav_message_monitor.h	2354
11.631 nav_message_packet.h File Reference	2354
11.631.1 Detailed Description	2355
11.632 nav_message_packet.h	2355
11.633 nav_message_udp_sink.h File Reference	2356
11.633.1 Detailed Description	2357
11.634 nav_message_udp_sink.h	2357
11.635 osnma_helper.h File Reference	2358
11.635.1 Detailed Description	2359
11.636 osnma_helper.h	2359
11.637 osnma_msg_receiver.h File Reference	2360
11.637.1 Detailed Description	2361
11.637.2 Macro Definition Documentation	2361
11.637.2.1 FRIEND_TEST	2361
11.638 osnma_msg_receiver.h	2361
11.639 osnma_nav_data_manager.h File Reference	2364
11.639.1 Detailed Description	2364
11.640 osnma_nav_data_manager.h	2365
11.641 serdes_nav_message.h File Reference	2365
11.641.1 Detailed Description	2366
11.642 serdes_nav_message.h	2366
11.643 string_converter.h File Reference	2368
11.643.1 Detailed Description	2369
11.644 string_converter.h	2369
11.645 uio_fpga.h File Reference	2370
11.645.1 Detailed Description	2370
11.646 uio_fpga.h	2371
11.647 gnss_synchro_monitor.h File Reference	2371
11.647.1 Detailed Description	2372
11.648 gnss_synchro_monitor.h	2372



11.649 gnss_synchro_udp_sink.h File Reference . . . . .	2373
11.649.1 Detailed Description . . . . .	2374
11.650 gnss_synchro_udp_sink.h . . . . .	2374
11.651 serdes_gnss_synchro.h File Reference . . . . .	2375
11.651.1 Detailed Description . . . . .	2376
11.652 serdes_gnss_synchro.h . . . . .	2376
11.653 concurrent_map.h File Reference . . . . .	2379
11.653.1 Detailed Description . . . . .	2379
11.654 concurrent_map.h . . . . .	2379
11.655 concurrent_queue.h File Reference . . . . .	2380
11.655.1 Detailed Description . . . . .	2381
11.656 concurrent_queue.h . . . . .	2381
11.657 control_thread.h File Reference . . . . .	2383
11.657.1 Detailed Description . . . . .	2383
11.658 control_thread.h . . . . .	2383
11.659 file_configuration.h File Reference . . . . .	2386
11.659.1 Detailed Description . . . . .	2387
11.660 file_configuration.h . . . . .	2387
11.661 gnss_block_factory.h File Reference . . . . .	2388
11.661.1 Detailed Description . . . . .	2389
11.662 gnss_block_factory.h . . . . .	2389
11.663 gnss_flowgraph.h File Reference . . . . .	2390
11.663.1 Detailed Description . . . . .	2391
11.664 gnss_flowgraph.h . . . . .	2391
11.665 in_memory_configuration.h File Reference . . . . .	2395
11.665.1 Detailed Description . . . . .	2395
11.666 in_memory_configuration.h . . . . .	2396
11.667 tcp_cmd_interface.h File Reference . . . . .	2397
11.667.1 Detailed Description . . . . .	2397
11.668 tcp_cmd_interface.h . . . . .	2397
11.669 agnss_ref_location.h File Reference . . . . .	2399
11.669.1 Detailed Description . . . . .	2399
11.670 agnss_ref_location.h . . . . .	2400
11.671 agnss_ref_time.h File Reference . . . . .	2401
11.671.1 Detailed Description . . . . .	2401
11.672 agnss_ref_time.h . . . . .	2402
11.673 Beidou_B1I.h File Reference . . . . .	2402
11.673.1 Detailed Description . . . . .	2404
11.674 Beidou_B1I.h . . . . .	2404
11.675 Beidou_B3I.h File Reference . . . . .	2405
11.675.1 Detailed Description . . . . .	2407
11.676 Beidou_B3I.h . . . . .	2407

11.677 Beidou_DNAV.h File Reference	2407
11.677.1 Detailed Description	2412
11.678 Beidou_DNAV.h	2412
11.679 beidou_dnav_almanac.h File Reference	2416
11.679.1 Detailed Description	2417
11.680 beidou_dnav_almanac.h	2417
11.681 beidou_dnav_ephemeris.h File Reference	2418
11.681.1 Detailed Description	2419
11.682 beidou_dnav_ephemeris.h	2420
11.683 beidou_dnav_iono.h File Reference	2421
11.683.1 Detailed Description	2423
11.684 beidou_dnav_iono.h	2423
11.685 beidou_dnav_navigation_message.h File Reference	2423
11.685.1 Detailed Description	2424
11.686 beidou_dnav_navigation_message.h	2424
11.687 beidou_dnav_utc_model.h File Reference	2428
11.687.1 Detailed Description	2429
11.688 beidou_dnav_utc_model.h	2430
11.689 display.h File Reference	2431
11.689.1 Detailed Description	2432
11.690 display.h	2432
11.691 galileo_almanac.h File Reference	2433
11.691.1 Detailed Description	2434
11.692 galileo_almanac.h	2434
11.693 galileo_almanac_helper.h File Reference	2435
11.693.1 Detailed Description	2436
11.693.1.1 autotoc_md688	2436
11.694 galileo_almanac_helper.h	2436
11.695 Galileo_CNAV.h File Reference	2437
11.695.1 Detailed Description	2440
11.696 Galileo_CNAV.h	2440
11.697 galileo_cnav_message.h File Reference	2442
11.697.1 Detailed Description	2443
11.698 galileo_cnav_message.h	2443
11.699 Galileo_E1.h File Reference	2445
11.699.1 Detailed Description	2446
11.700 Galileo_E1.h	2446
11.701 Galileo_E5a.h File Reference	2463
11.701.1 Detailed Description	2465
11.702 Galileo_E5a.h	2465
11.703 Galileo_E5b.h File Reference	2507
11.703.1 Detailed Description	2509

11.704 Galileo_E5b.h . . . . .	2509
11.705 Galileo_E6.h File Reference . . . . .	2551
11.705.1 Detailed Description . . . . .	2552
11.706 Galileo_E6.h . . . . .	2553
11.707 galileo_ephemeris.h File Reference . . . . .	2575
11.707.1 Detailed Description . . . . .	2576
11.708 galileo_ephemeris.h . . . . .	2576
11.709 Galileo_FNAV.h File Reference . . . . .	2577
11.709.1 Detailed Description . . . . .	2581
11.710 Galileo_FNAV.h . . . . .	2581
11.711 galileo_fnav_message.h File Reference . . . . .	2583
11.711.1 Detailed Description . . . . .	2584
11.712 galileo_fnav_message.h . . . . .	2584
11.713 galileo_has_data.h File Reference . . . . .	2588
11.713.1 Detailed Description . . . . .	2588
11.714 galileo_has_data.h . . . . .	2589
11.715 galileo_has_page.h File Reference . . . . .	2591
11.715.1 Detailed Description . . . . .	2592
11.716 galileo_has_page.h . . . . .	2592
11.717 Galileo_INAV.h File Reference . . . . .	2593
11.717.1 Detailed Description . . . . .	2598
11.718 Galileo_INAV.h . . . . .	2598
11.719 galileo_inav_message.h File Reference . . . . .	2602
11.719.1 Detailed Description . . . . .	2603
11.720 galileo_inav_message.h . . . . .	2603
11.721 galileo_iono.h File Reference . . . . .	2609
11.721.1 Detailed Description . . . . .	2610
11.722 galileo_iono.h . . . . .	2610
11.723 galileo_ism.h File Reference . . . . .	2611
11.723.1 Detailed Description . . . . .	2612
11.724 galileo_ism.h . . . . .	2612
11.725 Galileo_OSNMA.h File Reference . . . . .	2615
11.725.1 Detailed Description . . . . .	2616
11.726 Galileo_OSNMA.h . . . . .	2616
11.727 galileo_reduced_ced.h File Reference . . . . .	2618
11.727.1 Detailed Description . . . . .	2619
11.728 galileo_reduced_ced.h . . . . .	2619
11.729 galileo_utc_model.h File Reference . . . . .	2620
11.729.1 Detailed Description . . . . .	2621
11.730 galileo_utc_model.h . . . . .	2621
11.731 glonass_gnav_almanac.h File Reference . . . . .	2622
11.731.1 Detailed Description . . . . .	2623

11.732 glonass_gnav_almanac.h . . . . .	2624
11.733 glonass_gnav_ephemeris.h File Reference . . . . .	2625
11.733.1 Detailed Description . . . . .	2626
11.734 glonass_gnav_ephemeris.h . . . . .	2627
11.735 glonass_gnav_navigation_message.h File Reference . . . . .	2629
11.735.1 Detailed Description . . . . .	2630
11.736 glonass_gnav_navigation_message.h . . . . .	2631
11.737 glonass_gnav_utc_model.h File Reference . . . . .	2633
11.737.1 Detailed Description . . . . .	2634
11.738 glonass_gnav_utc_model.h . . . . .	2635
11.739 GLONASS_L1_L2_CA.h File Reference . . . . .	2636
11.739.1 Detailed Description . . . . .	2641
11.740 GLONASS_L1_L2_CA.h . . . . .	2641
11.741 gnss_almanac.h File Reference . . . . .	2645
11.741.1 Detailed Description . . . . .	2646
11.742 gnss_almanac.h . . . . .	2646
11.743 gnss_ephemeris.h File Reference . . . . .	2647
11.743.1 Detailed Description . . . . .	2648
11.744 gnss_ephemeris.h . . . . .	2648
11.745 gnss_frequencies.h File Reference . . . . .	2650
11.745.1 Detailed Description . . . . .	2651
11.746 gnss_frequencies.h . . . . .	2651
11.747 gnss_obs_codes.h File Reference . . . . .	2652
11.747.1 Detailed Description . . . . .	2655
11.748 gnss_obs_codes.h . . . . .	2655
11.749 gnss_satellite.h File Reference . . . . .	2656
11.749.1 Detailed Description . . . . .	2657
11.750 gnss_satellite.h . . . . .	2657
11.751 gnss_signal.h File Reference . . . . .	2658
11.751.1 Detailed Description . . . . .	2659
11.752 gnss_signal.h . . . . .	2660
11.753 gnss_synchro.h File Reference . . . . .	2660
11.753.1 Detailed Description . . . . .	2661
11.754 gnss_synchro.h . . . . .	2661
11.755 gps_acq_assist.h File Reference . . . . .	2664
11.755.1 Detailed Description . . . . .	2665
11.756 gps_acq_assist.h . . . . .	2665
11.757 gps_almanac.h File Reference . . . . .	2666
11.757.1 Detailed Description . . . . .	2667
11.758 gps_almanac.h . . . . .	2667
11.759 GPS_CNAV.h File Reference . . . . .	2668
11.759.1 Detailed Description . . . . .	2671

11.760 GPS_CNAV.h . . . . .	2671
11.761 gps_cnav_ephemeris.h File Reference . . . . .	2673
11.761.1 Detailed Description . . . . .	2674
11.762 gps_cnav_ephemeris.h . . . . .	2675
11.763 gps_cnav_iono.h File Reference . . . . .	2676
11.763.1 Detailed Description . . . . .	2677
11.764 gps_cnav_iono.h . . . . .	2678
11.765 gps_cnav_navigation_message.h File Reference . . . . .	2678
11.765.1 Detailed Description . . . . .	2679
11.766 gps_cnav_navigation_message.h . . . . .	2679
11.767 gps_cnav_utc_model.h File Reference . . . . .	2681
11.767.1 Detailed Description . . . . .	2682
11.768 gps_cnav_utc_model.h . . . . .	2682
11.769 gps_ephemeris.h File Reference . . . . .	2682
11.769.1 Detailed Description . . . . .	2683
11.770 gps_ephemeris.h . . . . .	2683
11.771 gps_iono.h File Reference . . . . .	2685
11.771.1 Detailed Description . . . . .	2686
11.772 gps_iono.h . . . . .	2686
11.773 GPS_L1_CA.h File Reference . . . . .	2687
11.773.1 Detailed Description . . . . .	2692
11.774 GPS_L1_CA.h . . . . .	2692
11.775 GPS_L2C.h File Reference . . . . .	2695
11.775.1 Detailed Description . . . . .	2696
11.776 GPS_L2C.h . . . . .	2697
11.777 GPS_L5.h File Reference . . . . .	2698
11.777.1 Detailed Description . . . . .	2699
11.778 GPS_L5.h . . . . .	2699
11.779 gps_navigation_message.h File Reference . . . . .	2701
11.779.1 Detailed Description . . . . .	2702
11.780 gps_navigation_message.h . . . . .	2703
11.781 gps_utc_model.h File Reference . . . . .	2706
11.781.1 Detailed Description . . . . .	2707
11.782 gps_utc_model.h . . . . .	2707
11.783 MATH_CONSTANTS.h File Reference . . . . .	2709
11.783.1 Detailed Description . . . . .	2712
11.784 MATH_CONSTANTS.h . . . . .	2712
11.785 osnma_data.h File Reference . . . . .	2714
11.785.1 Detailed Description . . . . .	2715
11.786 osnma_data.h . . . . .	2715
11.787 osnma_dsm_reader.h File Reference . . . . .	2718
11.787.1 Detailed Description . . . . .	2719

11.788 osnma_dsm_reader.h . . . . .	2719
11.789 reed_solomon.h File Reference . . . . .	2720
11.789.1 Detailed Description . . . . .	2721
11.790 reed_solomon.h . . . . .	2721
11.791 sbas_ephemeris.h File Reference . . . . .	2723
11.791.1 Detailed Description . . . . .	2724
11.792 sbas_ephemeris.h . . . . .	2724

<b>Index</b>	<b>2725</b>
--------------	-------------

# Chapter 1

## GNSS-SDR



Figure 1.1 GNSS-SDR logo

Welcome to GNSS-SDR!

GNSS-SDR is an open-source [GNSS software receiver](#) freely available to the research community. This project provides a common framework for GNSS signal processing which can operate in a variety of computer platforms. This tool is intended to foster collaboration, increase awareness, and reduce development costs in the field of GNSS receiver design and customized use of GNSS signals.

For details about GNSS-SDR and using it, please see the [main project page](#) or browse [the source code at GitHub](#). You could be also interested in [subscribing to the mailing list](#).

### 1.1 Contents

- [Overview](#)
- [Building GNSS-SDR](#)
- [Using GNSS-SDR](#)
- [Control plane](#)
- [Signal Processing plane](#)
- [About the software license](#)
- [Publications and Credits](#)
- [Ok, now what?](#)

More details on GNSS-SDR signal processing blocks:

- [Signal Source](#)
- [Signal Conditioner](#)

- Channel
  - Acquisition
  - Tracking
  - Decoding of the navigation message
- Observables
- Computation of Position, Velocity and Time

## 1.2 Overview

GNSS-SDR provides an interface to different suitable RF front-ends and implements all the receiver chain up to the navigation solution. Its design allows any kind of customization, including interchangeability of signal sources, signal processing algorithms, interoperability with other systems, output formats, and offers interfaces to all the intermediate signals, parameters and variables. The goal is to write efficient and truly reusable code, easy to read and maintain, with fewer bugs, and producing highly optimized executables in a variety of hardware platforms and operating systems. In that sense, the challenge consists of defining a gentle balance within level of abstraction and performance. GNSS-SDR runs in a personal computer and provides interfaces through USB and Ethernet buses to a variety of either commercially available or custom-made RF front-ends, adapting the processing algorithms to different sampling frequencies, intermediate frequencies and sample resolutions. This makes possible rapid prototyping of specific receivers intended, for instance, to geodetic applications, observation of the ionospheric impact on navigation signals, GNSS reflectometry, signal quality monitoring, or carrier-phase based navigation techniques.

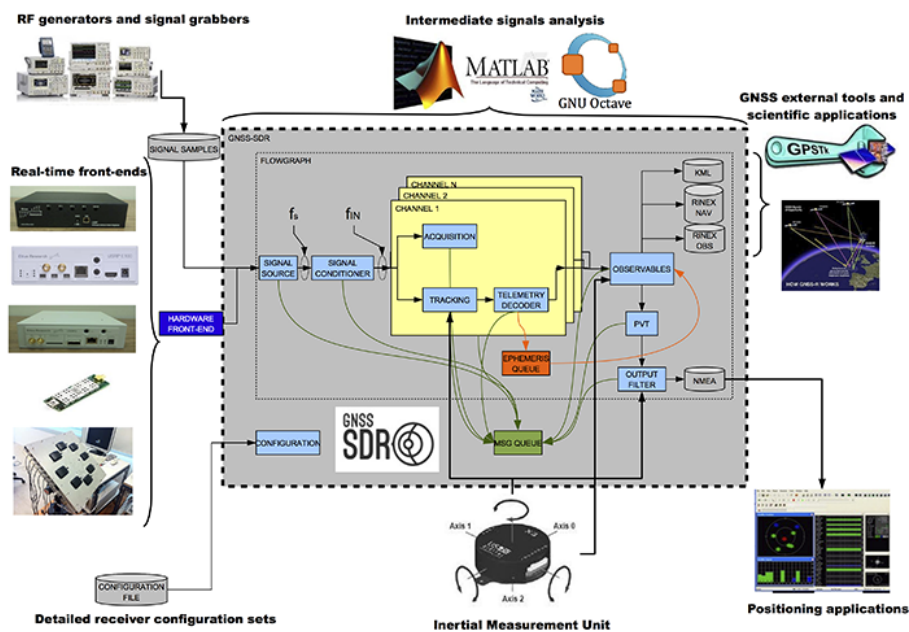


Figure 1.2 Overview

As signal inputs, it accepts:

- Raw data file captured with a data grabber (digitized at some intermediate frequency or directly at baseband).
- Any suitable RF configuration that can be driven by the Universal Software Radio Peripheral Hardware Driver ( [UHD](#) ). This includes all current and future [Ettus Research](#) products. The USRP1 + DBSRX 2.2 daughterboard is an example of working configuration for GPS L1 C/A and Galileo E1B and E1C signals.



- Experimentally, with some [USB DVB-T dongles based on the Realtek RTL2832U chipset](#).
- For mathematical representations of the targeted signals, check out the [Signal model](#) page.

As outputs, it provides:

- Dump of intermediate signals (configurable by the user)
- The processing is logged at a system temporary folder (usually, `/tmp`)
- Observables in form of RINEX file (experimental)
- Navigation message data in form of RINEX file
- Position, Velocity and Time solution in KML format and NMEA

## 1.3 Building GNSS-SDR

In principle, GNSS-SDR can be built in any Unix-like system. In practice, it depends on being able to install all the required dependencies. See the [building guide](#) page for details about the project's dependencies and build process. Mainly, it consists on installing [GNU Radio](#) plus some few more libraries:

- [Armadillo](#), a C++ linear algebra library,
- [Boost](#), a set of free peer-reviewed portable C++ source libraries,
- [Gflags](#), a library that implements commandline flags processing,
- [Glog](#), a library that implements application-level logging,
- [Googletest](#), Google's framework for writing C++ tests,
- [Mako](#), a template library written in Python,
- [Matio](#), a MATLAB MAT File I/O Library,
- [Protocol Buffers](#), a language-neutral, platform-neutral extensible mechanism for serializing structured data,
- [PugiXML](#), a light-weight, simple and fast XML parser for C++ with XPath support,
- [Volk](#), a Vector-Optimized Library of Kernels which provides an abstraction of optimized math routines targeting several SIMD processors,

and, optionally,

- GNU Radio modules for hardware interface ( [gr-uhd](#), [gr-osmosdr](#), [gr-iio](#) ),
- [Benchmark](#), a library to benchmark code snippets,
- [Gperftools](#), which provides fast, multi-threaded malloc() and performance analysis tools.

After all dependencies are installed, clone the GNSS-SDR repository:

```
$ git clone https://github.com/gnss-sdr/gnss-sdr
```

This will create a folder named `gnss-sdr` with the following structure:

```
|-gnss-sdr
|---cmake          <- CMake-related files
|---conf           <- Configuration files. Each file represents one receiver.
|---docs           <- Contains documentation-related files
|---install        <- Executables
|---src            <- Source code folder
|-----algorithms
|-----PVT
|-----acquisition
|-----channel
|-----conditioner
|-----data_type_adapter
|-----input_filter
|-----libs
|-----observables
|-----resampler
|-----signal_source
|-----telemetry_decoder
|-----tracking
|-----core
|-----interfaces
|-----libs
|-----receiver
|-----system_parameters
|-----main
|---tests
|---utils          <- some utilities (e.g. Matlab scripts)
```

You are now ready to build GNSS-SDR by using `CMake` as building tool:

```
$ cd gnss-sdr && mkdir build && cd build
$ cmake ..
$ make
```

If everything goes well, three new executables will be created at `gnss-sdr/install`, namely `gnss-sdr`, `volk_gnssdr_profile` and `run_tests`. You can run them from that folder, but if you prefer to install `gnss-sdr` on your system and have it available anywhere else, do:

```
$ sudo make install
```

This will make a copy of the `conf/` folder into `/usr/local/share/gnss-sdr/conf` for your reference. We suggest to create a working directory at your preferred location and store your own configuration and data files there.

You can create the documentation by doing:

```
$ make doc
```

from the building folder. In both cases, `Doxygen` will generate HTML documentation that can be retrieved pointing your browser of preference to `gnss-sdr/docs/html/index.html`.

There are two more extra targets available. In the building folder:

```
$ make doc-clean
```

will remove the content of previously-generated documentation and, if a LaTeX installation is detected in your system,

```
$ make pdfmanual
```

will create a PDF manual at `gnss-sdr/docs/GNSS-SDR_manual.pdf`. Please note that the PDF generation requires some fonts to be installed on the host system. In Ubuntu, those fonts do not come by default. You can install them by doing:

```
$ sudo apt install texlive-fonts-recommended
```

and then run `cmake ../` and `make pdfmanual` again.

### 1.3.1 Debug and Release builds

By default, CMake will build the Release version, meaning that the compiler will generate a faster, optimized executable. This is the recommended build type when using a RF front-end and you need to attain real time. If working with a file (and thus without real-time constraints), you may want to obtain more information about the internals of the receiver, as well as more fine-grained logging. This can be done by building the Debug version, by doing:

```
$ cd gnss-sdr && mkdir build-debug && cd build-debug
$ cmake -DCMAKE_BUILD_TYPE=Debug ..
$ make
```

### 1.3.2 Updating GNSS-SDR

If you checked out GNSS-SDR some days ago, it is possible that some developer had updated files at the Git repository. You can update your local copy by doing:

```
$ git checkout next
$ git pull https://github.com/gnss-sdr/gnss-sdr next
```

Before rebuilding the source code, it is safe (and recommended) to remove the remainders of old builds:

```
$ cd <building folder>
$ sudo make uninstall ; if you installed it before
$ rm -rf *
```

You can also check [The Git Book](#) for more information about Git usage.

## 1.4 Using GNSS-SDR

With GNSS-SDR, you can define your own receiver, work with captured raw data or from a RF front-end, dump into files intermediate signals, or tune every single algorithm used in the [Signal Processing plane](#). All the configuration is done in a single file. Those configuration files reside at the `gnss-sdr/conf` folder. By default, the executable `gnss-sdr` will read the configuration available at `gnss-sdr/conf/gnss-sdr.conf`. You can edit that file to fit your needs, or even better, define a new `my_receiver.conf` file with your own configuration. This new receiver can be done by invoking `gnss-sdr` with the `-config_file` flag pointing to your configuration file:

```
$ gnss-sdr --config_file=./conf/my_receiver.conf
```

You can see a guide of available implementations at [the online documentation](#). That folder contains other working examples as well. If you have a working configuration and want to share it with others, please email it to the [GNSS-SDR developers mailing list](#) and we will be happy to upload it to the server.

You can use a single configuration file for processing different data files, specifying the file to be processed with the `-signal_source` flag:

```
$ gnss-sdr --config_file=./conf/my_receiver.conf --signal_source=./my_captured_data.dat
```

This will override the `SignalSource.filename` specified in the configuration file.

You can get a complete list of available commandline flags by doing:

```
$ gnss-sdr --help
```

## 1.5 Control plane

GNSS-SDR's main method initializes the logging library, processes the command line flags, if any, provided by the user and instantiates a [ControlThread](#) object. Its constructor reads the configuration file, creates a control queue and creates a flowgraph according to the configuration. Then, the program's main method calls the `run()` method of the instantiated object, an action that connects the flowgraph and starts running it. After that, and until a stop message is received, it reads control messages sent by the receiver's modules through a safe-thread queue and processes them. Finally, when a stop message is received, the main method executes the destructor of the [ControlThread](#) object, which deallocates memory, does other cleanup and exits the program.

The [GNSSFlowgraph](#) class is responsible for preparing the graph of blocks according to the configuration, running it, modifying it during run-time and stopping it. Blocks are identified by its role. This class knows which roles it has to instantiate and how to connect them. It relies on the configuration to get the correct instances of the roles it needs and then it applies the connections between GNU Radio blocks to make the graph ready to be started. The complexity related to managing the blocks and the data stream is handled by GNU Radio's `gr::top_block` class. [GNSSFlowgraph](#) wraps the `gr::top_block` instance so we can take advantage of the [GNSS block factory](#), the configuration system and the processing blocks. This class is also responsible for applying changes to the configuration of the flowgraph during run-time, dynamically reconfiguring channels: it selects the strategy for selecting satellites. This can range from a sequential search over all the satellites' ID to smarter approaches that determine what are the satellites most likely in-view based on rough estimations of the receiver position in order to avoid searching satellites in the other side of the Earth.

The Control Plane is in charge of creating a flowgraph according to the configuration and then managing the modules. Configuration allows users to define in an easy way their own custom receiver by specifying the flowgraph (type of signal source, number of channels, algorithms to be used for each channel and each module, strategies for satellite selection, type of output format, etc.). Since it is difficult to foresee what future module implementations will be needed in terms of configuration, we used a very simple approach that can be extended without a major impact in the code. This can be achieved by simply mapping the names of the variables in the modules with the names of the parameters in the configuration.

### 1.5.1 Configuration

Properties are passed around within the program using the [ConfigurationInterface](#) class. There are two implementations of this interface: [FileConfiguration](#) and [InMemoryConfiguration](#). [FileConfiguration](#) reads the properties (pairs of property name and value) from a file and stores them internally. [InMemoryConfiguration](#) does not read from a file; it remains empty after instantiation and property values and names are set using the `set` property method. [FileConfiguration](#) is intended to be used in the actual GNSS-SDR application whereas [InMemoryConfiguration](#) is intended to be used in tests to avoid file-dependency in the file system. Classes that need to read configuration parameters will receive instances of [ConfigurationInterface](#) from where they will fetch the values. For instance, parameters related to `SignalSource` should look like this:

```
SignalSource.parameter1=value1
SignalSource.parameter2=value2
```

The name of these parameters can be anything but one reserved word: `implementation`. This parameter indicates in its value the name of the class that has to be instantiated by the factory for that role. For instance, if our signal source is providing data already at baseband and thus we want to use the implementation [Pass\\_Through](#) for module [SignalConditioner](#), the corresponding line in the configuration file would be

```
SignalConditioner.implementation=Pass_Through
```

Since the configuration is just a set of property names and values without any meaning or syntax, the system is very versatile and easily extendable. Adding new properties to the system only implies modifications in the classes that will make use of these properties. In addition, the configuration files are not checked against any strict syntax so it is always in a correct status (as long as it contains pairs of property names and values in [INI format](#)).

### 1.5.2 GNSS block factory

Hence, the application defines a simple accessor class to fetch the configuration pairs of values and passes them to a factory class called [GNSSBlockFactory](#). This factory decides, according to the configuration, which class needs to be instantiated and which parameters should be passed to the constructor. Hence, the factory encapsulates the complexity of blocks' instantiation. With that approach, adding a new block that requires new parameters will be as simple as adding the block class and modifying the factory to be able to instantiate it. This loose coupling between the blocks' implementations and the syntax of the configuration enables extending the application capacities in a high degree. It also allows to produce fully customized receivers, for instance a testbed for acquisition algorithms, and to place observers at any point of the receiver chain.

## 1.6 Signal Processing plane

GNU Radio's class `gr::basic_block` is the abstract base class for all signal processing blocks, a bare abstraction of an entity that has a name and a set of inputs and outputs. It is never instantiated directly; rather, this is the abstract parent class of both `gr::hier_block2`, which is a recursive container that adds or removes processing or hierarchical blocks to the internal graph, and `gr::block`, which is the abstract base class for all the processing blocks.

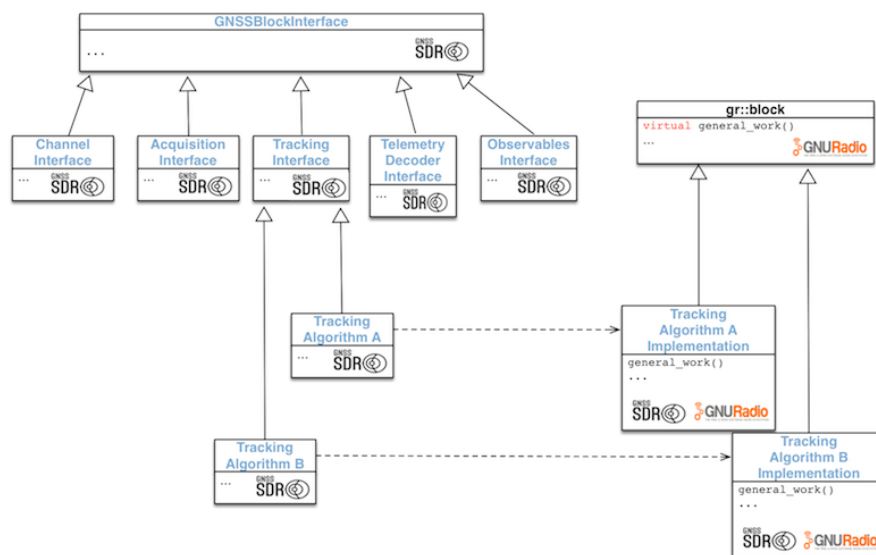


Figure 1.3 Class hierarchy of signal processing blocks

A signal processing flow is constructed by creating a tree of hierarchical blocks, which at any level may also contain terminal nodes that actually implement signal processing functions.

Class `gr::top_block` is the top-level hierarchical block representing a flowgraph. It defines GNU Radio run-time functions used during the execution of the program: `run()`, `start()`, `stop()`, `wait()`, etc. A subclass called [GNSSBlockInterface](#) is the common interface for all the GNSS-SDR modules. It defines pure virtual methods, that are required to be implemented by a derived class.

Subclassing [GNSSBlockInterface](#), we defined interfaces for the GNSS receiver blocks depicted in the figure above. This hierarchy provides the definition of different algorithms and different implementations, which will be instantiated according to the configuration. This strategy allows multiple implementations sharing a common interface, achieving the objective of decoupling interfaces from implementations: it defines a family of algorithms, encapsulates each one, and makes them interchangeable. Hence, we let the algorithm vary independently from the program that uses it.

### 1.6.1 Signal Source

The input of a software receiver are the raw bits that come out from the front-end's analog-to-digital converter (ADC). Those bits can be read from a file stored in the hard disk or directly in real-time from a hardware device through USB or Ethernet buses.

The Signal Source module is in charge of implementing the hardware driver, that is, the portion of the code that communicates with the RF front-end and receives the samples coming from the ADC. This communication is usually performed through USB or Ethernet buses. Since real-time processing requires a highly optimized implementation of the whole receiver, this module also allows to read samples from a file stored in a hard disk, and thus processing without time constraints. Relevant parameters of those samples are the intermediate frequency (or baseband I&Q components), the sampling rate and number of bits per sample, that must be specified by the user in the configuration file.

This module also performs bit-depth adaptation, since most of the existing RF front-ends provide samples quantized with 2 or 3 bits, while operations inside the processor are performed on 32- or 64-bit words, depending on its architecture. Although there are implementations of the most intensive computational processes (mainly correlation) that take advantage of specific data types and architectures for the sake of efficiency, the approach is processor-specific and hardly portable. We suggest to keep signal samples in standard data types and letting the compiler select the best library version (implemented using SIMD or any other processor-specific technology) of the required routines for a given processor.

Example: [FileSignalSource](#)

The user can configure the receiver for reading from a file, setting in the configuration file the data file location, sample format, and the sampling frequency and intermediate frequency at what the signal was originally captured.

```
##### SIGNAL_SOURCE CONFIG #####
SignalSource.implementation=File_Signal_Source
SignalSource.filename=/home/user/gnss-sdr/data/my_capture.dat
SignalSource.item_type=gr_complex
SignalSource.sampling_frequency=4000000 ; Sampling frequency in [Hz]
SignalSource.freq=1575420000 ; RF front-end center frequency in [Hz]
```

Example: [UhdSignalSource](#)

The user may prefer to use a UHD-compatible RF front-end and try real-time processing. For instance, for a USRP1 + DBSRX daughterboard, use:

```
##### SIGNAL_SOURCE CONFIG #####
SignalSource.implementation=UHD_Signal_Source
SignalSource.item_type=gr_complex
SignalSource.sampling_frequency=4000000 ; Sampling frequency in [Hz]
SignalSource.freq=1575420000 ; RF front-end center frequency in [Hz]
SignalSource.gain=60 ; Front-end gain in dB
SignalSource.subdevice=B:0 ; UHD subdevice specification (for USRP1 use A:0 or B:0)
```

Other examples are available at `gnss-sdr/conf`.

### 1.6.2 Signal Conditioner

The signal conditioner is in charge of resampling the signal and delivering a reference sample rate to the downstream processing blocks, acting as a facade between the signal source and the synchronization channels, providing a simplified interface to the input signal. In case of multiband front-ends, this module would be in charge of providing a separated data stream for each band.

### 1.6.3 Channel

A channel encapsulates all signal processing devoted to a single satellite. Thus, it is a large composite object which encapsulates the [Acquisition](#), [Tracking](#) and [Decoding of the navigation message](#) modules. As a composite object, it can be treated as a single entity, meaning that it can be easily replicated. Since the number of channels is selectable by the user in the configuration file, this approach helps improving the scalability and maintainability of the receiver.

This module is also in charge of managing the interplay between acquisition and tracking. Acquisition can be initialized in several ways, depending on the prior information available (called cold start when the receiver has no information about its position nor the satellites almanac; warm start when a rough location and the approximate time of day are available, and the receiver has a recently recorded almanac broadcast; or hot start when the receiver was tracking a satellite and the signal line of sight broke for a short period of time, but the ephemeris and almanac data is still valid, or this information is provided by other means), and an acquisition process can finish deciding that the satellite is not present, that longer integration is needed in order to confirm the presence of the satellite, or declaring the satellite present. In the latter case, acquisition process should stop and trigger the tracking module with coarse estimations of the synchronization parameters.

The abstract class [ChannelInterface](#) represents an interface to a channel GNSS block. Check [Channel](#) for an actual implementation.

#### 1.6.3.1 Acquisition

The first task of a GNSS receiver is to detect the presence or absence of in-view satellites. This is done by the acquisition system process, which also provides a coarse estimation of two signal parameters: the frequency shift with respect to the nominal IF frequency, and a delay term which allows the receiver to create a local code aligned with the incoming code. [AcquisitionInterface](#) is the common interface for all the acquisition algorithms and their corresponding implementations. Algorithms' interface, that may vary depending on the use of information external to the receiver, such as in Assisted GNSS, is defined in classes referred to as *adapters*. These adapters wrap the GNU Radio blocks interface into a compatible interface expected by [AcquisitionInterface](#). This allows the use of existing GNU Radio blocks derived from `gr::block`, and ensures that newly developed implementations will also be reusable in other GNU Radio-based applications. Moreover, it adds still another layer of abstraction, since each given acquisition algorithm can have different implementations (for instance using different numerical libraries). In such a way, implementations can be continuously improved without having any impact neither on the algorithm interface nor the general acquisition interface.

Check [GpsL1CaPcpsAcquisition](#) and [GalileoE1PcpsAmbiguousAcquisition](#) for examples of adapters from a Parallel Code Phase Search (PCPS) acquisition block, and `pcps_acquisition_cc` for an example of a block implementation. The source code of all the available acquisition algorithms is located at:

```
|-gnss-sdr
|---src
|----algorithms
|-----acquisition
|-----adapters          <- Adapters of the processing blocks to an AcquisitionInterface
|-----gnuradio_blocks   <- Signal processing blocks implementation
```

The user can select a given implementation for the algorithm to be used in each receiver channel, as well as their parameters, in the configuration file:

```
##### ACQUISITION GLOBAL CONFIG #####

;#implementation: Acquisition algorithm selection for this channel:
Acquisition_1C.implementation=GPS_L1_CA_PCPS_Acquisition
;#dump: Enable or disable the acquisition internal data file logging [true] or [false]
Acquisition_1C.dump=false
;#filename: Log path and filename
Acquisition_1C.dump_filename=./acq_dump.dat
```

```
;#item_type: Type and resolution for each of the signal samples. Use only gr_complex in this version.
Acquisition_1C.item_type=gr_complex
;#coherent_integration_time_ms: Signal block duration for the acquisition signal detection [ms]
Acquisition_1C.coherent_integration_time_ms=1
;#threshold: Acquisition threshold
Acquisition_1C.threshold=2.5
;#pfa: Acquisition false alarm probability. This option overrides the threshold option.
Acquisition_1C.pfa=0.0001
;#doppler_max: Maximum expected Doppler shift [Hz]
Acquisition_1C.doppler_max=5000
;#doppler_step: Doppler step in the grid search [Hz]
Acquisition_1C.doppler_step=250
```

### 1.6.3.2 Tracking

When a satellite is declared present, the parameters estimated by the acquisition module are then fed to the receiver tracking module, which represents the second stage of the signal processing unit, aiming to perform a local search for accurate estimates of code delay and carrier phase, and following their eventual variations.

Again, a class hierarchy consisting of a [TrackingInterface](#) class and subclasses implementing algorithms provides a way of testing different approaches, with full access to their parameters. Check [GpsL1CaDlPllTracking](#) or [GalileoE1DlPllVemlTracking](#) for examples of adapters, and [Gps\\_L1\\_Ca\\_Dll\\_Pll\\_Tracking\\_cc](#) for an example of a signal processing block implementation. There are also available some useful classes and functions for signal tracking; take a look at Correlator, [lock\\_detectors.h](#), [tracking\\_discriminators.h](#) or [tracking\\_2nd\\_DLL\\_filter.h](#).

The source code of all the available tracking algorithms is located at:

```
| -gnss-sdr
| ---src
| ----algorithms
| -----tracking
| -----adapters          <- Adapters of the processing blocks to a TrackingInterface
| -----gnuradio_blocks    <- Signal processing blocks implementation
| -----libs               <- libraries of tracking objects (e.g. correlators, discriminators, and so on)
```

The user can select a given implementation for the algorithm to be used in all the tracking blocks, as well as its parameters, in the configuration file:

```
##### TRACKING GLOBAL CONFIG #####

;#implementation: Selected tracking algorithm
Tracking_1C.implementation=GPS_L1_CA_DLL_PLL_Tracking
;#item_type: Type and resolution for each of the signal samples.
Tracking_1C.item_type=gr_complex

;#dump: Enable or disable the Tracking internal binary data file logging [true] or [false]
Tracking_1C.dump=false

;#dump_filename: Log path and filename. Notice that the tracking channel will add "x.dat" where x is the channel
Tracking_1C.dump_filename=./tracking_ch_

;#pll_bw_hz: PLL loop filter bandwidth [Hz]
Tracking_1C.pll_bw_hz=50.0;

;#dll_bw_hz: DLL loop filter bandwidth [Hz]
Tracking_1C.dll_bw_hz=2.0;

;#fll_bw_hz: FLL loop filter bandwidth [Hz]
Tracking_1C.fll_bw_hz=10.0;

Tracking_1C.pll_filter_order=3 ; PLL loop filter order [2] or [3]
Tracking_1C.dll_filter_order=2 ; DLL loop filter order [1], [2] or [3]

;#early_late_space_chips: correlator early-late space [chips].
Tracking_1C.early_late_space_chips=0.5;
```



### 1.6.3.3 Decoding of the navigation message

Most of GNSS signal links are modulated by a navigation message containing the time the message was transmitted, orbital parameters of satellites (also known as ephemeris) and an almanac (information about the general system health, rough orbits of all satellites in the network as well as data related to error correction). Navigation data bits are structured in words, pages, subframes, frames and superframes. Sometimes, bits corresponding to a single parameter are spread over different words, and values extracted from different frames are required for proper decoding. Some words are for synchronization purposes, others for error control and others contain actual information. There are also error control mechanisms, from parity checks to forward error correction (FEC) encoding and interleaving, depending on the system.

The common interface is [TelemetryDecoderInterface](#). Check [GpsL1CaTelemetryDecoder](#) for an example of the GPS L1 NAV message decoding adapter, and `gps_l1_ca_telemetry_decoder_cc` for an actual implementation of a signal processing block. Configuration example:

```
;##### TELEMETRY DECODER CONFIG #####
TelemetryDecoder_1C.implementation=Gps_L1_CA_Telemetry_Decoder
TelemetryDecoder_1C.dump=false
```

See the [Reference Documents](#) for more information about the signal format.

### 1.6.4 Observables

GNSS systems provide different kinds of observations. The most commonly used are the code observations, also called pseudoranges. The *pseudo* comes from the fact that on the receiver side the clock error is unknown and thus the measurement is not a pure range observation. High accuracy applications also use the carrier phase observations, which are based on measuring the difference between the carrier phase transmitted by the GNSS satellites and the phase of the carrier generated in the receiver. Both observables are computed from the outputs of the tracking module and the decoding of the navigation message. This module collects all the data provided by every tracked channel, aligns all received data into a coherent set, and computes the observables.

The common interface is [ObservablesInterface](#).

Configuration example:

```
;##### OBSERVABLES CONFIG #####
Observables.implementation=Hybrid_Observables

;#dump: Enable or disable the Observables internal binary data file logging [true] or [false]
Observables.dump=false

;#dump_filename: Log path and filename.
Observables.dump_filename=./observables.dat
```

### 1.6.5 Computation of Position, Velocity and Time

Although data processing for obtaining high-accuracy PVT solutions is out of the scope of GNSS-SDR, we provide a module that can compute a simple least square solution and leaves room for more sophisticated positioning methods. The integration with libraries and software tools that are able to deal with multi-constellation data such as [GNSSTk](#) or [gLAB](#) appears as a viable solution for high performance, completely customizable GNSS receivers.

The common interface is [PvtInterface](#). For instance, in order to use the implementation `RTKLIB_PVT`, add to the configuration file:

```

;##### PVT CONFIG #####
PVT.implementation=RTKLIB_PVT

;#nmea_dump_filename: NMEA log path and filename
PVT.nmea_dump_filename=./gnss_sdr_pvt.nmea;

;#flag_nmea_tty_port: Enable or disable the NMEA log to a serial TTY port (Can be used with real hardware or v
PVT.flag_nmea_tty_port=true;

;#nmea_dump_devname: serial device descriptor for NMEA logging
PVT.nmea_dump_devname=/dev/pts/4

;#dump: Enable or disable the PVT internal binary data file logging [true] or [false]
PVT.dump=false

```

This implementation allows tuning of the following parameters:

```

PVT.implementation=RTKLIB_PVT
PVT.positioning_mode=Single      ; options: Single, Static, Kinematic, PPP_Static, PPP_Kinematic
PVT.iono_model=Broadcast        ; options: OFF, Broadcast
PVT.trop_model=Saastamoinen     ; options: OFF, Saastamoinen
PVT.rinex_version=2             ; options: 2 or 3
PVT.output_rate_ms=100         ; Period in [ms] between two PVT outputs
PVT.display_rate_ms=500        ; Position console print (std::out) interval [ms].
PVT.nmea_dump_filename=./gnss_sdr_pvt.nmea ; NMEA log path and filename
PVT.flag_nmea_tty_port=false    ; Enables the NMEA log to a serial TTY port
PVT.nmea_dump_devname=/dev/pts/4 ; serial device descriptor for NMEA logging
PVT.flag_rtcn_server=true       ; Enables or disables a TCP/IP server dispatching RTCM messages
PVT.flag_rtcn_tty_port=false    ; Enables the RTCM log to a serial TTY port
PVT.rtcn_dump_devname=/dev/pts/1 ; serial device descriptor for RTCM logging
PVT.rtcn_tcp_port=2101
PVT.rtcn_MT1019_rate_ms=5000
PVT.rtcn_MT1045_rate_ms=5000
PVT.rtcn_MT1097_rate_ms=1000
PVT.rtcn_MT1077_rate_ms=1000

```

## 1.7 About the software license

GNSS-SDR is released under the [General Public License \(GPL\) v3](#), thus securing practical usability, inspection, and continuous improvement by the research community, allowing the discussion based on tangible code and the analysis of results obtained with real signals. The GPL implies that:

- Copies may be distributed free of charge or for money, but the source code has to be shipped or provided free of charge (or at cost price) on demand. The receiver of the source code has the same rights meaning he can share copies free of charge or resell.
- The licensed material may be analyzed or modified.
- Modified material may be distributed under the same licensing terms but **do not** have to be distributed.

That means that modifications only have to be made available to the public if distribution happens. So it is perfectly fine to take the GNSS-SDR source code, modify it heavily and use it in a not distributed application / library. This is how companies like Google can run their own patched versions of Linux for example.

But what this also means is that non-GPL code cannot use GPL code. This means that you cannot modify / use GNSS-SDR, blend it with non-GPL code, and make money with the resulting software. You cannot distribute the resulting software under a non-disclosure agreement or contract. Distributors under the GPL also grant a license for any of their patents practiced by the software, to practice those patents in GPL software. You can sell a device that runs with GNSS-SDR, but if you distribute the code, it has to remain under GPL.

## 1.8 Publications and Credits

If you use GNSS-SDR to produce a research paper or Thesis, we would appreciate if you reference any of these articles to credit the GNSS-SDR project:

- C. Fernández-Prades, J. Arribas, L. Esteve, D. Pubill, P. Closas, *An Open Source Galileo E1 Software Receiver*, in Proc. of the 6th ESA Workshop on Satellite Navigation Technologies (NAVITEC 2012), ESTEC, Noordwijk, The Netherlands, Dec. 2012.
- J. Arribas, *GNSS Array-based Acquisition: Theory and Implementation*, PhD Thesis, Universitat Politècnica de Catalunya, Barcelona, Spain, June 2012.
- C. Fernández-Prades, J. Arribas, P. Closas, C. Avilés, and L. Esteve, *GNSS-SDR: an open source tool for researchers and developers*, in Proc. of the ION GNSS 2011 Conference, Portland, Oregon, Sept. 19-23, 2011.
- C. Fernández-Prades, C. Avilés, L. Esteve, J. Arribas, and P. Closas, *Design patterns for GNSS software receivers*, in Proc. of the 5th ESA Workshop on Satellite Navigation Technologies (NAVITEC'2010), ESTEC, Noordwijk, The Netherlands, Dec. 2010. DOI:10.1109/NAVITEC.2010.5707981

For LaTeX users, these are the BibTeX cites for your convenience:

```
@INPROCEEDINGS{GNSS-SDR12
  author = {C.~{Fern}\{a\}ndez--Prades} and J.~Arribas and L.~Esteve and D.~Pubill and P.~Closas},
  title = {An Open Source {G}alileo {E1} Software Receiver},
  booktitle = {Proc. of the 6th ESA Workshop on Satellite Navigation Technologies (NAVITEC'2012)},
  year = {2012},
  address = {ESTEC, Noordwijk, The Netherlands},
  month = {Dec.} }

@PHDTHESIS{Arribas12,
  author = {J.~Arribas},
  title = {{GNSS} Array-based Acquisition: Theory and Implementation},
  school = {Universitat Polit\`{e}cnica de Catalunya},
  year = {2012},
  address = {Barcelona, Spain},
  month = {June} }

@INPROCEEDINGS{GNSS-SDR11,
  AUTHOR = {C.~{Fern}\{a\}ndez--Prades} and J.~Arribas and P.~Closas and C.~Avil\{e\}s and L.~Esteve},
  TITLE = {{GNSS-SDR}: An Open Source Tool For Researchers and Developers},
  BOOKTITLE = {Proc. of the ION GNSS 2011 Conference},
  YEAR = {2011},
  address = {Portland, Oregon},
  month = {Sept.} }

@INPROCEEDINGS{GNSS-SDR10,
  AUTHOR = {C.~{Fern}\{a\}ndez--Prades} and C.~Avil\{e\}s and L.~Esteve and J.~Arribas and P.~Closas},
  TITLE = {Design patterns for {GNSS} software receivers},
  BOOKTITLE = {Proc. of the 5th ESA Workshop on Satellite Navigation Technologies (NAVITEC'2010)},
  YEAR = {2010},
  address = {ESTEC, Noordwijk, The Netherlands},
  month = {Dec.},
  note = {doi:10.1109/NAVITEC.2010.5707981} }
```

More papers related to GNSS-SDR are available at the [publications page](#).

## 1.9 Ok, now what?

In order to start using GNSS-SDR, you may want to populate `gnss-sdr/data` folder (or anywhere else on your system) with raw data files. By "raw data" we mean the output of a Radio Frequency front-end's Analog-to-Digital converter. GNSS-SDR needs signal samples already in baseband or in passband, at a suitable intermediate frequency (on the order of MHz). Prepare your configuration file, and then you are ready for going to the `gnss-sdr/install` folder, running `./gnss-sdr`, and see how the file is processed. Please ask the Developer Team for a signal sample if you need one, and they will do their best ;-)

Another interesting option is working in real-time with a RF front-end. We provide drivers for UHD-compatible hardware (see [Signal Source](#)) and for some DVB-T USB dongles. Start with a low number of channels and then increase it in order to test how many channels your processor can handle in real-time.

You can find more information at the [GNSS-SDR Documentation page](#) or directly asking to the [GNSS-SDR Developers mailing list](#).

You are also very welcome to contribute to the project, there are many ways to [participate in GNSS-SDR](#). If you need some special feature not yet implemented, the Developer Team would love to be hired for developing it. Please do not hesitate to [contact them](#).

Enjoy GNSS-SDR!

The Developer Team.

## Chapter 2

# Reference Documents

### 2.1 Interface Control Documents

#### 2.1.1 GPS

All the current GPS Interface Control Documents can be downloaded from [GPS.gov](https://www.gps.gov), the official U.S. Government webpage for GPS.

- GPS L1 and L2C: Global Positioning System Directorate, [Interface Specification IS-GPS-200 Revision M](#). May, 2021.
- GPS L1C (available with first Block III launch): Global Positioning System Directorate, [Interface Specification IS-GPS-800 Revision H](#). May, 2021.
- GPS L5 (first Block IIF satellite launched on May, 2010): Global Positioning System Directorate, [Interface Specification IS-GPS-705 Revision H](#). May, 2021.

#### 2.1.2 GLONASS

Official GLONASS webpage: [Information-analytical centre official website](#).

- Standard Accuracy (ST) signals at L1 and L2: Russian Institute of Space Device Engineering, Global Navigation Satellite System GLONASS. [Interface Control Document. Navigational radiosignal in bands L1, L2. Edition 5.1](#), Moscow, Russia, 2008
- [GLONASS Interface Control Document. Open CDMA navigational radio signal in L1 band. Edition 1.0 \(in Russian\)](#). Russian Space Systems OJSC. 2016.
- [GLONASS Interface Control Document. Open CDMA navigational radio signal in L2 band. Edition 1.0 \(in Russian\)](#). Russian Space Systems OJSC. 2016.
- [GLONASS Interface Control Document. Open CDMA navigational radio signal in L3 band. Edition 1.0 \(in Russian\)](#). Russian Space Systems OJSC. 2016.

### 2.1.3 Galileo

Check the [Galileo website of the European Global Navigation Satellite Systems Agency \(GSA\)](#) and the [Galileo website of the European Space Agency](#). There is a website with [Galileo constellation status information](#) from the GSA.

- Galileo E5, E6, and E1: European GNSS (Galileo) Open Service. [Signal In Space Interface Control Document. Ref: OS SIS ICD, Issue 2.0](#), European Union, Jan. 2021.

The European Commission is granting free access to the technical information on the future Galileo open service signal, i.e. the specifications manufacturers and developers need to process data received from satellites. This document informs receiver manufacturers, application developers and service providers on how to use the future Galileo system and what they can expect in terms of performance.

### 2.1.4 BeiDou

Official webpage at [beidou.gov.cn](http://beidou.gov.cn)

- [BeiDou Navigation Satellite System Signal In Space Interface Control Document Open Service Signal B1I \(Version 3.0\)](#). China Satellite Navigation Office, Feb. 2019.
- [BeiDou Navigation Satellite System Signal In Space Interface Control Document Open Service Signal B1C \(Version 1.0\)](#). China Satellite Navigation Office, Jun. 2018.
- [BeiDou Navigation Satellite System Signal In Space Interface Control Document Open Service Signal B3I \(Version 1.0\)](#). China Satellite Navigation Office, Feb. 2018.
- [BeiDou Navigation Satellite System Signal In Space Interface Control Document Open Service Signal B2a \(Version 1.0\)](#). China Satellite Navigation Office, Dec. 2017.
- [BeiDou Navigation Satellite System Signal In Space Interface Control Document Open Service Signal \(Version 2.1\)](#). China Satellite Navigation Office, December 2016.

### 2.1.5 Satellite Based Augmentation Systems (SBAS)

- **Minimum Operational Performance Standards for Global Positioning System/Wide Area Augmentation System Airborne Equipment, DO-229D**, RTCA, Washington, DC, Dec. 13, 2006. The 'RTCA MOPS DO229D - appendix A' is the reference standard for WAAS/EGNOS application development. RTCA is an advisory committee of the US federal government, and issues standards for civil airborne equipment, among other duties. One such standard is MOPS 229D (Minimum Operational Performance Standards for Global Positioning System/Wide Area Augmentation System Airborne Equipment version D), which describes the implementation of satellite-based augmentation services (SBAS) for receivers designed for civil aviation use. An annex to DO229D contains the specifications for the SBAS signal and message. The RTCA provides regular updates to these standards. MOPS 229D is available for a fee from the [RTCA website](#).
- **Global Positioning System Wide Area Augmentation System (WAAS) Performance Standard, 1st Edition**, Department of Transportation and Federal Aviation Administration, Oct. 31, 2008. This document defines the levels of performance the U.S. Government makes available to users of the GPS SPS augmented by the Wide Area Augmentation System.
- **EGNOS Data Access Service (EDAS) Service Definition Document. Revision 2.2**, European GNSS Agency (GSA), June, 2019. This is a complementary document to the RTCA DO229D, mentioned above. It describes the scope of services provided by the EGNOS EDAS Service to be used by end-users or Application Specific Service Providers. It details the general conditions relating to the use of the EGNOS service, a technical description of the Signal-in-Space (SIS), the reference receiver, environmental conditions, the service performance achieved and aspects relating to service provision.
- **EGNOS Safety of Life Service Definition Document. Revision 3.3**, European GNSS Agency (GSA), Mar, 2019. The EGNOS Safety of Life (SoL) Service is provided openly and is freely accessible without any direct charge and is tailored to safety-critical transport applications in various domains, in particular for aviation applications. The service is thus compliant with the aviation APV-I (Approach with Vertical Guidance) requirements, as defined by ICAO in Annex 10, but may support also applications in other SoL domains.
- **EGNOS Open Service Service Definition Document. Revision 2.3**, European GNSS Agency (GSA), Sep., 2017.

More information about EGNOS can be found through the [EGNOS Portal](#).

## 2.2 Other Standards

### 2.2.1 RINEX

The final output of a navigation receiver is usually its position, speed or other related physical quantities. However, the calculation of those quantities are based on a series of measurements from one or more satellite constellations. Although receivers calculate positions in real time, in many cases it is interesting to store intermediate measures for later post-processing. RINEX is the standard format that allows the management and disposal of the measures generated by a receiver, as well as their off-line processing by a multitude of applications.

- The most common version at present is **RINEX: The Receiver Independent Exchange Format Version 2.12**, which enables storage of measurements from pseudorange, carrier-phase and Doppler systems for GPS, GLONASS, Galileo along with data from EGNOS and WAAS satellite based augmentation systems (SBAS).
- The most recent version is **RINEX: The Receiver Independent Exchange Format Version 3.03** published in July, 2015. It includes Galileo and improves the handling of multi-constellation data files.
- There is also available the **RINEX Extensions to Handle Clock Information**, published in September, 2010.

### 2.2.2 NMEA

The [National Marine Electronics Association](#) released the NMEA 0183 Interface Standard, which defines electrical signal requirements, data transmission protocol and time, and specific sentence formats for a 4800-baud serial data bus. The standard is [available for purchase](#).

### 2.2.3 KML

KML is an XML language focused on geographic visualization, including annotation of maps and images. Geographic visualization includes not only the presentation of graphical data on the globe, but also the control of the user's navigation in the sense of where to go and where to look. Google submitted KML (formerly Keyhole Markup Language) to the Open Geospatial Consortium (OGC) to be evolved within the OGC consensus process with the following goal: KML Version 2.2 has been adopted as an OGC implementation standard.

- Open Geospatial Consortium, Inc., [OGC KML Version 2.2.0](#), April 2008.

### 2.2.4 C++ Standards

The C++ programming language is standardized by the International Organization for Standardization (ISO), with the latest standard version ratified and published by ISO in December 2017 as ISO/IEC 14882:2017 (informally known as C++17). The list of supported C++ standards (the highest available is automatically selected by the CMake script):

- **Draft C++23:** Check the [C++ standard draft sources at GitHub](#).
- **C++20:** The current ISO C++ standard is officially known as *ISO International Standard ISO/IEC 14882:2020(E) – Programming languages – C++*. You can get it from [ISO](#), [IEC](#) or [ANSI](#). The closest free working document available is [N4868](#).
- **C++17:** A former ISO C++ standard was officially known as *ISO International Standard ISO/IEC 14882:2017 – Programming languages – C++*. You can get it from [ISO](#), [IEC](#) or [ANSI](#). The closest free working document available is [N4659](#).
- **C++14:** A former ISO C++ standard was officially known as *ISO International Standard ISO/IEC 14882:2014 – Programming languages – C++*. You can get it from [ISO](#) or [ANSI](#). The closest free working document available is [N4296](#).
- **C++11:** An older ISO C++ standard was ISO/IEC 14882:2011. You can get it from [ISO](#). The closest free working document available is [N3337](#).

### 2.2.5 Positioning protocols in wireless communication networks

Cellular industry location standards first appeared in the late 1990s, with the [3rd generation partnership project \(3GPP\)](#) radio resource location services protocol (RRLP) technical specification 44.031 positioning protocol for GSM networks. Today, RRLP is the de facto standardized protocol to carry GNSS assistance data to GNSS-enabled mobile devices, and the term "3GPP specification" now covers all GSM (including GPRS and EDGE), W-CDMA and LTE (including LTE-A) specifications. Precisely, the label "LTE-A" is applied to networks compliant with LTE Release 10 and beyond, which fulfill the requirements issued by the [International Telecommunication Union Radiocommunication Sector \(ITU-R\)](https://www.itu.int/en/ITU-R/Pages/default.aspx) in the global standard for international mobile telecommunications (IMT Advanced, also referred to as 4G) access technologies. Control plane protocols: \li Radio Resource LCS Protocol (RRLP): <https://www.3gpp.org/ftp/Specs/html-info/44031.htm>



target="\_blank"><b>3GPP Technical Specification 44.031</b></a>. \li LTE Positioning Protocol (LPP): <a href="https://www.3gpp.org/ftp/Specs/html-info/36355.htm" target="\_blank"><b>3GPP Technical Specification 36.355</b></a>. User plane protocols: \li Open Mobile Alliance (OMA), <a href="http://member.openmobilealliance.org/ftp/Public\_documents/LOC/Permanent\_documents/OMA-AD-SUPL-V1\_0-20070615-A.zip" target="\_blank"><b>Secure User Plane Location Architecture Version 1 (SUPL 1.0)</b></a>, June 2007. \li Open Mobile Alliance (OMA), <a href="http://member.openmobilealliance.org/ftp/Public\_documents/LOC/Permanent\_documents/OMA-AD-SUPL-V2\_0-20120417-A.zip" target="\_blank"><b>Secure User Plane Location Architecture Version 2 (SUPL 2.0)</b></a>, April 2012. LTE Release 9 introduced extension hooks in LPP messages, so that the bodies external to 3GPP could extend the LPP feature set. OMA LPP extensions (↔ LPPe), supported in SUPL 3.0, build on top of the 3GPP LPP reusing its procedures and data types. Check the <a href="https://technical.openmobilealliance.org/index.html" target="\_blank">OMA Specifications webpage</a> for updated information about LPP Extensions (LPPe) Specification. \li The <a href="http://member.openmobilealliance.org/ftp/Public\_documents/loc/Permanent\_documents/OMA-TS-MLP-V3\_5-20181119-D.zip" target="\_blank"><b>OMA Mobile Location Protocol (MLP) V3.5</b> is an application-level protocol for getting the position of mobile stations (mobile phones, wireless personal digital assistants, etc.) independent of underlying network technology. The MLP serves as the interface between a Location Server and a Location Services (LCS) Client. This specification defines the core set of operations that a Location Server should be able to perform.



## Chapter 3

# Signal model

### 3.1 GNSS signal model

This page describes signals transmitted by GNSS space vehicles. Signal models are mathematical representations of the electromagnetic waves that are exciting the receiver's antenna. The current induced by those waves is then amplified, filtered and downconverted to a suitable frequency (can be at some intermediate frequency or directly to baseband), and then converted to 0s and 1s by the Analog-to-Digital Converter (ADC). That is the job of the Radio Frequency front-end, which at its output delivers a stream of digital samples. Those samples constitute the input of a software receiver, so for GNSS-SDR the signal models described below can be seen as *the rules of the game*.

GNSS' space vehicles are modern versions of lighthouses, but with better visibility. Each satellite is a reference point, and if we know our distance to several reference points, we can compute our location, just as mariners do when they see a couple of lighthouses. For each in-view satellite  $i$  of system  $s$ , we can write:

$$\rho_i = \sqrt{(x_i^{\text{Tx}} - x)^2 + (y_i^{\text{Tx}} - y)^2 + (z_i^{\text{Tx}} - z)^2} + c\Delta t^{(s)} + \sigma_e, \quad (3.1)$$

where  $(x_i^{\text{Tx}}, y_i^{\text{Tx}}, z_i^{\text{Tx}})$  is the satellite's position (known from the navigation message),  $(x, y, z)$  the receiver's position, and  $\sigma_e$  gathers other sources of error. Since the receiver needs to estimate its own 3D position (three spatial unknowns) and its clock deviation with respect to the satellites' time basis, at least  $3 + N_s$  satellites must be seen by the receiver at the same time, where  $N_s$  is the number of different navigation systems available (in-view) at a given time. Each received satellite signal, once synchronized and demodulated at the receiver, defines one equation such as the one defined above, forming a set of nonlinear equations that can be solved algebraically by means of the [Bancroft algorithm](#) or numerically, resorting to multidimensional Newton-Raphson and weighted least square methods. When *a priori* information is added we resort to Bayesian estimation, a problem that can be solved recursively by a Kalman filter or any of its variants. The problem can be further expanded by adding other unknowns (for instance, parameters of ionospheric and tropospheric models), sources of information from other systems, mapping information, and even motion models of the receiver. In the design of multi-constellation GNSS receivers, the vector of unknowns can also include the receiver clock offset with respect to each system in order to take advantage of a higher number of in-view satellites and using them jointly in the navigation solution, therefore increasing accuracy.

The [analytic representation](#) of a signal received from a GNSS satellite can be generically expressed as

$$r(t) = \alpha(t)s_T(t - \tau(t))e^{-j2\pi f_d(t)}e^{j2\pi f_c t} + n(t), \quad (3.2)$$

where  $\alpha(t)$  is the amplitude,  $s_T(t)$  is the complex baseband transmitted signal,  $\tau(t)$  is the time-varying delay,  $f_d(t) = f_c\tau(t)$  is the Doppler shift,  $f_c$  is the carrier frequency, and  $n(t)$  is a noise term. These signals arrive to the Earth's surface at extremely low power (e.g.  $-158.5$  dBW for GPS L1 C/A-code,  $-157$  dBW for Galileo E1), well below the noise floor. In order to estimate its distances to satellites, the receiver must correlate time-aligned replicas of the corresponding pseudorandom code with the incoming signal, in a process called *despreading* that

provides processing gain only to the signal of interest. After a coarse and fine estimation stages of the synchronization parameters (usually known as acquisition and tracking, respectively), signal processing output is in form of *observables*:

i) the pseudorange (code) measurement, equivalent to the difference of the time of reception (expressed in the time frame of the receiver) and the time of transmission (expressed in the time frame of the satellite) of a distinct satellite signal; and optionally

ii) the carrier-phase measurement, actually being a measurement on the beat frequency between the received carrier of the satellite signal and a receiver-generated reference frequency. Carrier phase measurements are ambiguous, in the sense that the integer number of carrier wavelengths between satellite and the receiver's antenna is unknown. Techniques such as **Least-square AMBiguity Decorrelation Approach (LAMBDA)** or Multi Carrier Ambiguity Resolution (MCAR) can be applied to resolve such ambiguity and provide an accurate estimation of the distance between the satellite and the receiver.

Then, depending on the required accuracy, the navigation solution can range from pseudorange-only, computationally low demanding, and limited accuracy least squares methods to sophisticated combinations of code and phase observables at different frequencies for high demanding applications such as surveying, geodesy, and geophysics.

Next sections provide brief descriptions of the space segment of different GNSSs and their broadcast signal structures accessible by civilians.

### 3.1.1 Global Positioning System (GPS) signal in space

The Global Positioning System (GPS) is a space-based radio-navigation system owned by the United States Government (USG) and operated by the United States Air Force (USAF). GPS provides positioning and timing services to military and civilian users on a continuous, worldwide basis. Two GPS services are provided: the Precise Positioning Service (PPS), available primarily to the military of the United States and its allies, and the Standard Positioning Service (SPS) open to civilian users.

- **GPS L1**. Defined at **Interface Specification IS-GPS-200 Revision M**, this band is centered at  $f_{\text{GPS L1}} = 1575.42$  MHz. The complex baseband transmitted signal can be written as

$$s_T^{(\text{GPS L1})}(t) = e_{L1I}(t) + je_{L1Q}(t), \quad (3.3)$$

with

$$e_{L1I}(t) = \sum_{l=-\infty}^{\infty} D_{\text{NAV}}[l]_{204600} \oplus C_{P(Y)}[l]_{L_{P(Y)}} p(t - lT_{c,P(Y)}), \quad (3.4)$$

$$e_{L1Q}(t) = \sum_{l=-\infty}^{\infty} D_{\text{NAV}}[l]_{20460} \oplus C_{C/A}[l]_{1023} p(t - lT_{c,C/A}), \quad (3.5)$$

where  $\oplus$  is the exclusive-or operation (modulo-2 addition),  $|l|_L$  means  $l$  modulo  $L$ ,  $[l]_L$  means the integer part of  $\frac{l}{L}$ ,  $D_{\text{NAV}}$  is the GPS navigation message bit sequence, transmitted at 50 bps,  $T_{c,P(Y)} = \frac{1}{10.23} \mu\text{s}$ ,  $T_{c,C/A} = \frac{1}{1.023} \mu\text{s}$ ,  $L_{P(Y)} = 6.1871 \cdot 10^{12}$ , and  $p(t)$  is a rectangular pulse of a chip-period duration centered at  $t = 0$  and filtered at the transmitter. According to the chip rate, the binary phase-shift keying modulations in the equations above are denoted as BPSK(10) and BPSK(1), respectively. The precision P codes (named Y codes whenever the anti-spoofing mode is activated, encrypting the code and thus denying non-U.S. military users) are sequences of 7 days in length. Regarding the modernization plans for GPS, it is worthwhile to mention that there is a new civilian-use signal planned, called L1C and defined at **Interface Specification IS-GPS-800 Revision H**, to be broadcast on the same L1 frequency that currently contains the C/A signal. The L1C is available with first Block III launch. The implementation provides C/A code to ensure backward compatibility.

- **GPS L2C.** Defined at **Interface Specification IS-GPS-200 Revision M**, is only available on Block IIR-M and subsequent satellite blocks. Centered at  $f_{\text{GPS L2}} = 1227.60$  MHz, the signal structure is the same than in (eq:GPSL1}), with the precision code in the In-phase component, just as in (eq:L1CAI}) but with an optional presence of the navigation message  $D_{\text{NAV}}$ . For the Quadrature-phase component, three options are defined:

$$e_{L2CQ}(t) = \sum_{l=-\infty}^{\infty} D_{\text{CNAV}}[l]_{10230} \oplus \left( C_{\text{CL}}[l]_{L_{\text{CL}}} p_{1/2}(t - lT_{c,L2C}) + \right. \quad (3.6)$$

$$\left. + C_{\text{CM}}[l]_{L_{\text{CM}}} p_{1/2}\left(t - \left(l + \frac{3}{4}\right) T_{c,L2C}\right) \right), \quad (3.7)$$

$$e_{L2CQ}(t) = \sum_{l=-\infty}^{\infty} D_{\text{NAV}}[l]_{20460} \oplus C_{\text{C/A}}[l]_{1023} p(t - lT_{c,C/A}), \text{ or} \quad (3.8)$$

$$e_{L2CQ}(t) = \sum_{l=-\infty}^{\infty} C_{\text{C/A}}[l]_{1023} p(t - lT_{c,C/A}), \quad (3.9)$$

where  $T_{c,L2C} = \frac{1}{511.5}$  ms and  $p_{1/2}(t)$  is a rectangular pulse of half chip-period duration, thus time-multiplexing both codes. The civilian long code  $C_{\text{CL}}$  is  $L_{\text{CL}} = 767250$  chips long, repeating every 1.5 s, while the civilian moderate code  $C_{\text{CM}}$  is  $L_{\text{CL}} = 10230$  chips long and its repeats every 20 ms. The CNAV data is an upgraded version of the original NAV navigation message, containing higher precision representation and nominally more accurate data than the NAV data. It is transmitted at 25 bps with forward error correction (FEC) encoding, resulting in 50 sps.

- **GPS L5.** The GPS L5 link, defined at **Interface Specification IS-GPS-705 Revision H**, is only available in Block IIF (first satellite launched on May, 2010) and subsequent satellite blocks. Centered at  $f_{\text{GPS L5}} = 1176.45$  MHz, this signal in space can be written as:

$$s_T^{(\text{GPS L5})}(t) = e_{L5I}(t) + je_{L5Q}(t), \quad (3.10)$$

$$e_{L5I}(t) = \sum_{m=-\infty}^{+\infty} C_{nh10}[m]_{10} \oplus D_{\text{CNAV}}[m]_{10} \oplus \sum_{l=1}^{102300} C_{L5I}[l]_{10230} p(t - mT_{c,nh} - lT_{c,L5}), \quad (3.11)$$

$$e_{L5Q}(t) = \sum_{m=-\infty}^{+\infty} C_{nh20}[m]_{20} \oplus \sum_{l=1}^{102300} C_{L5Q}[l]_{10230} \cdot p(t - mT_{c,nh} - lT_{c,L5}), \quad (3.12)$$

where  $T_{c,nh} = 1$  ms and  $T_{c,L5} = \frac{1}{10.23}$   $\mu$ s, thus defining a BPSK(10) modulation. Both L5I and L5Q contain synchronization sequences.

### 3.1.2 GLONASS signal in space

The nominal baseline constellation of the Russian Federation's Global Navigation Satellite System (GLONASS) comprises 24 GLONASS-M satellites that are uniformly deployed in three roughly circular orbital planes at an inclination of  $64.8^\circ$  to the equator. The altitude of the orbit is 19,100 km. The orbit period of each satellite is 11 hours, 15 minutes, and 45 seconds. The orbital planes are separated by  $120^\circ$  right ascension of the ascending node. Eight satellites are equally spaced in each plane with  $45^\circ$  argument of latitude. Moreover, the orbital planes have an argument of latitude displacement of  $15^\circ$  relative to each other.

GLONASS civil signal-in-space is defined at **Interface Control Document. Navigational radiosignal in bands L1, L2. Edition 5.1**. This system makes use of a frequency-division multiple access (FDMA) signal structure, transmitting in two bands:  $f_{\text{GLOL1}}^{(k)} = 1602 + k \cdot 0.5625$  MHz and  $f_{\text{GLOL2}}^{(k)} = 1246 + k \cdot 0.4375$  MHz, where  $k \in \{-7, -6, \dots, 5, 6\}$  is the channel number. Satellites in opposite points of an orbit plane transmit signals on equal frequencies, as these satellites will never be in view simultaneously by a ground-based user.

- **GLONASS L1.** Two kind of signals are transmitted: a standard precision (SP) and an obfuscated high precision (HP) signal. The complex baseband transmitted signal can be written as

$$s_T^{(\text{GLO L1})}(t) = e_{L1I}(t) + je_{L1Q}(t), \quad (3.13)$$

with BPSK(5) and BPSK(0.5) modulations:

$$e_{L1I}(t) = \sum_{l=-\infty}^{\infty} D_{\text{GNAV}}[l]_{102200} \oplus C_{\text{HP}}[l]_{L_{\text{HP}}} p(t - lT_{c,\text{HP}}), \quad (3.14)$$

$$e_{L1Q}(t) = \sum_{l=-\infty}^{\infty} D_{\text{GNAV}}[l]_{10220} \oplus C_{\text{SP}}[l]_{511} p(t - lT_{c,\text{SP}}), \quad (3.15)$$

where  $T_{c,\text{HP}} = \frac{1}{5.11} \mu\text{s}$ ,  $T_{c,\text{SP}} = \frac{1}{0.511} \mu\text{s}$ , and  $L_{\text{HP}} = 3.3554 \cdot 10^7$ . The navigation message  $D_{\text{GNAV}}$  is transmitted at 50 bps. Details of its content and structure, as well as the generation of the  $C_{\text{SP}}$  code, can be found at the [ICD](#). The usage of the HP signal should be agreed with the Russian Federation Defense Ministry, and no more details have been disclosed.

- **GLONASS L2.** Beginning with the second generation of satellites, called GLONASS-M and first launched in 2001, a second civil signal is available using the same SP code than the one in the L1 band.

The use of FDMA techniques, in which the same code is used to broadcast navigation signals on different frequencies, and the placement of civil GLONASS transmissions on frequencies close to 1600 MHz, well above the GPS L1 band, have complicated the design of combined GLONASS/GPS receivers, particularly low-cost equipment for mass-market applications. Future plans of modernization are intended to increase compatibility and interoperability with other GNSS, and include the addition of a code-division multiple access (CDMA) structure, and possibly binary offset carrier (BOC) modulation, beginning with the third civil signal in the L3 band (1197.648 – 1212.255 MHz). Russia is implementing the new signals on the next-generation GLONASS-K satellites, with a first prototype successfully launched into orbit on February 26, 2011.

### 3.1.3 Galileo signal in space

The nominal Galileo constellation comprises a total of 27 operational satellites (plus 3 active spares), that are evenly distributed among three orbital planes inclined at  $56^\circ$  relative to the equator. There are nine operational satellites per orbital plane, occupying evenly distributed orbital slots. Three additional spare satellites (one per orbital plane) complement the nominal constellation configuration. The Galileo satellites are placed in quasi-circular Earth orbits with a nominal semi-major axis of about 30,000 km and an approximate revolution period of 14 hours. The Control segment full infrastructure will be composed of 30 – 40 sensor stations, 3 control centers, 9 Mission Uplink stations, and 5 TT&C stations.

Galileo's Open Service is defined at [Signal In Space Interface Control Document](#). [Ref↔](#): [OS SIS ICD, Issue 2.0](#), where the following signal structures are specified:

- **Galileo E1.** This band, centered at  $f_{\text{Gal E1}} = 1575.420$  MHz and with a reference bandwidth of 24.5520 MHz, uses the so-called composite binary offset carrier CBOC(6,1,  $\frac{1}{11}$ ) modulation, defined in baseband as:

$$s_T^{(\text{Gal E1})}(t) = \frac{1}{\sqrt{2}} \left( e_{E1B}(t) (\alpha sc_A(t) + \beta sc_B(t)) + \right. \quad (3.16)$$

$$\left. - e_{E1C}(t) (\alpha sc_A(t) - \beta sc_B(t)) \right), \quad (3.17)$$

where the subcarriers  $sc(t)$  are defined as

$$sc_A(t) = \text{sign} \left( \sin(2\pi f_{s,E1A} t) \right), \quad (3.18)$$

$$sc_B(t) = \text{sign} \left( \sin(2\pi f_{s,E1B} t) \right), \quad (3.19)$$

and  $f_{s,E1A} = 1.023$  MHz,  $f_{s,E1B} = 6.138$  MHz are the subcarrier rates,  $\alpha = \sqrt{\frac{10}{11}}$ , and  $\beta = \sqrt{\frac{1}{11}}$ . Channel B contains the I/NAV type of navigation message,  $D_{I/NAV}$ , intended for Safety-of-Life (SoL) services:

$$e_{E1B}(t) = \sum_{l=-\infty}^{+\infty} D_{I/NAV} \left[ [l]_{4092} \right] \oplus C_{E1B} \left[ [l]_{4092} \right] p(t - lT_{c,E1B}). \quad (3.20)$$

In case of channel C, it is a pilot (dataless) channel with a secondary code, forming a tiered code:

$$e_{E1C}(t) = \sum_{m=-\infty}^{+\infty} C_{E1Cs} \left[ [m]_{25} \right] \oplus \sum_{l=1}^{4092} C_{E1Cp} \left[ l \right] \cdot p(t - mT_{c,E1Cs} - lT_{c,E1Cp}), \quad (3.21)$$

with  $T_{c,E1B} = T_{c,E1Cp} = \frac{1}{1.023} \mu\text{s}$  and  $T_{c,E1Cs} = 4$  ms. The  $C_{E1B}$  and  $C_{E1Cp}$  primary codes are pseudorandom memory code sequences defined at Annex C of [OS SIS ICD](#). The binary sequence of the secondary code  $C_{E1Cs}$  is 0011100000001010110110010. This band also contains another component, Galileo E1A, intended for the Public Regulated Service (PRS). It uses a BOC(15,2.5) modulation with cosine-shaped subcarrier  $f_{s,E1A} = 15.345$  MHz and  $T_{c,E1A} = \frac{1}{2.5575} \mu\text{s}$ . The PRS spreading codes and the structure of the navigation message have not been made public.

- **Galileo E6.** Intended for the Commercial Service and centered at  $f_{\text{Gal E6}} = 1278.750$  MHz, this band provides pilot and data components

$$s_T^{(\text{Gal E6})}(t) = \frac{1}{\sqrt{2}} (e_{E6B}(t) - e_{E6C}(t)), \quad (3.22)$$

$$e_{E6B}(t) = \sum_{m=-\infty}^{+\infty} D_{C/NAV} \left[ [l]_{5115} \right] \oplus C_{E6B} \left[ [l]_{L_{E6B}} \right] \cdot p(t - lT_{c,E6}), \quad (3.23)$$

$$e_{E6C}(t) = \sum_{m=-\infty}^{+\infty} C_{E6Cs} \left[ [m]_{100} \right] \oplus \sum_{l=1}^{L_{E6C}} C_{E6Cp} \left[ l \right] \cdot p(t - mT_{c,E6s} - lT_{c,E6p}), \quad (3.24)$$

where  $D_{C/NAV}$  is the C/NAV navigation data stream, which is modulated with the encrypted ranging code  $C_{E6B}$  with chip period  $T_{c,E6} = \frac{1}{5.115} \mu\text{s}$ , thus being a BPSK(5) modulation. Codes  $C_{E6B}$  and primary codes  $C_{E6Cs}$  and their respective lengths,  $L_{E6B}$  and  $L_{E6C}$ , have not been published. The secondary codes for the pilot component,  $C_{E6Cs}$ , are available at the [OS SIS ICD](#). The receiver reference bandwidth for this signal is 40.920 MHz. This band also contains another component, Galileo E6A, intended for PRS.

- **Galileo E5.** Centered at  $f_{\text{Gal E5}} = 1191.795$  MHz and with a total bandwidth of 51.150 MHz, its signal structure deserves some analysis. The AltBOC modulation can be generically expressed as

$$s^{\text{AltBOC}}(t) = x_1(t)v^*(t) + x_2(t)v(t), \quad (3.25)$$

where  $v(t) = \frac{1}{\sqrt{2}} (\text{sign}(\cos(2\pi f_s t)) + j \text{sign}(\sin(2\pi f_s t)))$  is the single side-band subcarrier,  $f_s$  is the subcarrier frequency,  $(\cdot)^*$  stands for the conjugate operation, and  $x_1(t)$  and  $x_2(t)$  are QPSK signals. The resulting waveform does not exhibit constant envelope. In case of Galileo, the need for high efficiency of the satellites' onboard High Power Amplifier (HPA) has pushed a modification on the signal in order to make it envelope-constant and thus use the HPA at saturation. This can be done by adding some inter-modulation products to the expression above, coming up with the following definition:

$$s_T^{(\text{Gal E5})}(t) = e_{E5a}(t)ssc_s^*(t) + e_{E5b}(t)ssc_s(t) + \bar{e}_{E5a}(t)ssc_p^*(t) + \bar{e}_{E5b}(t)ssc_p(t), \quad (3.26)$$

where the single and product side-band signal subcarriers are

$$ssc_s(t) = sc_s(t) + jsc_s\left(t - \frac{T_s}{4}\right), \quad (3.27)$$

$$ssc_p(t) = sc_p(t) + jsc_p\left(t - \frac{T_s}{4}\right), \quad (3.28)$$

and

$$e_{E5a}(t) = e_{E5aI}(t) + je_{E5aQ}(t), \quad (3.29)$$

$$e_{E5b}(t) = e_{E5bI}(t) + je_{E5bQ}(t), \quad (3.30)$$

$$\bar{e}_{E5a}(t) = \bar{e}_{E5aI}(t) + j\bar{e}_{E5aQ}(t), \quad (3.31)$$

$$\bar{e}_{E5b}(t) = \bar{e}_{E5bI}(t) + j\bar{e}_{E5bQ}(t), \quad (3.32)$$

$$\bar{e}_{E5aI}(t) = e_{E5aQ}(t)e_{E5bI}(t)e_{E5bQ}(t), \quad (3.33)$$

$$\bar{e}_{E5aQ}(t) = e_{E5aI}(t)e_{E5bI}(t)e_{E5bQ}(t), \quad (3.34)$$

$$\bar{e}_{E5bI}(t) = e_{E5bQ}(t)e_{E5aI}(t)e_{E5aQ}(t), \quad (3.35)$$

$$\bar{e}_{E5bQ}(t) = e_{E5bI}(t)e_{E5aI}(t)e_{E5aQ}(t). \quad (3.36)$$

The signal components are defined as

$$e_{E5aI}(t) = \sum_{m=-\infty}^{+\infty} C_{E5aIs} \left[ |m|_{20} \right] \oplus \sum_{l=1}^{10230} C_{E5aIp} \left[ l \right] \oplus \quad (3.37)$$

$$\oplus D_{F/NAV} \left[ [l]_{204600} \right] p(t - mT_{c,E5s} - lT_{c,E5p}), \quad (3.38)$$

$$e_{E5aQ}(t) = \sum_{m=-\infty}^{+\infty} C_{E5aQs} \left[ |m|_{100} \right] \oplus \sum_{l=1}^{10230} C_{E5aQp} \left[ l \right] \cdot \quad (3.39)$$

$$\cdot p(t - mT_{c,E5s} - lT_{c,E5p}), \quad (3.40)$$

$$e_{E5bI}(t) = \sum_{m=-\infty}^{+\infty} C_{E5bIs} \left[ |m|_4 \right] \oplus \sum_{l=1}^{10230} C_{E5bIp} \left[ l \right] \oplus \quad (3.41)$$

$$\oplus D_{I/NAV} \left[ [l]_{40920} \right] p(t - mT_{c,E5s} - lT_{c,E5p}), \quad (3.42)$$

$$e_{E5bQ}(t) = \sum_{m=-\infty}^{+\infty} C_{E5bQs} \left[ |m|_{100} \right] \oplus \sum_{l=1}^{10230} C_{E5bQp} \left[ l \right] \cdot \quad (3.43)$$

$$\cdot p(t - mT_{c,E5s} - lT_{c,E5p}), \quad (3.44)$$

where  $T_{c,E5s} = 1$  ms and  $T_{c,E5p} = \frac{1}{10.23} \mu\text{s}$ . **Channel A** contains the F/NAV type of navigation message,  $D_{F/NAV}$ , intended for the Open Service. The I/NAV message structures for the E5bI and E1B signals use the same page layout. Only page sequencing is different, with page swapping between both components in order to allow a fast reception of data by a dual frequency receiver. The single subcarrier  $sc_s(t)$  and the product subcarrier  $sc_p(t)$  are defined as:

$$sc_s(t) = \frac{\sqrt{2}}{4} \text{sign} \left( \cos \left( 2\pi f_s t - \frac{\pi}{4} \right) \right) + \quad (3.45)$$

$$+ \frac{1}{2} \text{sign} \left( \cos (2\pi f_s t) \right) + \frac{\sqrt{2}}{4} \text{sign} \left( \cos \left( 2\pi f_s t + \frac{\pi}{4} \right) \right), \quad (3.46)$$

$$sc_p(t) = -\frac{\sqrt{2}}{4} \text{sign} \left( \cos \left( 2\pi f_s t - \frac{\pi}{4} \right) \right) + \quad (3.47)$$

$$+ \frac{1}{2} \text{sign} \left( \cos (2\pi f_s t) \right) - \frac{\sqrt{2}}{4} \text{sign} \left( \cos \left( 2\pi f_s t + \frac{\pi}{4} \right) \right), \quad (3.48)$$



with a subcarrier frequency of  $f_s = 15.345$  MHz, thus defining an AltBOC(15,10) modulation. The QPSK(10) signal  $e_{E5a}(t)$  defined above is shifted to  $f_{\text{Gal E5a}} \doteq f_{\text{Gal E5}} - f_s = 1176.450$  MHz, while  $e_{E5b}(t)$  is shifted to  $f_{\text{Gal E5b}} \doteq f_{\text{Gal E5}} + f_s = 1207.140$  MHz. Thus, we can bandpass filter around  $f_{\text{Gal E5a}}$  and get a good approximation of a QPSK(10) signal, with very low energy components of  $e_{E5b}(t)$ ,  $\bar{e}_{E5a}(t)$ , and  $\bar{e}_{E5b}(t)$ :

$$s_T^{(\text{Gal E5a})}(t) \simeq e_{E5aI}(t) + je_{E5aQ}(t). \quad (3.49)$$

The same applies to  $e_{E5b}(t)$ , allowing an independent reception of two QPSK(10) signals and thus requiring considerably less bandwidth than the processing of the whole E5 band.

### 3.1.4 Reference

This text is an except of the following paper:

- C. Fernández-Prades, L. Lo Presti, E. Falletti, *Satellite Radiolocalization From GPS to GNSS and Beyond: Novel Technologies and Applications for Civil Mass↔Market*. Proceedings of the IEEE. Vol 99, No. 11, pp. 1882-1904. November, 2011. doi: 10.1109/JPROC.2011.2158032



## Chapter 4

# Todo List

Member **ALPHA\_0** ({{69, 8}})

read all pages of subframe 4

Member **T\_OA** ({{69, 8}})

read all pages of subframe 5



# Chapter 5

## Topic Index

### 5.1 Topics

Here is a list of all topics with brief descriptions:

Acquisition . . . . .	75
acquisition_adapters . . . . .	76
acquisition_gr_blocks . . . . .	77
acquisition_libs . . . . .	80
Channel . . . . .	81
channel_adapters . . . . .	81
channel_libs . . . . .	82
Signal Conditioner . . . . .	83
conditioner_adapters . . . . .	83
Data Type Adapters . . . . .	84
data_type_adapters . . . . .	84
data_type_gr_blocks . . . . .	85
Input Filter . . . . .	87
input_filter_adapters . . . . .	87
input_filter_gr_blocks . . . . .	88
Algorithms Common Library . . . . .	90
algorithms_libs . . . . .	90
gnss_sdr_flags . . . . .	113
PVT . . . . .	118
algorithms_libs_rtklib . . . . .	118
pvt_adapters . . . . .	169
pvt_gr_blocks . . . . .	170
pvt_libs . . . . .	170
Observables . . . . .	173
obs_adapters . . . . .	173
obs_gr_blocks . . . . .	174
observables_libs . . . . .	175
Resampler . . . . .	175
resampler_adapters . . . . .	176
resampler_gr_blocks . . . . .	176
Signal Source . . . . .	178
signal_source_adapters . . . . .	178

signal_source_gr_blocks . . . . .	180
signal_source_libs . . . . .	183
Telemetry Decoder . . . . .	185
telemetry_decoder_adapters . . . . .	186
telemetry_decoder_gr_blocks . . . . .	187
telemetry_decoder_libswiftcnv . . . . .	189
telemetry_decoder_libs . . . . .	192
Tracking . . . . .	193
tracking_adapters . . . . .	193
tracking_gr_blocks . . . . .	195
tracking_libs . . . . .	198
Core GNSS Receiver . . . . .	205
GNSS block interfaces . . . . .	206
core_libs . . . . .	207
core_monitor . . . . .	210
core_receiver . . . . .	211
core_system_parameters . . . . .	212

## Chapter 6

# Hierarchical Index

### 6.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Acq_Conf . . . . .	324
Acq_Conf_Fpga . . . . .	329
ad936x_iio_custom . . . . .	337
ad936x_iio_samples . . . . .	338
Agnss_Ref_Location . . . . .	349
Agnss_Ref_Time . . . . .	350
Allocator . . . . .	352
alm_t . . . . .	353
ambc_t . . . . .	356
An_Packet_Printer . . . . .	357
an_packet_t . . . . .	358
Bayesian_estimator . . . . .	363
Beidou_Dnav_Navigation_Message . . . . .	387
Beidou_Dnav_Utc_Model . . . . .	390
gr::block	
Galileo_E1_Tcp_Connector_Tracking_cc . . . . .	607
Glonass_L1_Ca_Dll_Pll_Tracking_cc . . . . .	814
Glonass_L2_Ca_Dll_Pll_Tracking_cc . . . . .	821
Gps_L1_Ca_Dll_Pll_Tracking_GPU_cc . . . . .	947
Gps_L1_Ca_Gaussian_Tracking_cc . . . . .	949
Gps_L1_Ca_Tcp_Connector_Tracking_cc . . . . .	950
Notch . . . . .	1197
NotchLite . . . . .	1205
ad936x_iio_source . . . . .	339
beidou_b1i_telemetry_decoder_gs . . . . .	369
beidou_b3i_telemetry_decoder_gs . . . . .	371
channel_msg_receiver_cc . . . . .	431
channel_status_msg_receiver . . . . .	433
direct_resampler_conditioner_cb . . . . .	483
direct_resampler_conditioner_cc . . . . .	485
direct_resampler_conditioner_cs . . . . .	486
dll_pll_veml_tracking . . . . .	504
dll_pll_veml_tracking_fpga . . . . .	505
galileo_e5a_noncoherentIQ_acquisition_caf_cc . . . . .	608
galileo_e6_has_msg_receiver . . . . .	614

galileo_pcps_8ms_acquisition_cc . . . . .	645
galileo_telemetry_decoder_gs . . . . .	653
galileo_tow_map . . . . .	656
glonass_l1_ca_dll_pll_c_aid_tracking_cc . . . . .	811
glonass_l1_ca_dll_pll_c_aid_tracking_sc . . . . .	813
glonass_l1_ca_telemetry_decoder_gs . . . . .	816
glonass_l2_ca_dll_pll_c_aid_tracking_cc . . . . .	818
glonass_l2_ca_dll_pll_c_aid_tracking_sc . . . . .	820
glonass_l2_ca_telemetry_decoder_gs . . . . .	823
gnss_sdr_fpga_sample_counter . . . . .	882
gnss_sdr_time_counter . . . . .	892
gnss_synchro_monitor . . . . .	905
gps_l1_ca_telemetry_decoder_gs . . . . .	952
gps_l2c_telemetry_decoder_gs . . . . .	954
gps_l5_telemetry_decoder_gs . . . . .	956
hybrid_observables_gs . . . . .	1091
kf_tracking . . . . .	1139
labsat23_source . . . . .	1141
nav_message_monitor . . . . .	1184
osnma_msg_receiver . . . . .	1231
pcps_acquisition . . . . .	1242
pcps_acquisition_fine_doppler_cc . . . . .	1246
pcps_assisted_acquisition_cc . . . . .	1256
pcps_cccwsr_acquisition_cc . . . . .	1261
pcps_openc1_acquisition_cc . . . . .	1266
pcps_quicksync_acquisition_cc . . . . .	1272
pcps_tong_acquisition_cc . . . . .	1278
pulse_blanking_cc . . . . .	1303
sbas_l1_telemetry_decoder_gs . . . . .	1388
signal_generator_c . . . . .	1423
Channel_Event . . . . .	431
ChannelFsm . . . . .	435
cl_fft_plan . . . . .	441
clFFT_Complex . . . . .	444
clFFT_Dim3 . . . . .	445
clFFT_SplitComplex . . . . .	446
cnav_msg_decoder_t . . . . .	446
cnav_msg_t . . . . .	448
cnav_v27_part_t . . . . .	449
Command_Event . . . . .	452
Concurrent_Map< Data > . . . . .	456
Concurrent_Queue< Data > . . . . .	457
ConfigurationInterface . . . . .	459
FileConfiguration . . . . .	540
InMemoryConfiguration . . . . .	1108
ControlThread . . . . .	465
Cpu_Multicorrelator . . . . .	467
Cpu_Multicorrelator_16sc . . . . .	468
Cpu_Multicorrelator_Real_Codes . . . . .	469
CubatureFilter . . . . .	476
cuda_multicorrelator . . . . .	476
Deleter . . . . .	481
dgps_t . . . . .	482
Dll_Pll_Conf . . . . .	492
Dll_Pll_Conf_Fpga . . . . .	498
DSM_dsm_header . . . . .	512
DSM_KROOT_message . . . . .	513
DSM_nma_header . . . . .	516



DSM_PKR_message . . . . .	517
eph_t . . . . .	519
erp_t . . . . .	524
erpd_t . . . . .	525
Exponential_Smoother . . . . .	526
exterr_t . . . . .	528
fcbsd_t . . . . .	530
file_t . . . . .	537
Fpga_Acquisition . . . . .	576
Fpga_buffer_monitor . . . . .	579
Fpga_DMA . . . . .	581
Fpga_dynamic_bit_selection . . . . .	582
Fpga_Multicorrelator_8sc . . . . .	583
Fpga_spidev . . . . .	588
Fpga_Switch . . . . .	589
ftp_t . . . . .	594
Galileo_Almanac_Helper . . . . .	599
Galileo_Cnav_Message . . . . .	605
Galileo_Fnav_Message . . . . .	623
Galileo_HAS_data . . . . .	625
Galileo_HAS_page . . . . .	635
Galileo_Inav_Message . . . . .	638
Galileo_Iono . . . . .	641
Galileo_ISM . . . . .	644
Galileo_Reduced_CED . . . . .	650
Galileo_Utc_Model . . . . .	657
Geohash . . . . .	785
GeoJSON_Printer . . . . .	786
geph_t . . . . .	788
Glonass_Gnav_Almanac . . . . .	790
Glonass_Gnav_Ephemeris . . . . .	794
Glonass_Gnav_Navigation_Message . . . . .	804
Glonass_Gnav_Utc_Model . . . . .	809
Gnss_Almanac . . . . .	859
Beidou_Dnav_Almanac . . . . .	374
Galileo_Almanac . . . . .	596
Gps_Almanac . . . . .	918
Gnss_circular_deque< T > . . . . .	865
Gnss_Crypto . . . . .	867
Gnss_Ephemeris . . . . .	870
Beidou_Dnav_Ephemeris . . . . .	377
Galileo_Ephemeris . . . . .	616
Gps_CNAV_Ephemeris . . . . .	922
Gps_Ephemeris . . . . .	936
Gnss_Satellite . . . . .	878
Gnss_Sdr_Supl_Client . . . . .	885
Gnss_Signal . . . . .	896
Gnss_Synchro . . . . .	897
Gnss_Synchro_Udp_Sink . . . . .	906
GNSSBlockFactory . . . . .	907
GNSSBlockInterface . . . . .	908
AcquisitionInterface . . . . .	332
BeidouB1iPcpsAcquisition . . . . .	397
BeidouB3iPcpsAcquisition . . . . .	411
GalileoE1Pcps8msAmbiguousAcquisition . . . . .	673
GalileoE1PcpsAmbiguousAcquisition . . . . .	679
GalileoE1PcpsAmbiguousAcquisitionFpga . . . . .	685
GalileoE1PcpsCccwsrAmbiguousAcquisition . . . . .	692

GalileoE1PcpsQuickSyncAmbiguousAcquisition	697
GalileoE1PcpsTongAmbiguousAcquisition	703
GalileoE5aNoncoherentIQAcquisitionCaf	723
GalileoE5aPcpsAcquisition	729
GalileoE5aPcpsAcquisitionFpga	734
GalileoE5bPcpsAcquisition	750
GalileoE5bPcpsAcquisitionFpga	756
GalileoE6PcpsAcquisition	771
GlonassL1CaPcpsAcquisition	833
GlonassL2CaPcpsAcquisition	850
GpsL1CaPcpsAcquisition	985
GpsL1CaPcpsAcquisitionFineDoppler	991
GpsL1CaPcpsAcquisitionFpga	997
GpsL1CaPcpsAssistedAcquisition	1003
GpsL1CaPcpsOpenCIAcquisition	1009
GpsL1CaPcpsQuickSyncAcquisition	1014
GpsL1CaPcpsTongAcquisition	1020
GpsL2MPcpsAcquisition	1044
GpsL2MPcpsAcquisitionFpga	1050
GpsL5iPcpsAcquisition	1064
GpsL5iPcpsAcquisitionFpga	1070
ArraySignalConditioner	359
BeamformerFilter	366
ByteToShort	422
ChannelInterface	436
Channel	426
CshortToGrComplex	473
DirectResamplerConditioner	488
FirFilter	558
FreqXlatingFirFilter	590
IbyteToCbyte	1097
IbyteToComplex	1101
IbyteToCshort	1104
IshortToComplex	1125
IshortToCshort	1129
MmseResamplerConditioner	1166
NotchFilter	1199
NotchFilterLite	1202
ObservablesInterface	1219
HybridObservables	1093
Pass_Through	1237
PulseBlankingFilter	1304
PvtInterface	1320
Rtklib_Pvt	1359
SignalConditioner	1424
SignalGenerator	1428
SignalSourceInterface	1435
GenSignalSource	781
SignalSourceBase	1432
Ad936xCustomSignalSource	341
Adrv9361z7035SignalSourceFPGA	346
CustomUDPSignalSource	477
DMASignalSourceFPGA	509
FifoSignalSource	533
FileSourceBase	547
FileSignalSource	543
FileTimestampSignalSource	553
FourBitCpxFileSignalSource	572

NsrFileSignalSource . . . . .	1206
SpirFileSignalSource . . . . .	1448
TwoBitCpxFileSignalSource . . . . .	1502
TwoBitPackedFileSignalSource . . . . .	1506
FlexibandSignalSource . . . . .	561
Fmcomms2SignalSource . . . . .	566
Fmcomms5SignalSourceFPGA . . . . .	569
IONGSMSSignalSource . . . . .	1122
LabsatSignalSource . . . . .	1143
LimesdrSignalSource . . . . .	1152
MAX2771EVKITSignalSourceFPGA . . . . .	1163
MultichannelFileSignalSource . . . . .	1180
OsmosdrSignalSource . . . . .	1223
PlutosdrSignalSource . . . . .	1289
RawArraySignalSource . . . . .	1327
RtlTcpSignalSource . . . . .	1382
SpirGSS6450FileSignalSource . . . . .	1453
UhdSignalSource . . . . .	1511
ZmqSignalSource . . . . .	1533
TelemetryDecoderInterface . . . . .	1483
BeidouB1iTelemetryDecoder . . . . .	403
BeidouB3iTelemetryDecoder . . . . .	417
GalileoE1BTelemetryDecoder . . . . .	660
GalileoE5aTelemetryDecoder . . . . .	741
GalileoE5bTelemetryDecoder . . . . .	763
GalileoE6TelemetryDecoder . . . . .	777
GlonassL1CaTelemetryDecoder . . . . .	839
GlonassL2CaTelemetryDecoder . . . . .	856
GpsL1CaTelemetryDecoder . . . . .	1028
GpsL2CTelemetryDecoder . . . . .	1032
GpsL5TelemetryDecoder . . . . .	1077
SbasL1TelemetryDecoder . . . . .	1390
TrackingInterface . . . . .	1498
BeidouB1iDIIPITracking . . . . .	393
BeidouB3iDIIPITracking . . . . .	407
GalileoE1DIIPIVemITracking . . . . .	664
GalileoE1DIIPIVemITrackingFpga . . . . .	668
GalileoE1TcpConnectorTracking . . . . .	709
GalileoE5aDIIPITracking . . . . .	713
GalileoE5aDIIPITrackingFpga . . . . .	717
GalileoE5bDIIPITracking . . . . .	745
GalileoE6DIIPITracking . . . . .	767
GlonassL1CaDIIPICaIdTracking . . . . .	825
GlonassL1CaDIIPITracking . . . . .	829
GlonassL2CaDIIPICaIdTracking . . . . .	842
GlonassL2CaDIIPITracking . . . . .	846
GpsL1CaDIIPITracking . . . . .	964
GpsL1CaDIIPITrackingFpga . . . . .	968
GpsL1CaDIIPITrackingGPU . . . . .	973
GpsL1CaGaussianTracking . . . . .	977
GpsL1CaKfTracking . . . . .	981
GpsL1CaTcpConnectorTracking . . . . .	1025
GpsL2MDIIPITracking . . . . .	1036
GpsL2MDIIPITrackingFpga . . . . .	1040
GpsL5DIIPITracking . . . . .	1056
GpsL5DIIPITrackingFpga . . . . .	1059
GNSSFlowgraph . . . . .	911
GnssTime . . . . .	915

Gps_Acq_Assist . . . . .	916
Gps_CNAV_Navigation_Message . . . . .	932
Gps_Iono . . . . .	944
Beidou_Dnav_Iono . . . . .	384
Gps_CNAV_Iono . . . . .	930
Gps_Navigation_Message . . . . .	958
Gps_Utc_Model . . . . .	960
Gps_CNAV_Utc_Model . . . . .	934
GPU_Complex . . . . .	1080
GPU_Complex_Short . . . . .	1082
Gpx_Printer . . . . .	1083
gtime_t . . . . .	1086
half_cyc_tag . . . . .	1087
HAS_clock_corrections . . . . .	1088
HAS_obs_corrections . . . . .	1089
HAS_orbit_corrections . . . . .	1090
Has_Simple_Printer . . . . .	1091
INIReader . . . . .	1107
IONGSMSChunkData . . . . .	1117
IONGSMSChunkUnpackingCtx< WT > . . . . .	1118
kernel_info_t . . . . .	1132
Kf_Conf . . . . .	1134
Kml_Printer . . . . .	1140
lex_t . . . . .	1146
lexeph_t . . . . .	1148
lexion_t . . . . .	1150
lexmsg_t . . . . .	1151
MACK_header . . . . .	1156
Mack_lookup . . . . .	1157
MACK_message . . . . .	1159
MACK_tag_and_info . . . . .	1161
MACK_tag_info . . . . .	1162
ModelFunction . . . . .	1169
Monitor_Ephemeris_Udp_Sink . . . . .	1169
Monitor_Pvt . . . . .	1170
Monitor_Pvt_Udp_Sink . . . . .	1176
msm_h_t . . . . .	1177
mt1_header . . . . .	1179
Nav_Message_Packet . . . . .	1186
Nav_Message_Udp_Sink . . . . .	1189
nav_t . . . . .	1190
Nmea_Printer . . . . .	1196
ntrip_t . . . . .	1212
Obs_Conf . . . . .	1214
obs_t . . . . .	1216
obsd_t . . . . .	1217
opt_t . . . . .	1222
OSNMA_data . . . . .	1226
OSNMA_DSM_Reader . . . . .	1228
Osnma_Helper . . . . .	1229
OSNMA_msg . . . . .	1230
OSNMA_NavData . . . . .	1234
OSNMA_NavDataManager . . . . .	1236
pclk_t . . . . .	1241
pcps_acquisition_fpga . . . . .	1252
pcv_t . . . . .	1284
pcvs_t . . . . .	1286
peph_t . . . . .	1287

pppcorr_t . . . . .	1292
pps_tcp_rx . . . . .	1294
PpsSamplestamp . . . . .	1295
prcopt_t . . . . .	1296
Pvt_Conf . . . . .	1307
Pvt_Kf . . . . .	1314
Pvt_Solution . . . . .	1315
Rtklib_Solver . . . . .	1368
raw_t . . . . .	1323
ReedSolomon . . . . .	1331
Rinex_Printer . . . . .	1333
Rtcm . . . . .	1337
Rtcm_Printer . . . . .	1350
rtcm_t . . . . .	1353
rtk_t . . . . .	1357
rtksvr_t . . . . .	1375
Rtl_Tcp_Dongle_Info . . . . .	1379
Sbas_Ephemeris . . . . .	1386
sbs_t . . . . .	1394
sbsfcorr_t . . . . .	1396
sbsigp_t . . . . .	1397
sbsigpband_t . . . . .	1399
sbsion_t . . . . .	1400
sbslcorr_t . . . . .	1401
sbsmsg_t . . . . .	1402
sbssat_t . . . . .	1404
sbssatp_t . . . . .	1406
sdr_gnss_packet_t . . . . .	1407
seph_t . . . . .	1409
Serdes_Galileo_Eph . . . . .	1411
Serdes_Gnss_Synchro . . . . .	1413
Serdes_Gps_Eph . . . . .	1415
Serdes_Monitor_Pvt . . . . .	1417
Serdes_Nav_Message . . . . .	1419
serial_t . . . . .	1421
snrmask_t . . . . .	1437
sol_t . . . . .	1438
solbuf_t . . . . .	1440
soloft_t . . . . .	1442
solstat_t . . . . .	1445
solstatbuf_t . . . . .	1447
ssat_t . . . . .	1458
ssr_t . . . . .	1461
sta_t . . . . .	1464
stec_t . . . . .	1466
stream_cfg . . . . .	1467
stream_t . . . . .	1468
StringConverter . . . . .	1470
gr::sync_block	
FifoReader . . . . .	531
Gnss_Sdr_Timestamp . . . . .	893
Gnss_Sdr_Valve . . . . .	894
Gr_Complex_Ip_Packet_Source . . . . .	1084
IONGSMSFileSource . . . . .	1120
beamformer . . . . .	364
byte_x2_to_complex_byte . . . . .	421
complex_byte_to_float_x2 . . . . .	453
complex_float_to_complex_byte . . . . .	454

conjugate_cc . . . . .	461
conjugate_ic . . . . .	463
conjugate_sc . . . . .	464
cshort_to_float_x2 . . . . .	470
cshort_to_gr_complex . . . . .	472
rtklib_pvt_gs . . . . .	1365
rtl_tcp_signal_source_c . . . . .	1380
short_x2_to_cshort . . . . .	1421
gr::sync_decimator	
gnss_sdr_sample_counter . . . . .	884
interleaved_byte_to_complex_byte . . . . .	1112
interleaved_byte_to_complex_short . . . . .	1114
interleaved_short_to_complex_short . . . . .	1115
gr::sync_interpolator	
unpack_2bit_samples . . . . .	1515
unpack_byte_2bit_cpx_samples . . . . .	1516
unpack_byte_2bit_samples . . . . .	1518
unpack_byte_4bit_samples . . . . .	1519
unpack_intspir_1bit_samples . . . . .	1521
unpack_short_byte_samples . . . . .	1522
unpack_spir_gss6450_samples . . . . .	1524
Tag . . . . .	1472
Tcp_Communication . . . . .	1475
Tcp_Packet_Data . . . . .	1475
tcp_t . . . . .	1477
tcpcli_t . . . . .	1479
TcpCmdInterface . . . . .	1480
tcpsvr_t . . . . .	1481
tec_t . . . . .	1482
tfe_t . . . . .	1485
tled_t . . . . .	1487
Tlm_Conf . . . . .	1490
Tlm_CRC_Stats . . . . .	1492
Tracking_2nd_DLL_filter . . . . .	1493
Tracking_2nd_PLL_filter . . . . .	1494
Tracking_FLL_PLL_filter . . . . .	1495
Tracking_loop_filter . . . . .	1496
TrackingCmd . . . . .	1497
trop_t . . . . .	1501
UnscentedFilter . . . . .	1526
url_t . . . . .	1527
v27_decision_t . . . . .	1528
v27_poly_t . . . . .	1528
v27_t . . . . .	1529
Viterbi_Decoder . . . . .	1530
Viterbi_Decoder_Sbas . . . . .	1532

# Chapter 7

## Class Index

### 7.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">Acq_Conf</a> . . . . .	324
<a href="#">Acq_Conf_Fpga</a> . . . . .	329
<a href="#">AcquisitionInterface</a> This abstract class represents an interface to an acquisition GNSS block . . . . .	332
<a href="#">ad936x_iio_custom</a> . . . . .	337
<a href="#">ad936x_iio_samples</a> . . . . .	338
<a href="#">ad936x_iio_source</a> This class implements conversion between Labsat 2, 3 and 3 Wideband formats to gr_complex . . . . .	339
<a href="#">Ad936xCustomSignalSource</a> This class instantiates the Ad936xCustom gnuradio signal source. It has support also for a customized Ad936xCustom firmware and signal source to support PPS samplestamp reading . . . . .	341
<a href="#">Adrv9361z7035SignalSourceFPGA</a> . . . . .	346
<a href="#">Agnss_Ref_Location</a> Interface of an Assisted GNSS REFERENCE LOCATION storage . . . . .	349
<a href="#">Agnss_Ref_Time</a> Interface of an Assisted GNSS REFERENCE TIME storage . . . . .	350
<a href="#">Allocator</a> . . . . .	352
<a href="#">alm_t</a> . . . . .	353
<a href="#">ambc_t</a> . . . . .	356
<a href="#">An_Packet_Printer</a> Class that prints PVT solutions in a serial device following a custom version of the Advanced Navigation Packet Protocol . . . . .	357
<a href="#">an_packet_t</a> . . . . .	358
<a href="#">ArraySignalConditioner</a> This class wraps blocks to change data_type_adapter, input_filter and resampler to be applied to the input flow of sampled signal . . . . .	359
<a href="#">Bayesian_estimator</a> <a href="#">Bayesian_estimator</a> is an estimator of noise characteristics (i.e. mean, covariance) . . . . .	363
<a href="#">beamformer</a> This class implements a real-time software-defined spatial filter using the CTTC GNSS experi- mental antenna array input and a set of dynamically reloadable weights . . . . .	364
<a href="#">BeamformerFilter</a> Interface of an adapter of a digital beamformer block to a <a href="#">GNSSBlockInterface</a> . . . . .	366
<a href="#">beidou_b1i_telemetry_decoder_gs</a> This class implements a block that decodes the BeiDou DNAV data . . . . .	369

<a href="#">beidou_b3i_telemetry_decoder_gs</a>	
This class implements a block that decodes the BeiDou DNAV data	371
<a href="#">Beidou_Dnav_Almanac</a>	
This class is a storage for the BeiDou D1 almanac	374
<a href="#">Beidou_Dnav_Ephemeris</a>	
This is a storage class for the BeiDou SV ephemeris data as described in BeiDou Navigation Satellite System Signal In Space Interface Control Document Open Service Signal B1I (Version 3.0)	377
<a href="#">Beidou_Dnav_Iono</a>	
This class is a storage for the BEIDOU IONOSPHERIC data as described in ICD v2.1	384
<a href="#">Beidou_Dnav_Navigation_Message</a>	
This class decodes a BeiDou D1 NAV Data message	387
<a href="#">Beidou_Dnav_Utc_Model</a>	
This class is a storage for the BeiDou DNAV UTC Model	390
<a href="#">BeidouB1iDIPIITracking</a>	
This class implements a code DLL + carrier PLL tracking loop	393
<a href="#">BeidouB1iPcpsAcquisition</a>	
This class adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for GPS L1 C/A signals	397
<a href="#">BeidouB1iTelemetryDecoder</a>	
This class implements a NAV data decoder for BEIDOU B1I	403
<a href="#">BeidouB3iDIPIITracking</a>	
This class implements a code DLL + carrier PLL tracking loop	407
<a href="#">BeidouB3iPcpsAcquisition</a>	
This class adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for BeiDou B3I signals	411
<a href="#">BeidouB3iTelemetryDecoder</a>	
This class implements a NAV data decoder for BEIDOU B1I	417
<a href="#">byte_x2_to_complex_byte</a>	
This class adapts two signed char streams into a <code>std::complex&lt;signed char&gt;</code> stream	421
<a href="#">ByteToShort</a>	
Adapts an 8-bits sample stream (IF) to a short int stream (IF)	422
<a href="#">Channel</a>	
This class represents a GNSS channel. It wraps an <a href="#">AcquisitionInterface</a> , a <a href="#">TrackingInterface</a> and a <a href="#">TelemetryDecoderInterface</a> , and handles their interaction through a Finite State Machine	426
<a href="#">Channel_Event</a>	431
<a href="#">channel_msg_receiver_cc</a>	
GNU Radio block that receives asynchronous channel messages from acquisition and tracking blocks	431
<a href="#">channel_status_msg_receiver</a>	
GNU Radio block that receives asynchronous channel messages from tlm blocks	433
<a href="#">ChannelFsm</a>	
This class implements a State Machine for channel	435
<a href="#">ChannelInterface</a>	
This abstract class represents an interface to a channel GNSS block	436
<a href="#">cl_fft_plan</a>	441
<a href="#">clFFT_Complex</a>	444
<a href="#">clFFT_Dim3</a>	445
<a href="#">clFFT_SplitComplex</a>	446
<a href="#">cnav_msg_decoder_t</a>	446
<a href="#">cnav_msg_t</a>	448
<a href="#">cnav_v27_part_t</a>	449
<a href="#">Command_Event</a>	452
<a href="#">complex_byte_to_float_x2</a>	
This class adapts a <code>std::complex&lt;signed char&gt;</code> stream into two 16-bits (short) streams	453
<a href="#">complex_float_to_complex_byte</a>	
This class adapts a <code>gr_complex</code> stream into a <code>std::complex&lt;signed char&gt;</code> stream	454
<a href="#">Concurrent_Map&lt; Data &gt;</a>	
This class implements a thread-safe <code>std::map</code>	456



<a href="#">Concurrent_Queue&lt; Data &gt;</a>	
This class implements a thread-safe std::queue . . . . .	457
<a href="#">ConfigurationInterface</a>	
This abstract class represents an interface to configuration parameters . . . . .	459
<a href="#">conjugate_cc</a>	
This class adapts a std::complex<short> stream into two 32-bits (float) streams . . . . .	461
<a href="#">conjugate_ic</a>	
This class adapts a std::complex<short> stream into two 32-bits (float) streams . . . . .	463
<a href="#">conjugate_sc</a>	
This class adapts a std::complex<short> stream into two 32-bits (float) streams . . . . .	464
<a href="#">ControlThread</a>	
This class represents the main thread of the application, so the name is <a href="#">ControlThread</a> . This is the GNSS Receiver Control Plane: it connects the flowgraph, starts running it, and while it does not stop, reads the control messages generated by the blocks, processes them, and applies the corresponding actions . . . . .	465
<a href="#">Cpu_Multicorrelator</a>	
Class that implements carrier wipe-off and correlators . . . . .	467
<a href="#">Cpu_Multicorrelator_16sc</a>	
Class that implements carrier wipe-off and correlators . . . . .	468
<a href="#">Cpu_Multicorrelator_Real_Codes</a>	
Class that implements carrier wipe-off and correlators . . . . .	469
<a href="#">cshort_to_float_x2</a>	
This class adapts a std::complex<short> stream into two 32-bits (float) streams . . . . .	470
<a href="#">cshort_to_gr_complex</a>	
This class adapts a short (16-bits) interleaved sample stream into a std::complex<float> stream . . . . .	472
<a href="#">CshortToGrComplex</a>	
Adapts an 16-bits complex sample stream to a float complex stream . . . . .	473
<a href="#">CubatureFilter</a> . . . . .	476
<a href="#">cuda_multicorrelator</a>	
Class that implements carrier wipe-off and correlators using NVIDIA CUDA GPU accelerators . . . . .	476
<a href="#">CustomUDPSignalSource</a>	
This class reads from UDP packets, which streams interleaved I/Q samples over a network . . . . .	477
<a href="#">Deleter</a> . . . . .	481
<a href="#">dgps_t</a> . . . . .	482
<a href="#">direct_resampler_conditioner_cb</a>	
This class implements a direct resampler conditioner for std::complex<signed char> . . . . .	483
<a href="#">direct_resampler_conditioner_cc</a>	
This class implements a direct resampler conditioner for complex data . . . . .	485
<a href="#">direct_resampler_conditioner_cs</a>	
This class implements a direct resampler conditioner for std::complex<short> . . . . .	486
<a href="#">DirectResamplerConditioner</a>	
Interface of an adapter of a direct resampler conditioner block to a SignalConditionerInterface . . . . .	488
<a href="#">Dll_Pll_Conf</a> . . . . .	492
<a href="#">Dll_Pll_Conf_Fpga</a> . . . . .	498
<a href="#">dll_pll_veml_tracking</a>	
This class implements a code DLL + carrier PLL tracking block . . . . .	504
<a href="#">dll_pll_veml_tracking_fpga</a>	
This class implements a code DLL + carrier PLL tracking block . . . . .	505
<a href="#">DMASignalSourceFPGA</a> . . . . .	509
<a href="#">DSM_dsm_header</a> . . . . .	512
<a href="#">DSM_KROOT_message</a> . . . . .	513
<a href="#">DSM_nma_header</a> . . . . .	516
<a href="#">DSM_PKR_message</a> . . . . .	517
<a href="#">eph_t</a> . . . . .	519
<a href="#">erp_t</a> . . . . .	524
<a href="#">erpd_t</a> . . . . .	525
<a href="#">Exponential_Smoothing</a>	
Class that implements a first-order exponential smoother . . . . .	526

<a href="#">exterr_t</a>	528
<a href="#">fcbd_t</a>	530
<a href="#">FifoReader</a>	531
<a href="#">FifoSignalSource</a>	
Class that reads a sample stream from a Unix FIFO	533
<a href="#">file_t</a>	537
<a href="#">FileConfiguration</a>	
This class is an implementation of the interface <a href="#">ConfigurationInterface</a>	540
<a href="#">FileSignalSource</a>	
Class that reads signals samples from a file and adapts it to a <a href="#">SignalSourceInterface</a>	543
<a href="#">FileSourceBase</a>	
Base class to file-oriented <a href="#">SignalSourceBase</a> GNSS blocks	547
<a href="#">FileTimestampSignalSource</a>	
Class that reads signals samples from a file and adapts it to a <a href="#">SignalSourceInterface</a>	553
<a href="#">FirFilter</a>	
This class adapts a GNU Radio <code>gr_fir_filter</code> designed with <code>pm_remez</code>	558
<a href="#">FlexibandSignalSource</a>	
This class configures and reads samples from Teleorbit Flexiband front-end. This software requires a Flexiband GNU Radio driver installed (not included with GNSS-SDR)	561
<a href="#">Fmcomms2SignalSource</a>	566
<a href="#">Fmcomms5SignalSourceFPGA</a>	569
<a href="#">FourBitCpxFileSignalSource</a>	
Class that reads signals samples from a file and adapts it to a <a href="#">SignalSourceInterface</a>	572
<a href="#">Fpga_Acquisition</a>	
Class that implements carrier wipe-off and correlators	576
<a href="#">Fpga_buffer_monitor</a>	
Class that checks the receiver buffer overflow flags and monitors the status of the receiver buffers	579
<a href="#">Fpga_DMA</a>	
Class that controls the switch DMA in the FPGA	581
<a href="#">Fpga_dynamic_bit_selection</a>	
Class that controls the switch in the FPGA, which connects the FPGA acquisition and multicorrelator modules to either the DMA or the Analog Front-End	582
<a href="#">Fpga_Multicorrelator_8sc</a>	
Class that implements carrier wipe-off and correlators	583
<a href="#">Fpga_spidev</a>	588
<a href="#">Fpga_Switch</a>	
Class that controls the switch in the FPGA, which connects the FPGA acquisition and multicorrelator modules to either the DMA or the Analog Front-End	589
<a href="#">FreqXlatingFirFilter</a>	
This class adapts a <code>gnuradio gr_freq_xlating_fir_filter</code> designed with <code>pm_remez</code>	590
<a href="#">ftp_t</a>	594
<a href="#">Galileo_Almanac</a>	
This class is a storage for the Galileo SV ALMANAC data	596
<a href="#">Galileo_Almanac_Helper</a>	
This class is a storage for the GALILEO ALMANAC data as described in GALILEO ICD	599
<a href="#">Galileo_Cnav_Message</a>	
This class handles the Galileo CNAV Data message, as described in the Galileo High Accuracy Service Signal-In-Space Interface Control Document (HAS SIS ICD) Issue 1.0, May 2022	605
<a href="#">Galileo_E1_Tcp_Connector_Tracking_cc</a>	
This class implements a code DLL + carrier PLL VEML (Very Early Minus Late) tracking block for Galileo E1 signals	607
<a href="#">galileo_e5a_noncoherentIQ_acquisition_caf_cc</a>	
This class implements a Parallel Code Phase Search Acquisition	608
<a href="#">galileo_e6_has_msg_receiver</a>	
GNU Radio block that receives asynchronous Galileo HAS message pages from the telemetry blocks, stores them in memory, and decodes HAS messages when enough data have been received. The decoded HAS message is sent to the PVT block	614

<a href="#">Galileo_Ephemeris</a>	This class is a storage and orbital model functions for the Galileo SV ephemeris data as described in Galileo ICD paragraph 5.1.1 . . . . .	616
<a href="#">Galileo_Fnav_Message</a>	This class handles the Galileo F/NAV Data message, as described in the Galileo Open Service Signal in Space Interface Control Document (OS SIS ICD), Issue 2.0 (Jan. 2021). See <a href="https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.0.pdf">https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.0.pdf</a> . . . . .	623
<a href="#">Galileo_HAS_data</a>	This class is a storage for Galileo HAS message type 1, as defined in Galileo High Accuracy Service Signal-In-Space Interface Control Document (HAS SIS ICD) Issue 1.0, May 2022. See <a href="https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_HAS_SIS_ICD_v1.0.pdf">https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_HAS_SIS_ICD_v1.0.pdf</a> . . . . .	625
<a href="#">Galileo_HAS_page</a>	This class is a storage for Galileo HAS message page, as defined in Galileo High Accuracy Service Signal-In-Space Interface Control Document (HAS SIS ICD) Issue 1.0, May 2022 . . .	635
<a href="#">Galileo_Inav_Message</a>	This class handles the Galileo I/NAV Data message, as described in the Galileo Open Service Signal in Space Interface Control Document (OS SIS ICD), Issue 2.0 (Jan. 2021). See <a href="https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.0.pdf">https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.0.pdf</a> . . . . .	638
<a href="#">Galileo_Iono</a>	This class is a storage for the GALILEO IONOSPHERIC data as described in Galileo ICD paragraph 5.1.6 . . . . .	641
<a href="#">Galileo_ISM</a>	This class is a storage for the GALILEO Integrity Support Message as described in Galileo ICD paragraph 5.2 . . . . .	644
<a href="#">galileo_pcps_8ms_acquisition_cc</a>	This class implements a Parallel Code Phase Search Acquisition for Galileo E1 signals with coherent integration time = 8 ms (two codes) . . . . .	645
<a href="#">Galileo_Reduced_CED</a>	This class manages the Galileo Reduced Clock and Ephemeris Data . . . . .	650
<a href="#">galileo_telemetry_decoder_gs</a>	This class implements a block that decodes the INAV and FNAV data defined in Galileo ICD . .	653
<a href="#">galileo_tow_map</a>		656
<a href="#">Galileo_Utc_Model</a>	This class is a storage for the GALILEO UTC MODEL data as described in Galileo ICD <a href="https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.0.pdf">https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.0.pdf</a> paragraph 5.1.7 . . . . .	657
<a href="#">GalileoE1BTelemetryDecoder</a>	This class implements a NAV data decoder for Galileo INAV frames in E1B radio link . . . . .	660
<a href="#">GalileoE1DIPIIVemlTracking</a>	This class Adapts a DLL+PLL VEML (Very Early Minus Late) tracking loop block to a <a href="#">TrackingInterface</a> for Galileo E1 signals . . . . .	664
<a href="#">GalileoE1DIPIIVemlTrackingFpga</a>	This class Adapts a DLL+PLL VEML (Very Early Minus Late) tracking loop block to a <a href="#">TrackingInterface</a> for Galileo E1 signals . . . . .	668
<a href="#">GalileoE1Pcps8msAmbiguousAcquisition</a>	Adapts a PCPS 8ms acquisition block to an <a href="#">AcquisitionInterface</a> for Galileo E1 Signals . . . .	673
<a href="#">GalileoE1PcpsAmbiguousAcquisition</a>	This class adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for Galileo E1 Signals . .	679
<a href="#">GalileoE1PcpsAmbiguousAcquisitionFpga</a>	This class adapts a PCPS acquisition block off-loaded on an FPGA to an <a href="#">AcquisitionInterface</a> for Galileo E1 Signals . . . . .	685
<a href="#">GalileoE1PcpsCccwsrAmbiguousAcquisition</a>	Adapts a PCPS CCCWSR acquisition block to an <a href="#">AcquisitionInterface</a> for Galileo E1 Signals .	692
<a href="#">GalileoE1PcpsQuickSyncAmbiguousAcquisition</a>	This class adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for Galileo E1 Signals . .	697

<a href="#">GalileoE1PcpsTongAmbiguousAcquisition</a>	Adapts a PCPS Tong acquisition block to an <a href="#">AcquisitionInterface</a> for Galileo E1 Signals . . . .	703
<a href="#">GalileoE1TcpConnectorTracking</a>	This class implements a code DLL + carrier PLL tracking loop . . . . .	709
<a href="#">GalileoE5aDIIPILTracking</a>	This class implements a code DLL + carrier PLL tracking loop . . . . .	713
<a href="#">GalileoE5aDIIPILTrackingFpga</a>	This class implements a code DLL + carrier PLL tracking loop . . . . .	717
<a href="#">GalileoE5aNoncoherentIQAcquisitionCaf</a>		723
<a href="#">GalileoE5aPcpsAcquisition</a>		729
<a href="#">GalileoE5aPcpsAcquisitionFpga</a>	This class adapts a PCPS acquisition block off-loaded on an FPGA to an <a href="#">AcquisitionInterface</a> for Galileo E5a signals . . . . .	734
<a href="#">GalileoE5aTelemetryDecoder</a>	This class implements a NAV data decoder for Galileo INAV frames in E1B radio link . . . . .	741
<a href="#">GalileoE5bDIIPILTracking</a>	This class implements a code DLL + carrier PLL tracking loop . . . . .	745
<a href="#">GalileoE5bPcpsAcquisition</a>		750
<a href="#">GalileoE5bPcpsAcquisitionFpga</a>	This class adapts a PCPS acquisition block off-loaded on an FPGA to an <a href="#">AcquisitionInterface</a> for Galileo E5b signals . . . . .	756
<a href="#">GalileoE5bTelemetryDecoder</a>	This class implements a NAV data decoder for Galileo INAV frames in E5b radio link . . . . .	763
<a href="#">GalileoE6DIIPILTracking</a>	This class implements a code DLL + carrier PLL tracking loop . . . . .	767
<a href="#">GalileoE6PcpsAcquisition</a>	This class adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for Galileo E6 Signals . .	771
<a href="#">GalileoE6TelemetryDecoder</a>	This class implements a NAV data decoder for Galileo CNAV frames in E6 radio link . . . . .	777
<a href="#">GenSignalSource</a>	This class wraps blocks that generates synthesized GNSS signal and filters the signal . . . . .	781
<a href="#">Geohash</a>	Class for geohash encoding / decoding See <a href="https://en.wikipedia.org/wiki/Geohash">https://en.wikipedia.org/wiki/Geohash</a> . . . . .	785
<a href="#">GeoJSON_Printer</a>	Prints PVT solutions in GeoJSON format file . . . . .	786
<a href="#">geph_t</a>		788
<a href="#">Glonass_Gnav_Almanac</a>	This class is a storage for the GLONASS SV ALMANAC data as described GLONASS ICD (Edition 5.1) . . . . .	790
<a href="#">Glonass_Gnav_Ephemeris</a>	This class is a storage and orbital model functions for the GLONASS SV ephemeris data as described in GLONASS ICD (Edition 5.1) . . . . .	794
<a href="#">Glonass_Gnav_Navigation_Message</a>	This class decodes a GLONASS GNAV Data message as described in GLONASS ICD (Edition 5.1) . . . . .	804
<a href="#">Glonass_Gnav_Utc_Model</a>	This class is a storage for the GLONASS GNAV UTC MODEL data as described in GLONASS ICD (Edition 5.1) . . . . .	809
<a href="#">glonass_l1_ca_dll_pll_c_aid_tracking_cc</a>	This class implements a DLL + PLL tracking loop block . . . . .	811
<a href="#">glonass_l1_ca_dll_pll_c_aid_tracking_sc</a>	This class implements a DLL + PLL tracking loop block . . . . .	813
<a href="#">Glonass_L1_Ca_Dll_Pll_Tracking_cc</a>	This class implements a DLL + PLL tracking loop block . . . . .	814
<a href="#">glonass_l1_ca_telemetry_decoder_gs</a>	This class implements a block that decodes the GNAV data defined in GLONASS ICD v5.1 . .	816

<a href="#">glonass_l2_ca_dll_pll_c_aid_tracking_cc</a>	
This class implements a DLL + PLL tracking loop block . . . . .	818
<a href="#">glonass_l2_ca_dll_pll_c_aid_tracking_sc</a>	
This class implements a DLL + PLL tracking loop block . . . . .	820
<a href="#">Glonass_L2_Ca_Dll_Pll_Tracking_cc</a>	
This class implements a DLL + PLL tracking loop block . . . . .	821
<a href="#">glonass_l2_ca_telemetry_decoder_gs</a>	
This class implements a block that decodes the GNAV data defined in GLONASS ICD v5.1 . . . . .	823
<a href="#">GlonassL1CaDIPIICAidTracking</a>	
This class implements a code DLL + carrier PLL tracking loop . . . . .	825
<a href="#">GlonassL1CaDIPIITracking</a>	
This class implements a code DLL + carrier PLL tracking loop . . . . .	829
<a href="#">GlonassL1CaPcpsAcquisition</a>	
This class adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for GPS L1 C/A signals . . . . .	833
<a href="#">GlonassL1CaTelemetryDecoder</a>	
This class implements a NAV data decoder for GLONASS L1 C/A . . . . .	839
<a href="#">GlonassL2CaDIPIICAidTracking</a>	
This class implements a code DLL + carrier PLL tracking loop . . . . .	842
<a href="#">GlonassL2CaDIPIITracking</a>	
This class implements a code DLL + carrier PLL tracking loop . . . . .	846
<a href="#">GlonassL2CaPcpsAcquisition</a>	
This class adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for GLONASS L2 C/A signals . . . . .	850
<a href="#">GlonassL2CaTelemetryDecoder</a>	
This class implements a NAV data decoder for GLONASS L2 C/A . . . . .	856
<a href="#">Gnss_Almanac</a>	
Base class for GNSS almanac storage . . . . .	859
<a href="#">Gnss_circular_deque&lt; T &gt;</a>	
. . . . .	865
<a href="#">Gnss_Crypto</a>	
Class implementing cryptographic functions for Navigation Message Authentication . . . . .	867
<a href="#">Gnss_Ephemeris</a>	
Base class for GNSS ephemeris storage . . . . .	870
<a href="#">Gnss_Satellite</a>	
This class represents a GNSS satellite . . . . .	878
<a href="#">gnss_sdr_fpga_sample_counter</a>	
. . . . .	882
<a href="#">gnss_sdr_sample_counter</a>	
. . . . .	884
<a href="#">Gnss_Sdr_Supl_Client</a>	
Class that implements a C++ interface to external Secure User Location Protocol (SUPL) client library. . . . .	885
<a href="#">gnss_sdr_time_counter</a>	
. . . . .	892
<a href="#">Gnss_Sdr_Timestamp</a>	
. . . . .	893
<a href="#">Gnss_Sdr_Valve</a>	
Implementation of a GNU Radio block that sends a STOP message to the control queue right after a specific number of samples have passed through it . . . . .	894
<a href="#">Gnss_Signal</a>	
This class represents a GNSS signal . . . . .	896
<a href="#">Gnss_Synchro</a>	
This is the class that contains the information that is shared by the processing blocks . . . . .	897
<a href="#">gnss_synchro_monitor</a>	
This class implements a monitoring block which allows sending a data stream with the receiver internal parameters ( <a href="#">Gnss_Synchro</a> objects) to local or remote clients over UDP . . . . .	905
<a href="#">Gnss_Synchro_Udp_Sink</a>	
This class sends serialized <a href="#">Gnss_Synchro</a> objects over UDP to one or multiple endpoints . . . . .	906
<a href="#">GNSSBlockFactory</a>	
Class that produces all kinds of GNSS blocks . . . . .	907
<a href="#">GNSSBlockInterface</a>	
This abstract class represents an interface to GNSS blocks . . . . .	908

<a href="#">GNSSFlowgraph</a>	
This class represents a GNSS flow graph	911
<a href="#">GnssTime</a>	915
<a href="#">Gps_Acq_Assist</a>	
This class is a storage for the GPS GSM RRLT acquisition assistance data as described in Digital cellular telecommunications system (Phase 2+); Location Services (LCS); Mobile Station (MS) - Serving Mobile Location Centre (SMLC) Radio Resource LCS Protocol (RRLP) (3GPP TS 44.031 version 5.12.0 Release 5)	916
<a href="#">Gps_Almanac</a>	
This class is a storage for the GPS SV ALMANAC data as described in IS-GPS-200M	918
<a href="#">Gps_CNAV_Ephemeris</a>	
This is a storage class for the GPS CNAV ephemeris data as described in IS-GPS-200M	922
<a href="#">Gps_CNAV_Iono</a>	
This class is a storage for the GPS IONOSPHERIC data as described in IS-GPS-200M	930
<a href="#">Gps_CNAV_Navigation_Message</a>	
This class decodes a GPS CNAV Data message as described in IS-GPS-200M	932
<a href="#">Gps_CNAV_Utc_Model</a>	
This class is a storage for the GPS UTC MODEL data as described in in IS-GPS-200M	934
<a href="#">Gps_Ephemeris</a>	
This class is a storage and orbital model functions for the GPS SV ephemeris data as described in IS-GPS-200M	936
<a href="#">Gps_Iono</a>	
This class is a storage for the GPS IONOSPHERIC data as described in IS-GPS-200M	944
<a href="#">Gps_L1_Ca_Dll_Pll_Tracking_GPU_cc</a>	
This class implements a DLL + PLL tracking loop block	947
<a href="#">Gps_L1_Ca_Gaussian_Tracking_cc</a>	
This class implements a DLL + PLL tracking loop block	949
<a href="#">Gps_L1_Ca_Tcp_Connector_Tracking_cc</a>	
This class implements a DLL + PLL tracking loop block	950
<a href="#">gps_l1_ca_telemetry_decoder_gs</a>	
This class implements a block that decodes the NAV data defined in IS-GPS-200M	952
<a href="#">gps_l2c_telemetry_decoder_gs</a>	
This class implements a block that decodes CNAV data defined in IS-GPS-200M	954
<a href="#">gps_l5_telemetry_decoder_gs</a>	
This class implements a GPS L5 Telemetry decoder	956
<a href="#">Gps_Navigation_Message</a>	
This class decodes a GPS NAV Data message as described in IS-GPS-200M	958
<a href="#">Gps_Utc_Model</a>	
This class is a storage for the GPS UTC MODEL data as described in IS-GPS-200M	960
<a href="#">GpsL1CaDllPllTracking</a>	
This class implements a code DLL + carrier PLL tracking loop	964
<a href="#">GpsL1CaDllPllTrackingFpga</a>	
This class implements a code DLL + carrier PLL tracking loop	968
<a href="#">GpsL1CaDllPllTrackingGPU</a>	
This class implements a code DLL + carrier PLL tracking loop using GPU accelerated functions	973
<a href="#">GpsL1CaGaussianTracking</a>	
This class implements a code DLL + carrier PLL tracking loop	977
<a href="#">GpsL1CaKfTracking</a>	
This class implements a code + carrier Kalman Filter tracking loop with VTL capabilities	981
<a href="#">GpsL1CaPcpsAcquisition</a>	
This class adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for GPS L1 C/A signals	985
<a href="#">GpsL1CaPcpsAcquisitionFineDoppler</a>	
This class Adapts a PCPS acquisition block with fine Doppler estimation to an <a href="#">AcquisitionInterface</a> for GPS L1 C/A signals	991
<a href="#">GpsL1CaPcpsAcquisitionFpga</a>	
This class adapts a PCPS acquisition block off-loaded on an FPGA to an <a href="#">AcquisitionInterface</a> for GPS L1 C/A signals	997

<a href="#">GpsL1CaPcpsAssistedAcquisition</a>	
This class adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for GPS L1 C/A signals	1003
<a href="#">GpsL1CaPcpsOpenCLAcquisition</a>	
This class adapts an OpenCL PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for GPS L1 C/A signals	1009
<a href="#">GpsL1CaPcpsQuickSyncAcquisition</a>	
This class adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for GPS L1 C/A signals	1014
<a href="#">GpsL1CaPcpsTongAcquisition</a>	
This class adapts a PCPS Tong acquisition block to an <a href="#">AcquisitionInterface</a> for GPS L1 C/A signals	1020
<a href="#">GpsL1CaTcpConnectorTracking</a>	
This class implements a code DLL + carrier PLL tracking loop	1025
<a href="#">GpsL1CaTelemetryDecoder</a>	
This class implements a NAV data decoder for GPS L1 C/A	1028
<a href="#">GpsL2CTelemetryDecoder</a>	
This class implements a NAV data decoder for GPS L2 M	1032
<a href="#">GpsL2MDIIPITracking</a>	
This class implements a code DLL + carrier PLL tracking loop	1036
<a href="#">GpsL2MDIIPITrackingFpga</a>	
This class implements a code DLL + carrier PLL tracking loop	1040
<a href="#">GpsL2MPcpsAcquisition</a>	
This class adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for GPS L2 M signals	1044
<a href="#">GpsL2MPcpsAcquisitionFpga</a>	
This class adapts a PCPS acquisition block off-loaded on an FPGA to an <a href="#">AcquisitionInterface</a> for GPS L2 M signals	1050
<a href="#">GpsL5DIIPITracking</a>	
This class implements a code DLL + carrier PLL tracking loop	1056
<a href="#">GpsL5DIIPITrackingFpga</a>	
This class implements a code DLL + carrier PLL tracking loop	1059
<a href="#">GpsL5iPcpsAcquisition</a>	
This class adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for GPS L5i signals	1064
<a href="#">GpsL5iPcpsAcquisitionFpga</a>	
This class adapts a PCPS acquisition block off-loaded on an FPGA to an <a href="#">AcquisitionInterface</a> for GPS L5i signals	1070
<a href="#">GpsL5TelemetryDecoder</a>	
This class implements a NAV data decoder for GPS L5	1077
<a href="#">GPU_Complex</a>	1080
<a href="#">GPU_Complex_Short</a>	1082
<a href="#">Gpx_Printer</a>	
Prints PVT information to GPX format file	1083
<a href="#">Gr_Complex_Ip_Packet_Source</a>	1084
<a href="#">gtime_t</a>	1086
<a href="#">half_cyc_tag</a>	1087
<a href="#">HAS_clock_corrections</a>	1088
<a href="#">HAS_obs_corrections</a>	1089
<a href="#">HAS_orbit_corrections</a>	1090
<a href="#">Has_Simple_Printer</a>	
Prints HAS messages content in a txt file. See HAS-SIS-ICD for a message description	1091
<a href="#">hybrid_observables_gs</a>	
This class implements a block that computes observables	1091
<a href="#">HybridObservables</a>	
This class implements an <a href="#">ObservablesInterface</a> for observables of all kind of GNSS signals	1093
<a href="#">lbyteToCbyte</a>	1097
<a href="#">lbyteToComplex</a>	
Adapts an I/Q interleaved byte integer sample stream to a gr_complex (float) stream	1101
<a href="#">lbyteToCshort</a>	
Adapts a short integer (16 bits) interleaved sample stream into a std::complex<short> stream	1104



<a href="#">INIReader</a>	
Read an INI file into easy-to-access name/value pairs. (Note that I've gone for simplicity here rather than speed, but it should be pretty decent.)	1107
<a href="#">InMemoryConfiguration</a>	
This class is an implementation of the interface <a href="#">ConfigurationInterface</a>	1108
<a href="#">interleaved_byte_to_complex_byte</a>	
This class adapts an 8-bits interleaved sample stream into a 16-bits complex stream (std::complex<unsigned char>)	1112
<a href="#">interleaved_byte_to_complex_short</a>	
This class adapts a short (16-bits) interleaved sample stream into a std::complex<short> stream	1114
<a href="#">interleaved_short_to_complex_short</a>	
This class adapts a short (16-bits) interleaved sample stream into a std::complex<short> stream	1115
<a href="#">IONGSMSChunkData</a>	1117
<a href="#">IONGSMSChunkUnpackingCtx&lt; WT &gt;</a>	1118
<a href="#">IONGSMSFileSource</a>	1120
<a href="#">IONGSMSSignalSource</a>	
Class that reads signals samples from a file and adapts it to a <a href="#">SignalSourceInterface</a>	1122
<a href="#">IshortToComplex</a>	
Adapts an I/Q interleaved short integer sample stream to a gr_complex (float) stream	1125
<a href="#">IshortToCshort</a>	
Adapts a short integer (16 bits) interleaved sample stream into a std::complex<short> stream	1129
<a href="#">kernel_info_t</a>	1132
<a href="#">Kf_Conf</a>	1134
<a href="#">kf_tracking</a>	
This class implements a code DLL + carrier PLL tracking block	1139
<a href="#">Kml_Printer</a>	
Prints PVT information to OGC KML format file (can be viewed with Google Earth)	1140
<a href="#">labsat23_source</a>	
This class implements conversion between Labsat 2, 3 and 3 Wideband formats to gr_complex	1141
<a href="#">LabsatSignalSource</a>	
This class reads samples stored in LabSat version 2, 3, and 3 Wideband format	1143
<a href="#">lex_t</a>	1146
<a href="#">lexeph_t</a>	1148
<a href="#">lexion_t</a>	1150
<a href="#">lexmsg_t</a>	1151
<a href="#">LimesdrSignalSource</a>	
This class instantiates the LimeSDR gnuradio signal source. It has support also for a customized LimeSDR firmware and signal source to support PPS samplestamp reading	1152
<a href="#">MACK_header</a>	1156
<a href="#">Mack_lookup</a>	1157
<a href="#">MACK_message</a>	1159
<a href="#">MACK_tag_and_info</a>	1161
<a href="#">MACK_tag_info</a>	1162
<a href="#">MAX2771EVKITSignalSourceFPGA</a>	1163
<a href="#">MmseResamplerConditioner</a>	
Interface of a MMSE resampler block adapter to a <a href="#">SignalConditionerInterface</a>	1166
<a href="#">ModelFunction</a>	1169
<a href="#">Monitor_Ephemeris_Udp_Sink</a>	1169
<a href="#">Monitor_Pvt</a>	
This class contains parameters and outputs of the PVT block	1170
<a href="#">Monitor_Pvt_Udp_Sink</a>	1176
<a href="#">msm_h_t</a>	1177
<a href="#">mt1_header</a>	1179
<a href="#">MultichannelFileSignalSource</a>	
Class that reads signals samples from files at different frequency bands and adapts it to a <a href="#">SignalSourceInterface</a>	1180



<a href="#">nav_message_monitor</a>	
GNU Radio block that receives asynchronous <a href="#">Nav_Message_Packet</a> objects from the telemetry blocks and sends them via UDP	1184
<a href="#">Nav_Message_Packet</a>	1186
<a href="#">Nav_Message_Udp_Sink</a>	1189
<a href="#">nav_t</a>	1190
<a href="#">Nmea_Printer</a>	
This class provides a implementation of a subset of the NMEA-0183 standard for interfacing marine electronic devices as defined by the National Marine Electronics Association (NMEA)	1196
<a href="#">Notch</a>	
This class implements a real-time software-defined multi state notch filter	1197
<a href="#">NotchFilter</a>	1199
<a href="#">NotchFilterLite</a>	1202
<a href="#">NotchLite</a>	
This class implements a real-time software-defined multi state notch filter light version	1205
<a href="#">NsrFileSignalSource</a>	
Class that reads signals samples from a file and adapts it to a <a href="#">SignalSourceInterface</a>	1206
<a href="#">ntrip_t</a>	1212
<a href="#">Obs_Conf</a>	1214
<a href="#">obs_t</a>	1216
<a href="#">obsd_t</a>	1217
<a href="#">ObservablesInterface</a>	
This abstract class represents an interface to an observables block	1219
<a href="#">opt_t</a>	1222
<a href="#">OsmosdrSignalSource</a>	
This class reads samples OsmoSDR-compatible front-ends, such as HackRF or Realtek's RTL2832U-based USB dongle DVB-T receivers (see <a href="https://osmocom.org/projects/rtl-sdr/wiki">https://osmocom.org/projects/rtl-sdr/wiki</a> )	1223
<a href="#">OSNMA_data</a>	
This class handles OSNMA data See <a href="https://www.gsc-europa.eu/sites/default/files/sites/all/Galileo_OSNMA_User_ICD_for_Test_Phase_v1.0.pdf">https://www.gsc-europa.eu/sites/default/files/sites/all/Galileo_OSNMA_User_ICD_for_Test_Phase_v1.0.pdf</a>	1226
<a href="#">OSNMA_DSM_Reader</a>	1228
<a href="#">Osnma_Helper</a>	1229
<a href="#">OSNMA_msg</a>	
This class fills the <a href="#">OSNMA_msg</a> structure with the data received from the telemetry blocks	1230
<a href="#">osnma_msg_receiver</a>	
GNU Radio block that receives asynchronous OSNMA messages from the telemetry blocks, stores them in memory, and decodes OSNMA info when enough data have been received. The decoded OSNMA data is sent to the PVT block	1231
<a href="#">OSNMA_NavData</a>	1234
<a href="#">OSNMA_NavDataManager</a>	
Class for managing OSNMA navigation data	1236
<a href="#">Pass_Through</a>	
This class implements a block that connects input and output (does nothing)	1237
<a href="#">pclk_t</a>	1241
<a href="#">pcps_acquisition</a>	
This class implements a Parallel Code Phase Search Acquisition	1242
<a href="#">pcps_acquisition_fine_doppler_cc</a>	
This class implements a Parallel Code Phase Search Acquisition	1246
<a href="#">pcps_acquisition_fpga</a>	
This class implements a Parallel Code Phase Search Acquisition that uses the FPGA	1252
<a href="#">pcps_assisted_acquisition_cc</a>	
This class implements a Parallel Code Phase Search Acquisition	1256
<a href="#">pcps_cccwsr_acquisition_cc</a>	
This class implements a Parallel Code Phase Search Acquisition with Coherent Channel Combining With Sign Recovery scheme	1261
<a href="#">pcps_opencl_acquisition_cc</a>	
This class implements a Parallel Code Phase Search Acquisition	1266

<a href="#">pcps_quicksync_acquisition_cc</a>	
This class implements a Parallel Code Phase Search Acquisition with the implementation of the Sparse QuickSync Algorithm . . . . .	1272
<a href="#">pcps_tong_acquisition_cc</a>	
This class implements a Parallel Code Phase Search Acquisition with Tong algorithm . . . . .	1278
<a href="#">pcv_t</a> . . . . .	1284
<a href="#">pcvs_t</a> . . . . .	1286
<a href="#">peph_t</a> . . . . .	1287
<a href="#">PlutosdrSignalSource</a> . . . . .	1289
<a href="#">pppcorr_t</a> . . . . .	1292
<a href="#">pps_tcp_rx</a> . . . . .	1294
<a href="#">PpsSamplestamp</a> . . . . .	1295
<a href="#">prcopt_t</a> . . . . .	1296
<a href="#">pulse_blanking_cc</a> . . . . .	1303
<a href="#">PulseBlankingFilter</a> . . . . .	1304
<a href="#">Pvt_Conf</a> . . . . .	1307
<a href="#">Pvt_Kf</a>	
Kalman Filter for Position and Velocity . . . . .	1314
<a href="#">Pvt_Solution</a>	
Base class for a PVT solution . . . . .	1315
<a href="#">PvtInterface</a>	
This class represents an interface to a PVT block . . . . .	1320
<a href="#">raw_t</a> . . . . .	1323
<a href="#">RawArraySignalSource</a>	
This class reads samples from an antenna array RF front-end signal sampler . . . . .	1327
<a href="#">ReedSolomon</a>	
Class implementing a Reed-Solomon encoder and decoder RS(255,K,d) where k=255-nroots is the information vector length and d=nroots+1 is the minimum Hamming distance, with symbols of 8 bits. It allows shortened RS codes . . . . .	1331
<a href="#">Rinex_Printer</a>	
Class that handles the generation of Receiver INdependent EXchange format (RINEX) files . . . . .	1333
<a href="#">Rtcm</a>	
This class implements the generation and reading of some Message Types defined in the RTCM 3.2 Standard, plus some utilities to handle messages . . . . .	1337
<a href="#">Rtcm_Printer</a>	
This class provides a implementation of a subset of the RTCM Standard 10403.2 messages . . . . .	1350
<a href="#">rtcm_t</a> . . . . .	1353
<a href="#">rtk_t</a> . . . . .	1357
<a href="#">Rtklib_Pvt</a>	
This class implements a <a href="#">PvtInterface</a> for the RTKLIB PVT block . . . . .	1359
<a href="#">rtklib_pvt_gs</a>	
This class implements a block that computes the PVT solution using the RTKLIB integrated library . . . . .	1365
<a href="#">Rtklib_Solver</a>	
This class implements a PVT solution based on RTKLIB . . . . .	1368
<a href="#">rtksvr_t</a> . . . . .	1375
<a href="#">Rtl_Tcp_Dongle_Info</a>	
This class represents the dongle information which is sent by rtl_tcp . . . . .	1379
<a href="#">rtl_tcp_signal_source_c</a>	
This class reads interleaved I/Q samples from an rtl_tcp server and outputs complex types . . . . .	1380
<a href="#">RtlTcpSignalSource</a>	
This class reads from rtl_tcp, which streams interleaved I/Q samples over TCP. (see <a href="https://osmocom.org/projects/rtl-sdr/wiki">https://osmocom.org/projects/rtl-sdr/wiki</a> ) . . . . .	1382
<a href="#">Sbas_Ephemeris</a>	
This class stores SBAS SV ephemeris data . . . . .	1386
<a href="#">sbas_l1_telemetry_decoder_gs</a>	
This class implements a block that decodes the SBAS integrity and corrections data defined in RTCA MOPS DO-229 . . . . .	1388

<a href="#">SbasL1TelemetryDecoder</a>	
This class implements a NAV data decoder for SBAS frames in L1 radio link	1390
<a href="#">sbs_t</a>	1394
<a href="#">sbsfcrr_t</a>	1396
<a href="#">sbsigp_t</a>	1397
<a href="#">sbsigpband_t</a>	1399
<a href="#">sbsion_t</a>	1400
<a href="#">sbslcorr_t</a>	1401
<a href="#">sbsmsg_t</a>	1402
<a href="#">sbssat_t</a>	1404
<a href="#">sbssatp_t</a>	1406
<a href="#">sdr_gnss_packet_t</a>	1407
<a href="#">seph_t</a>	1409
<a href="#">Serdes_Galileo_Eph</a>	
This class implements serialization and deserialization of <a href="#">Galileo_Ephemeris</a> using Protocol Buffers	1411
<a href="#">Serdes_Gnss_Synchro</a>	
This class implements serialization and deserialization of <a href="#">Gnss_Synchro</a> objects using Protocol Buffers	1413
<a href="#">Serdes_Gps_Eph</a>	
This class implements serialization and deserialization of <a href="#">Gps_Ephemeris</a> objects using Protocol Buffers	1415
<a href="#">Serdes_Monitor_Pvt</a>	
This class implements serialization and deserialization of <a href="#">Monitor_Pvt</a> objects using Protocol Buffers	1417
<a href="#">Serdes_Nav_Message</a>	
This class implements serialization and deserialization of <a href="#">Nav_Message_Packet</a> objects using Protocol Buffers	1419
<a href="#">serial_t</a>	1421
<a href="#">short_x2_to_cshort</a>	
This class adapts two short streams into a <code>std::complex&lt;short&gt;</code> stream	1421
<a href="#">signal_generator_c</a>	
This class generates synthesized GNSS signal	1423
<a href="#">SignalConditioner</a>	
This class wraps blocks to change <code>data_type_adapter</code> , <code>input_filter</code> and <code>resampler</code> to be applied to the input flow of sampled signal	1424
<a href="#">SignalGenerator</a>	
This class generates synthesized GNSS signal	1428
<a href="#">SignalSourceBase</a>	1432
<a href="#">SignalSourceInterface</a>	
This abstract class represents an interface to <code>signal_source</code> GNSS block	1435
<a href="#">snrmask_t</a>	1437
<a href="#">sol_t</a>	1438
<a href="#">solbuf_t</a>	1440
<a href="#">solopt_t</a>	1442
<a href="#">solstat_t</a>	1445
<a href="#">solstatbuf_t</a>	1447
<a href="#">SpirFileSignalSource</a>	
Class that reads signals samples from a file and adapts it to a <a href="#">SignalSourceInterface</a>	1448
<a href="#">SpirGSS6450FileSignalSource</a>	
Class that reads signals samples from a file and adapts it to a <a href="#">SignalSourceInterface</a>	1453
<a href="#">ssat_t</a>	1458
<a href="#">ssr_t</a>	1461
<a href="#">sta_t</a>	1464
<a href="#">stec_t</a>	1466
<a href="#">stream_cfg</a>	1467
<a href="#">stream_t</a>	1468

<a href="#">StringConverter</a>	
Class that interprets the contents of a string and converts it into different types	1470
<a href="#">Tag</a>	1472
<a href="#">Tcp_Communication</a>	
TCP communication class	1475
<a href="#">Tcp_Packet_Data</a>	
Class that implements a TCP data packet	1475
<a href="#">tcp_t</a>	1477
<a href="#">tcpcli_t</a>	1479
<a href="#">TcpCmdInterface</a>	1480
<a href="#">tcpsvr_t</a>	1481
<a href="#">tec_t</a>	1482
<a href="#">TelemetryDecoderInterface</a>	
This abstract class represents an interface to a navigation GNSS block	1483
<a href="#">tle_t</a>	1485
<a href="#">tled_t</a>	1487
<a href="#">Tlm_Conf</a>	1490
<a href="#">Tlm_CRC_Stats</a>	
Class that computes the telemetry CRC statistics	1492
<a href="#">Tracking_2nd_DLL_filter</a>	
This class implements a 2nd order DLL filter for code tracking loop	1493
<a href="#">Tracking_2nd_PLL_filter</a>	
This class implements a 2nd order PLL filter for carrier tracking loop	1494
<a href="#">Tracking_FLL_PLL_filter</a>	
This class implements a hybrid FLL and PLL filter for tracking carrier loop	1495
<a href="#">Tracking_loop_filter</a>	
This class implements a generic 1st, 2nd or 3rd order loop filter	1496
<a href="#">TrackingCmd</a>	1497
<a href="#">TrackingInterface</a>	
This abstract class represents an interface to a tracking block	1498
<a href="#">trop_t</a>	1501
<a href="#">TwoBitCpxFileSignalSource</a>	
Class that reads signals samples from a file and adapts it to a <a href="#">SignalSourceInterface</a>	1502
<a href="#">TwoBitPackedFileSignalSource</a>	
Class that reads signals samples from a file and adapts it to a <a href="#">SignalSourceInterface</a>	1506
<a href="#">UhdSignalSource</a>	
This class reads samples from a UHD device (see <a href="http://code.ettus.com/redmine/ettus/projects/uhd/wiki">http://code.ettus.com/redmine/ettus/projects/uhd/wiki</a> )	1511
<a href="#">unpack_2bit_samples</a>	
This class takes 2 bit samples that have been packed into bytes or shorts as input and generates a byte for each sample. It generates eight times as much data as is input (every two bits become 16 bits)	1515
<a href="#">unpack_byte_2bit_cpx_samples</a>	
This class implements conversion between byte packet samples to 2bit_cpx samples 1 byte = 2 x complex 2bit I, + 2bit Q samples	1516
<a href="#">unpack_byte_2bit_samples</a>	
This class implements conversion between byte packet samples to 2bit samples 1 byte = 4 2bit samples	1518
<a href="#">unpack_byte_4bit_samples</a>	
This class implements conversion between byte packet samples to 4bit_cpx samples 1 byte = 1 x complex 4bit I, + 4bit Q samples	1519
<a href="#">unpack_intspir_1bit_samples</a>	
This class implements conversion between byte packet samples to 2bit samples 1 byte = 4 2bit samples	1521
<a href="#">unpack_short_byte_samples</a>	
This class implements conversion between short packet samples to byte samples 1 short = 2 byte samples	1522
<a href="#">unpack_spir_gss6450_samples</a>	1524

<a href="#">UnscentedFilter</a>	1526
<a href="#">url_t</a>	1527
<a href="#">v27_decision_t</a>	1528
<a href="#">v27_poly_t</a>	1528
<a href="#">v27_t</a>	1529
<a href="#">Viterbi_Decoder</a>	
Class that implements a Viterbi decoder	1530
<a href="#">Viterbi_Decoder_Sbas</a>	
Class that implements a Viterbi decoder	1532
<a href="#">ZmqSignalSource</a>	
This class supports the following properties:	1533



# Chapter 8

## File Index

### 8.1 File List

Here is a list of all documented files with brief descriptions:

<a href="#">beidou_b1i_pcps_acquisition.h</a>	Adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for Beidou B1I signals . . . . .	1539
<a href="#">beidou_b3i_pcps_acquisition.h</a>	Adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for Beidou B3I signals . . . . .	1542
<a href="#">galileo_e1_pcps_8ms_ambiguous_acquisition.h</a>	Adapts a PCPS 8ms acquisition block to an <a href="#">AcquisitionInterface</a> for Galileo E1 Signals . . . . .	1544
<a href="#">galileo_e1_pcps_ambiguous_acquisition.h</a>	Adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for Galileo E1 Signals . . . . .	1547
<a href="#">galileo_e1_pcps_ambiguous_acquisition_fpga.h</a>	Adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for Galileo E1 Signals for the FPGA . . . . .	1550
<a href="#">galileo_e1_pcps_cccwsr_ambiguous_acquisition.h</a>	Adapts a PCPS CCCWSR acquisition block to an <a href="#">AcquisitionInterface</a> for Galileo E1 Signals . . . . .	1554
<a href="#">galileo_e1_pcps_quicksync_ambiguous_acquisition.h</a>	Adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for Galileo E1 Signals . . . . .	1556
<a href="#">galileo_e1_pcps_tong_ambiguous_acquisition.h</a>	Adapts a PCPS Tong acquisition block to an <a href="#">AcquisitionInterface</a> for Galileo E1 Signals . . . . .	1559
<a href="#">galileo_e5a_noncoherent_iq_acquisition_caf.h</a>	Adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for Galileo E5a data and pilot Signals . . . . .	1562
<a href="#">galileo_e5a_pcps_acquisition.h</a>	Adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for Galileo E5a data and pilot Signals . . . . .	1565
<a href="#">galileo_e5a_pcps_acquisition_fpga.h</a>	Adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for Galileo E5a data and pilot Signals for the FPGA . . . . .	1568
<a href="#">galileo_e5b_pcps_acquisition.h</a>	Adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for Galileo E5b data and pilot Signals . . . . .	1571
<a href="#">galileo_e5b_pcps_acquisition_fpga.h</a>	Adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for Galileo E5b data and pilot Signals for the FPGA . . . . .	1574
<a href="#">galileo_e6_pcps_acquisition.h</a>	Adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for Galileo E6 B/C Signals . . . . .	1578
<a href="#">glonass_l1_ca_pcps_acquisition.h</a>	Adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for Glonass L1 C/A signals . . . . .	1581
<a href="#">glonass_l2_ca_pcps_acquisition.h</a>	Adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for Glonass L2 C/A signals . . . . .	1583
<a href="#">gps_l1_ca_pcps_acquisition.h</a>	Adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for GPS L1 C/A signals . . . . .	1586

<a href="#">gps_l1_ca_pcps_acquisition_fine_doppler.h</a>	Adapts a PCPS acquisition block with fine Doppler estimation to an <a href="#">AcquisitionInterface</a> for GPS L1 C/A signals . . . . .	1589
<a href="#">gps_l1_ca_pcps_acquisition_fpga.h</a>	Adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for GPS L1 C/A signals for the FPGA	1592
<a href="#">gps_l1_ca_pcps_assisted_acquisition.h</a>	Adapts a PCPS Assisted acquisition block to an <a href="#">AcquisitionInterface</a> for GPS L1 C/A signals . . . . .	1595
<a href="#">gps_l1_ca_pcps_opencl_acquisition.h</a>	Adapts an OpenCL PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for GPS L1 C/A signals . . . . .	1598
<a href="#">gps_l1_ca_pcps_quicksync_acquisition.h</a>	Adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for GPS L1 C/A signals implementing the QuickSync Algorithm . . . . .	1601
<a href="#">gps_l1_ca_pcps_tong_acquisition.h</a>	Adapts a PCPS Tong acquisition block to an <a href="#">AcquisitionInterface</a> for GPS L1 C/A signals . . . . .	1604
<a href="#">gps_l2_m_pcps_acquisition.h</a>	Adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for GPS L2 M signals . . . . .	1607
<a href="#">gps_l2_m_pcps_acquisition_fpga.h</a>	Adapts an FPGA-offloaded PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for GPS L2 M signals . . . . .	1610
<a href="#">gps_l5i_pcps_acquisition.h</a>	Adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for GPS L5i signals . . . . .	1613
<a href="#">gps_l5i_pcps_acquisition_fpga.h</a>	Adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for GPS L5i signals for the FPGA . . . . .	1615
<a href="#">galileo_e5a_noncoherent_iq_acquisition_caf_cc.h</a>	Adapts a PCPS acquisition block to an <a href="#">AcquisitionInterface</a> for Galileo E5a data and pilot Signals	1619
<a href="#">galileo_pcps_8ms_acquisition_cc.h</a>	This class implements a Parallel Code Phase Search Acquisition for Galileo E1 signals with coherent integration time = 8 ms (two codes) . . . . .	1623
<a href="#">pcps_acquisition.h</a>	This class implements a Parallel Code Phase Search Acquisition . . . . .	1627
<a href="#">pcps_acquisition_fine_doppler_cc.h</a>	This class implements a Parallel Code Phase Search Acquisition with multi-dwells and fine Doppler estimation for GPS L1 C/A signal . . . . .	1632
<a href="#">pcps_acquisition_fpga.h</a>	This class implements a Parallel Code Phase Search Acquisition for the FPGA . . . . .	1637
<a href="#">pcps_assisted_acquisition_cc.h</a>	This class implements a Parallel Code Phase Search Acquisition with assistance and multi-dwells . . . . .	1640
<a href="#">pcps_cccwsr_acquisition_cc.h</a>	This class implements a Parallel Code Phase Search acquisition with Coherent <a href="#">Channel</a> Combining With Sign Recovery scheme . . . . .	1645
<a href="#">pcps_opencl_acquisition_cc.h</a>	This class implements a Parallel Code Phase Search Acquisition using OpenCL to offload some functions to the GPU . . . . .	1649
<a href="#">pcps_quicksync_acquisition_cc.h</a>	This class implements a Parallel Code Phase Search Acquisition with the QuickSync Algorithm . . . . .	1654
<a href="#">pcps_tong_acquisition_cc.h</a>	This class implements a Parallel Code Phase Search Acquisition with Tong algorithm . . . . .	1658
<a href="#">acq_conf.h</a>	Class that contains all the configuration parameters for generic acquisition block based on the PCPS algorithm . . . . .	1663
<a href="#">acq_conf_fpga.h</a>	Class that contains all the configuration parameters for generic acquisition block based on the PCPS algorithm running in the FPGA . . . . .	1665
<a href="#">fpga_acquisition.h</a>	Highly optimized FPGA vector correlator class . . . . .	1667
<a href="#">channel.h</a>	Interface of a GNSS channel . . . . .	1671



<a href="#">channel_fsm.h</a>	Interface of the State Machine for channel . . . . .	1673
<a href="#">channel_msg_receiver_cc.h</a>	GNU Radio block that receives asynchronous channel messages from acquisition and tracking blocks . . . . .	1675
<a href="#">array_signal_conditioner.h</a>	It wraps blocks to change data type, filter and resample input data, adapted to array receiver . . . . .	1677
<a href="#">signal_conditioner.h</a>	It wraps blocks to change data type, filter and resample input data . . . . .	1678
<a href="#">byte_to_short.h</a>	Adapts an 8-bits sample stream (IF) to a short int stream (IF) . . . . .	1680
<a href="#">cshort_to_grcomplex.h</a>	Adapts an 16-bits complex sample stream to a float complex stream . . . . .	1682
<a href="#">ibyte_to_cbyte.h</a>	Adapts an I/Q interleaved byte (unsigned char) sample stream into a std::complex<unsigned char> stream . . . . .	1683
<a href="#">ibyte_to_complex.h</a>	Adapts an I/Q interleaved byte integer sample stream to a gr_complex (float) stream . . . . .	1685
<a href="#">ibyte_to_cshort.h</a>	Adapts a short interleaved sample stream into a std::complex<short> stream . . . . .	1687
<a href="#">ishort_to_complex.h</a>	Adapts an I/Q interleaved short integer sample stream to a gr_complex (float) stream . . . . .	1689
<a href="#">ishort_to_cshort.h</a>	Adapts a short interleaved sample stream into a std::complex<short> stream . . . . .	1691
<a href="#">cshort_to_gr_complex.h</a>	Adapts a complex short (16 + 16 bits) sample stream into a std::complex<float> stream (32 + 32 bits) . . . . .	1692
<a href="#">interleaved_byte_to_complex_byte.h</a>	Adapts an 8-bits interleaved sample stream into a 16-bits complex stream . . . . .	1694
<a href="#">interleaved_byte_to_complex_short.h</a>	Adapts a byte (8-bits) interleaved sample stream into a std::complex<short> stream . . . . .	1696
<a href="#">interleaved_short_to_complex_short.h</a>	Adapts a short (16-bits) interleaved sample stream into a std::complex<short> stream . . . . .	1698
<a href="#">beamformer_filter.h</a>	Interface of an adapter of a digital beamformer . . . . .	1700
<a href="#">fir_filter.h</a>	Adapts a gnuradio gr_fir_filter designed with pm_remez . . . . .	1702
<a href="#">freq_xlating_fir_filter.h</a>	Adapts a gnuradio gr_freq_xlating_fir_filter designed with gr_remez . . . . .	1704
<a href="#">notch_filter.h</a>	Adapter of a multistate <a href="#">Notch</a> filter . . . . .	1706
<a href="#">notch_filter_lite.h</a>	Adapts a light version of a multistate notch filter . . . . .	1708
<a href="#">pulse_blanking_filter.h</a>	Instantiates the GNSS-SDR pulse blanking filter . . . . .	1709
<a href="#">beamformer.h</a>	Simple spatial filter using RAW array input and beamforming coefficients . . . . .	1711
<a href="#">notch_cc.h</a>	Implements a notch filter algorithm . . . . .	1712
<a href="#">notch_lite_cc.h</a>	Implements a notch filter light algorithm . . . . .	1715
<a href="#">pulse_blanking_cc.h</a>	Implements a pulse blanking algorithm . . . . .	1717
<a href="#">beidou_b1i_signal_replica.h</a>	This file implements various functions for BeiDou B1I signal replica generation . . . . .	1719
<a href="#">beidou_b3i_signal_replica.h</a>	This file implements various functions for BeiDou B3I signal replica generation . . . . .	1720

<a href="#">byte_x2_to_complex_byte.h</a>	Adapts two signed char streams into a <code>std::complex&lt;signed char&gt;</code> stream . . . . .	1722
<a href="#">complex_byte_to_float_x2.h</a>	Adapts a <code>std::complex&lt;signed char&gt;</code> stream into two 16-bits (short) streams . . . . .	1724
<a href="#">complex_float_to_complex_byte.h</a>	Adapts a <code>gr_complex</code> stream into a <code>std::complex&lt;signed char&gt;</code> stream . . . . .	1725
<a href="#">conjugate_cc.h</a>	Conjugate a stream of <code>gr_complex</code> . . . . .	1727
<a href="#">conjugate_ic.h</a>	Conjugate a stream of <code>lv_8sc_t ( std::complex&lt;char&gt; )</code> . . . . .	1729
<a href="#">conjugate_sc.h</a>	Conjugate a stream of <code>lv_16sc_t ( std::complex&lt;short&gt; )</code> . . . . .	1730
<a href="#">cshort_to_float_x2.h</a>	Adapts a <code>std::complex&lt;short&gt;</code> stream into two float streams . . . . .	1732
<a href="#">galileo_e1_signal_replica.h</a>	This library implements various functions for Galileo E1 signal replica generation . . . . .	1734
<a href="#">galileo_e5_signal_replica.h</a>	This library implements various functions for Galileo E5 signal replica generation . . . . .	1736
<a href="#">galileo_e6_signal_replica.h</a>	This library implements various functions for Galileo E6 signal replica generation . . . . .	1738
<a href="#">geofunctions.h</a>	A set of coordinate transformations functions and helpers, some of them migrated from MATLAB, for geographic information systems . . . . .	1740
<a href="#">glonass_l1_signal_replica.h</a>	. . . . .	1744
<a href="#">glonass_l2_signal_replica.h</a>	This file implements various functions for GLONASS L2 CA signal replica generation . . . . .	1745
<a href="#">gnss_circular_deque.h</a>	This class implements a circular deque for <a href="#">Gnss_Synchro</a> . . . . .	1747
<a href="#">gnss_sdr_create_directory.h</a>	Create a directory . . . . .	1749
<a href="#">gnss_sdr_fft.h</a>	Helper file for FFT interface . . . . .	1751
<a href="#">gnss_sdr_filesystem.h</a>	Helper file for filesystem library interface . . . . .	1753
<a href="#">gnss_sdr_flags.h</a>	Helper file for gnss-sdr cmdline flags . . . . .	1755
<a href="#">gnss_sdr_make_unique.h</a>	This file implements <code>std::make_unique</code> for C++11 . . . . .	1759
<a href="#">gnss_sdr_string_literals.h</a>	This file implements the <code>""s</code> operator for <code>std::string</code> in C++11, and puts it into the <code>std::string_</code> ↵ literals namespace. This is already implemented in C++14, so this is only compiled when using C++11. The <code>.cc</code> file is required for avoiding the duplication of symbols . . . . .	1760
<a href="#">gnss_signal_replica.h</a>	This library gathers a few functions used for GNSS signal replica generation regardless of system used . . . . .	1761
<a href="#">gnss_time.h</a>	Class that stores both the receiver time, relative to the receiver start and the GNSS time (absolute) . . . . .	1763
<a href="#">gps_l2c_signal_replica.h</a>	This file implements signal generators for GPS L2C signals . . . . .	1764
<a href="#">gps_l5_signal_replica.h</a>	This file implements signal generators for GPS L5 signals . . . . .	1766
<a href="#">gps_sdr_signal_replica.h</a>	This file implements functions for GPS L1 C/A signal replica generation . . . . .	1767
<a href="#">item_type_helpers.h</a>	Utility functions for converting between item types . . . . .	1769
<a href="#">clFFT.h</a>	FFT in OpenCL . . . . .	1771

<a href="#">fft_base_kernels.h</a>	FFT base kernels for OpenCL . . . . .	1774
<a href="#">fft_internal.h</a>	Internals of FFT for OpenCL . . . . .	1778
<a href="#">pass_through.h</a>	Interface of a block that just puts its input in its output . . . . .	1781
<a href="#">rtklib.h</a>	Main header file for the rtklib library . . . . .	1783
<a href="#">rtklib_conversions.h</a>	GNSS-SDR to RTKLIB data structures conversion functions . . . . .	1808
<a href="#">rtklib_ephemeris.h</a>	Satellite ephemeris and clock functions . . . . .	1810
<a href="#">rtklib_ionex.h</a>	Ionex functions . . . . .	1813
<a href="#">rtklib_lambda.h</a>	Integer ambiguity resolution . . . . .	1815
<a href="#">rtklib_pntpos.h</a>	Standard code-based positioning . . . . .	1817
<a href="#">rtklib_ppp.h</a>	Precise Point Positioning . . . . .	1822
<a href="#">rtklib_preceph.h</a>	Precise ephemeris and clock functions . . . . .	1827
<a href="#">rtklib_rtcn.h</a>	RTCM functions headers . . . . .	1830
<a href="#">rtklib_rtcn2.h</a>	RTCM v2 functions headers . . . . .	1832
<a href="#">rtklib_rtcn3.h</a>	RTCM v3 functions headers . . . . .	1834
<a href="#">rtklib_rtkcmn.h</a>	Rtklib common functions . . . . .	1840
<a href="#">rtklib_rtkpos.h</a>	Rtklib ppp-related functions . . . . .	1848
<a href="#">rtklib_rtksvr.h</a>	Rtk server functions . . . . .	1852
<a href="#">rtklib_sbas.h</a>	Sbas functions . . . . .	1856
<a href="#">rtklib_solution.h</a>	Solution functions headers . . . . .	1859
<a href="#">rtklib_stream.h</a>	Streaming functions . . . . .	1863
<a href="#">rtklib_tides.h</a>	Tidal displacement corrections . . . . .	1870
<a href="#">short_x2_to_cshort.h</a>	Adapts two short streams into a std::complex<short> stream . . . . .	1872
<a href="#">trackingcmd.h</a>	Class that stores information to update the GNSS signal tracking estimations . . . . .	1874
<a href="#">hybrid_observables.h</a>	Implementation of an adapter of an observables block accepting all kind of signals to a <a href="#">ObservablesInterface</a> . . . . .	1875
<a href="#">hybrid_observables_gs.h</a>	Interface of the observables computation block . . . . .	1877
<a href="#">obs_conf.h</a>	Class that contains all the configuration parameters for generic observables block . . . . .	1880
<a href="#">rtklib_pvt.h</a>	Interface of a Position Velocity and Time computation block . . . . .	1882
<a href="#">rtklib_pvt_gs.h</a>	Interface of a Position Velocity and Time computation block . . . . .	1885

<a href="#">an_packet_printer.h</a>	Interface of a class that prints PVT solutions in a serial device following a custom version of the Advanced Navigation Packet Protocol . . . . .	1890
<a href="#">geohash.h</a>	Interface of a class that encodes / decodes geohashes . . . . .	1893
<a href="#">geojson_printer.h</a>	Interface of a class that prints PVT solutions in GeoJSON format . . . . .	1894
<a href="#">gpx_printer.h</a>	Interface of a class that prints PVT information to a gpx file . . . . .	1896
<a href="#">has_simple_printer.h</a>	Interface of a class that prints HAS messages content in a txt file . . . . .	1897
<a href="#">kml_printer.h</a>	Interface of a class that prints PVT information to a kml file . . . . .	1899
<a href="#">monitor_ephemeris_udp_sink.h</a>	Interface of a class that sends serialized <a href="#">Gps_Ephemeris</a> and <a href="#">Galileo_Ephemeris</a> objects over udp to one or multiple endpoints . . . . .	1900
<a href="#">monitor_pvt.h</a>	Interface of the <a href="#">Monitor_Pvt</a> class . . . . .	1902
<a href="#">monitor_pvt_udp_sink.h</a>	Interface of a class that sends serialized <a href="#">Monitor_Pvt</a> objects over udp to one or multiple endpoints . . . . .	1905
<a href="#">nmea_printer.h</a>	Interface of a NMEA 2.1 printer for GNSS-SDR This class provides a implementation of a subset of the NMEA-0183 standard for interfacing marine electronic devices as defined by the National Marine Electronics Association (NMEA). See <a href="https://www.nmea.org/">https://www.nmea.org/</a> for the NMEA 183 standard . . . . .	1907
<a href="#">pvt_conf.h</a>	Class that contains all the configuration parameters for the PVT block . . . . .	1909
<a href="#">pvt_kf.h</a>	Kalman Filter for Position and Velocity . . . . .	1911
<a href="#">pvt_solution.h</a>	Interface of a base class for a PVT solution . . . . .	1913
<a href="#">rinex_printer.h</a>	Interface of a RINEX 2.11 / 3.01 printer See <a href="ftp://igs.org/pub/data/format/rinex301.pdf">ftp://igs.org/pub/data/format/rinex301.pdf</a> . . . . .	1916
<a href="#">rtcm.h</a>	Interface for the RTCM 3.2 Standard . . . . .	1931
<a href="#">rtcm_printer.h</a>	Interface of a RTCM 3.2 printer for GNSS-SDR This class provides a implementation of a subset of the RTCM Standard 10403.2 for Differential GNSS Services . . . . .	1952
<a href="#">rtklib_solver.h</a>	PVT solver based on rtklib library functions adapted to the GNSS-SDR data flow and structures	1956
<a href="#">serdes_galileo_eph.h</a>	Serialization / Deserialization of <a href="#">Galileo_Ephemeris</a> objects using Protocol Buffers . . . . .	1959
<a href="#">serdes_gps_eph.h</a>	Serialization / Deserialization of <a href="#">Gps_Ephemeris</a> objects using Protocol Buffers . . . . .	1962
<a href="#">serdes_monitor_pvt.h</a>	Serialization / Deserialization of <a href="#">Monitor_Pvt</a> objects using Protocol Buffers . . . . .	1966
<a href="#">direct_resampler_conditioner.h</a>	Interface of an adapter of a direct resampler conditioner block to a <a href="#">SignalConditionerInterface</a> .	1969
<a href="#">mmse_resampler_conditioner.h</a>	Interface of an adapter of a mmse resampler conditioner block to a <a href="#">SignalConditionerInterface</a>	1971
<a href="#">direct_resampler_conditioner_cb.h</a>	Nearest neighborhood resampler with <code>std::complex&lt;signed char&gt;</code> input and <code>std::complex&lt;signed char&gt;</code> output . . . . .	1972
<a href="#">direct_resampler_conditioner_cc.h</a>	Nearest neighborhood resampler with <code>gr_complex</code> input and <code>gr_complex</code> output . . . . .	1974

<a href="#">direct_resampler_conditioner_cs.h</a>	
Nearest neighborhood resampler with std::complex<short> input and std::complex<short> output . . . . .	1976
<a href="#">signal_generator.h</a>	
Adapter of a class that generates synthesized GNSS signal . . . . .	1978
<a href="#">signal_generator_c.h</a>	
GNU Radio source block that generates synthesized GNSS signal . . . . .	1980
<a href="#">ad936x_custom_signal_source.h</a>	
A direct IIO custom front-end gnss-sdr signal source for the AD936x AD front-end family with special FPGA custom functionalities . . . . .	1983
<a href="#">adrv9361_z7035_signal_source_fpga.h</a>	
Signal source for the Analog Devices ADRV9361-Z7035 evaluation board directly connected to the FPGA accelerators. This source implements only the AD9361 control. It is NOT compatible with conventional SDR acquisition and tracking blocks. Please use the fmcomms2 source if conventional SDR acquisition and tracking is selected in the configuration file . . . . .	1986
<a href="#">custom_udp_signal_source.h</a>	
Receives ip frames containing samples in UDP frame encapsulation using a high performance packet capture library (libpcap) . . . . .	1988
<a href="#">dma_signal_source_fpga.h</a>	
Signal source for a DMA connected directly to FPGA accelerators. This source implements only the DMA control. It is NOT compatible with conventional SDR acquisition and tracking blocks . . . . .	1990
<a href="#">fifo_signal_source.h</a>	
Header file of the class for retrieving samples through a Unix FIFO . . . . .	1992
<a href="#">file_signal_source.h</a>	
Interface of a class that reads signals samples from a file and adapts it to a <a href="#">SignalSourceInterface</a> . . . . .	1994
<a href="#">file_source_base.h</a>	
Header file of the base class to file-oriented signal_source GNSS blocks . . . . .	1995
<a href="#">file_timestamp_signal_source.h</a>	
This class reads samples stored in a file and generate stream tags with its timestamp information stored in separated file . . . . .	1999
<a href="#">flexiband_signal_source.h</a>	
Signal Source adapter for the Teleorbit Flexiband front-end device. This adapter requires a Flexiband GNU Radio driver installed (not included with GNSS-SDR) . . . . .	2000
<a href="#">fmcomms2_signal_source.h</a>	
Interface to use SDR hardware based in FMCOMMS2 driver from analog devices, for example FMCOMMS4 and ADALM-PLUTO (PlutoSdr) . . . . .	2002
<a href="#">fmcomms5_signal_source_fpga.h</a>	
Signal source for the Analog Devices FMCOMMS5 directly connected to the FPGA accelerators. This source implements only the AD9361 control. It is NOT compatible with conventional SDR acquisition and tracking blocks. Please use the fmcomms2 source if conventional SDR acquisition and tracking is selected in the configuration file . . . . .	2004
<a href="#">four_bit_cpx_file_signal_source.h</a>	
Interface of a class that reads signals samples from a 2 bit complex sampler front-end file and adapts it to a <a href="#">SignalSourceInterface</a> . . . . .	2007
<a href="#">gen_signal_source.h</a>	
It wraps blocks that generates synthesized GNSS signal and filters it . . . . .	2008
<a href="#">ion_gsms_signal_source.h</a>	
GNSS-SDR Signal Source that reads sample streams following ION's GNSS-SDR metadata standard . . . . .	2010
<a href="#">labsat_signal_source.h</a>	
LabSat version 2, 3, and 3 Wideband format reader . . . . .	2011
<a href="#">limesdr_signal_source.h</a>	
. . . . .	2013
<a href="#">max2771_evkit_signal_source_fpga.h</a>	
Signal source for the MAX2771EVKIT evaluation board connected directly to FPGA accelerators. This source implements only the MAX2771 control. It is NOT compatible with conventional SDR acquisition and tracking blocks . . . . .	2014

<a href="#">multichannel_file_signal_source.h</a>	Implementation of a class that reads signals samples from files at different frequency band and adapts it to a <a href="#">SignalSourceInterface</a> . . . . .	2017
<a href="#">nsr_file_signal_source.h</a>	Implementation of a class that reads signals samples from a NSR 2 bits sampler front-end file and adapts it to a <a href="#">SignalSourceInterface</a> . More information about the front-end here <a href="http://www.ifen.com/products/sx-scientific-gnss-solutions/nsr-software-receiver.html">http://www.ifen.com/products/sx-scientific-gnss-solutions/nsr-software-receiver.html</a> . . . . .	2019
<a href="#">osmosdr_signal_source.h</a>	Signal source wrapper for OsmoSDR-compatible front-ends, such as HackRF or Realtek's RTL2832U-based USB dongle DVB-T receivers (see <a href="https://osmocom.org/projects/rtl-sdr/wiki">https://osmocom.org/projects/rtl-sdr/wiki</a> for more information) . . . . .	2021
<a href="#">plutosdr_signal_source.h</a>	Signal source for PlutoSDR . . . . .	2023
<a href="#">raw_array_signal_source.h</a>	CTTC Experimental GNSS 8 channels array signal source . . . . .	2025
<a href="#">rtl_tcp_signal_source.h</a>	Signal source which reads from rtl_tcp. (see <a href="https://osmocom.org/projects/rtl-sdr/wiki">https://osmocom.org/projects/rtl-sdr/wiki</a> for more information) . . . . .	2027
<a href="#">signal_source_base.h</a>	Header file of the base class to signal_source GNSS blocks . . . . .	2029
<a href="#">spir_file_signal_source.h</a>	Implementation of a class that reads signals samples from a SPIR file and adapts it to a <a href="#">SignalSourceInterface</a> . . . . .	2030
<a href="#">spir_gss6450_file_signal_source.h</a>	Implementation of a class that reads signals samples from a SPIR file and adapts it to a <a href="#">SignalSourceInterface</a> . . . . .	2031
<a href="#">two_bit_cpx_file_signal_source.h</a>	Interface of a class that reads signals samples from a 2 bit complex sampler front-end file and adapts it to a <a href="#">SignalSourceInterface</a> . . . . .	2034
<a href="#">two_bit_packed_file_signal_source.h</a>	Interface of a class that reads signals samples from a file. Each sample is two bits, which are packed into bytes or shorts . . . . .	2035
<a href="#">uhd_signal_source.h</a>	Interface for the Universal Hardware Driver signal source . . . . .	2037
<a href="#">zmq_signal_source.h</a>	Signal source which reads from ZeroMQ . . . . .	2039
<a href="#">ad936x_iio_source.h</a>	A direct IIO custom front-end gnss-sdr signal gnuradio block for the AD936x AD front-end family with special FPGA custom functionalities . . . . .	2041
<a href="#">fifo_reader.h</a>	Header file to retrieve samples from an existing Unix FIFO . . . . .	2044
<a href="#">gr_complex_ip_packet_source.h</a>	Receives ip frames containing samples in UDP frame encapsulation using a high performance packet capture library (libpcap) . . . . .	2046
<a href="#">ion_gsms.h</a>	GNU Radio block that reads a Block from a file following ION's GNSS-SDR metadata standard . . . . .	2049
<a href="#">labsat23_source.h</a>	Unpacks capture files in the LabSat 2 (ls2), LabSat 3 (ls3), or LabSat 3 Wideband (LS3W) formats . . . . .	2050
<a href="#">rtl_tcp_signal_source_c.h</a>	Interface of an rtl_tcp signal source reader . . . . .	2053
<a href="#">unpack_2bit_samples.h</a>	Unpacks 2 bit samples samples may be packed in any of the following ways: 1) Into bytes [ item == byte ] 1a) Big endian ordering within the byte 1b) Little endian ordering within the byte 2) Into shorts [ item == short ] 2a) Big endian ordering of bytes, big endian within the byte 2b) Big endian ordering of bytes, little endian within the byte 2c) Little endian ordering of bytes, big endian within the byte 2d) Little endian ordering of bytes, little endian within the byte . . . . .	2055

<a href="#">unpack_byte_2bit_cpx_samples.h</a>	Unpacks byte samples to 2 bits complex samples. Packing Order Most Significant Nibble - Sample n Least Significant Nibble - Sample n+1 Packing order in Nibble Q1 Q0 I1 I0 . . . . .	2058
<a href="#">unpack_byte_2bit_samples.h</a>	Unpacks byte samples to NSR 2 bits samples . . . . .	2060
<a href="#">unpack_byte_4bit_samples.h</a>	Unpacks byte samples to 4 bits samples. Packing Order Packing order in Nibble I0 I1 I2 I3 I0 I1 I2 I3 . . . . .	2062
<a href="#">unpack_intspir_1bit_samples.h</a>	Unpacks SPIR int samples to NSR 1 bit samples . . . . .	2064
<a href="#">unpack_short_byte_samples.h</a>		2066
<a href="#">unpack_spir_gss6450_samples.h</a>	Unpacks SPIR int samples . . . . .	2067
<a href="#">ad9361_manager.h</a>	An Analog Devices AD9361 front-end configuration library wrapper for configure some functions via iiod link . . . . .	2069
<a href="#">ad936x_iio_custom.h</a>	A direct IIO custom front-end driver for the AD936x AD front-end family with special FPGA custom functionalities . . . . .	2072
<a href="#">ad936x_iio_samples.h</a>	A class that holds a custom sample buffer for Analog Devices AD936x family front-ends . . . .	2075
<a href="#">fpga_buffer_monitor.h</a>	Check receiver buffer overflow and monitor the status of the receiver buffers . . . . .	2077
<a href="#">fpga_dma-proxy.h</a>	FPGA DMA control. This code is based in the Xilinx DMA proxy test application: <a href="https://github.com/Xilinx-Wiki-Projects/software-prototypes/tree/master/linux-user-software">https://github.com/Xilinx-Wiki-Projects/software-prototypes/tree/master/linux-user-software</a> . . . . .	2079
<a href="#">fpga_dynamic_bit_selection.h</a>	Dynamic bit selection in the received signal . . . . .	2082
<a href="#">fpga_spidev.h</a>	FPGA SPI control . . . . .	2084
<a href="#">fpga_switch.h</a>	Switch that connects the HW accelerator queues to the analog front end or the DMA . . . . .	2086
<a href="#">gnss_sdr_timestamp.h</a>	GNURadio block that adds to sample stream timestamp metadata information stored on a separated file . . . . .	2088
<a href="#">gnss_sdr_valve.h</a>	Interface of a GNU Radio block that sends a STOP message to the control queue right after a specific number of samples have passed through it . . . . .	2090
<a href="#">ion_gsms_chunk_data.h</a>	Holds logic for reading and decoding samples from a chunk . . . . .	2092
<a href="#">ion_gsms_chunk_unpacking_ctx.h</a>	Holds state and provides utilities for unpacking samples from a chunk . . . . .	2096
<a href="#">ion_gsms_stream_encodings.h</a>	Implements look up tables for all encodings in the standard . . . . .	2099
<a href="#">pps_samplestamp.h</a>	A simple container for the sample count associated to PPS rising edge . . . . .	2106
<a href="#">ppstcprx.h</a>	TCP client class for front-end PPS samplestamp information reception . . . . .	2107
<a href="#">rtl_tcp_commands.h</a>	Defines structures and constants for communicating with rtl_tcp . . . . .	2109
<a href="#">rtl_tcp_dongle_info.h</a>	Interface for a structure sent by rtl_tcp defining the hardware . . . . .	2111
<a href="#">beidou_b1i_telemetry_decoder.h</a>	Interface of an adapter of a Beidou B1I NAV data decoder block to a <a href="#">TelemetryDecoderInterface</a>	2113
<a href="#">beidou_b3i_telemetry_decoder.h</a>	Interface of an adapter of a Beidou B3I NAV data decoder block to a <a href="#">TelemetryDecoderInterface</a>	2114

<a href="#">galileo_e1b_telemetry_decoder.h</a>	
Interface of an adapter of a GALILEO E1B NAV data decoder block to a <a href="#">TelemetryDecoderInterface</a>	
2116	
<a href="#">galileo_e5a_telemetry_decoder.h</a>	
Interface of an adapter of a GALILEO E5a FNAV data decoder block to a <a href="#">TelemetryDecoderInterface</a>	
2118	
<a href="#">galileo_e5b_telemetry_decoder.h</a>	
Interface of an adapter of a GALILEO E5B NAV data decoder block to a <a href="#">TelemetryDecoderInterface</a>	
2120	
<a href="#">galileo_e6_telemetry_decoder.h</a>	
Interface of an adapter of a GALILEO E6 CNAV data decoder block to a <a href="#">TelemetryDecoderInterface</a>	
2122	
<a href="#">glonass_l1_ca_telemetry_decoder.h</a>	
Interface of an adapter of a GLONASS L1 C/A NAV data decoder block to a <a href="#">TelemetryDecoderInterface</a>	
2124	
<a href="#">glonass_l2_ca_telemetry_decoder.h</a>	
Interface of an adapter of a GLONASS L2 C/A NAV data decoder block to a <a href="#">TelemetryDecoderInterface</a>	
2126	
<a href="#">gps_l1_ca_telemetry_decoder.h</a>	
Interface of an adapter of a GPS L1 C/A NAV data decoder block to a <a href="#">TelemetryDecoderInterface</a>	2127
<a href="#">gps_l2c_telemetry_decoder.h</a>	
Interface of an adapter of a GPS L2C (CNAV) data decoder block to a <a href="#">TelemetryDecoderInterface</a>	2129
<a href="#">gps_l5_telemetry_decoder.h</a>	
Interface of an adapter of a GPS L5 (CNAV) data decoder block to a <a href="#">TelemetryDecoderInterface</a>	2131
<a href="#">sbas_l1_telemetry_decoder.h</a>	
Interface of an adapter of a SBAS telemetry data decoder block to a <a href="#">TelemetryDecoderInterface</a>	2133
<a href="#">beidou_b1i_telemetry_decoder_gs.h</a>	
Implementation of a BEIDOU B1I DNAV data decoder block	2135
<a href="#">beidou_b3i_telemetry_decoder_gs.h</a>	
Implementation of a BEIDOU B3I DNAV data decoder block	2137
<a href="#">galileo_telemetry_decoder_gs.h</a>	
Implementation of a Galileo unified INAV and FNAV message demodulator block	2140
<a href="#">glonass_l1_ca_telemetry_decoder_gs.h</a>	
Implementation of a GLONASS L1 C/A NAV data decoder block	2143
<a href="#">glonass_l2_ca_telemetry_decoder_gs.h</a>	
Implementation of a GLONASS L2 C/A NAV data decoder block	2146
<a href="#">gps_l1_ca_telemetry_decoder_gs.h</a>	
Interface of a NAV message demodulator block based on Kay Borre book MATLAB-based GPS receiver	2149
<a href="#">gps_l2c_telemetry_decoder_gs.h</a>	
Interface of a CNAV message demodulator block	2152
<a href="#">gps_l5_telemetry_decoder_gs.h</a>	
Interface of a CNAV message demodulator block	2154
<a href="#">sbas_l1_telemetry_decoder_gs.h</a>	
Interface of a SBAS telemetry data decoder block	2157
<a href="#">bits.h</a>	
Utilities for bit manipulation of the libswiftnav library	2160
<a href="#">cnav_msg.h</a>	
Utilities for CNAV message manipulation of the libswiftnav library	2162
<a href="#">edc.h</a>	
Utilities for CRC computation of the libswiftnav library	2165
<a href="#">fec.h</a>	
Utilities for the convolutional encoder of the libswiftnav library	2166
<a href="#">swift_common.h</a>	
Common definitions used throughout the libswiftnav library	2169
<a href="#">tlm_conf.h</a>	
Class that contains all the configuration parameters for generic telemetry decoder block	2170



<a href="#">tlm_crc_stats.h</a>	
Class that computes the telemetry CRC statistics . . . . .	2172
<a href="#">tlm_utils.h</a>	
Utilities for the telemetry decoder blocks . . . . .	2174
<a href="#">viterbi_decoder.h</a>	
Class that implements a Viterbi decoder . . . . .	2175
<a href="#">viterbi_decoder_sbass.h</a>	
Interface of a Viterbi decoder class based on the Iterative Solutions Coded Modulation Library by Matthew C. Valenti . . . . .	2177
<a href="#">beidou_b1i_dll_pll_tracking.h</a>	
Interface of an adapter of a DLL+PLL tracking loop block for Beidou B1I to a <a href="#">TrackingInterface</a> .	2179
<a href="#">beidou_b3i_dll_pll_tracking.h</a>	
Interface of an adapter of a DLL+PLL tracking loop block for Beidou B3I to a <a href="#">TrackingInterface</a> .	2181
<a href="#">galileo_e1_dll_pll_veml_tracking.h</a>	
Adapts a DLL+PLL VEML (Very Early Minus Late) tracking loop block to a <a href="#">TrackingInterface</a> for Galileo E1 signals . . . . .	2183
<a href="#">galileo_e1_dll_pll_veml_tracking_fpga.h</a>	
Adapts a DLL+PLL VEML (Very Early Minus Late) tracking loop block to a <a href="#">TrackingInterface</a> for Galileo E1 signals for the FPGA . . . . .	2185
<a href="#">galileo_e1_tcp_connector_tracking.h</a>	
Interface of an adapter of a TCP connector block based on code DLL + carrier PLL for Galileo E1 to a <a href="#">TrackingInterface</a> . . . . .	2187
<a href="#">galileo_e5a_dll_pll_tracking.h</a>	
Adapts a code DLL + carrier PLL tracking block to a <a href="#">TrackingInterface</a> for Galileo E5a signals .	2189
<a href="#">galileo_e5a_dll_pll_tracking_fpga.h</a>	
Adapts a code DLL + carrier PLL tracking block to a <a href="#">TrackingInterface</a> for Galileo E5a signals for the FPGA . . . . .	2191
<a href="#">galileo_e5b_dll_pll_tracking.h</a>	
Adapts a code DLL + carrier PLL tracking block to a <a href="#">TrackingInterface</a> for Galileo E5b signals .	2193
<a href="#">galileo_e6_dll_pll_tracking.h</a>	
Adapts a code DLL + carrier PLL tracking block to a <a href="#">TrackingInterface</a> for Galileo E6 signals . .	2195
<a href="#">glonass_l1_ca_dll_pll_c_aid_tracking.h</a>	
Interface of an adapter of a DLL+PLL tracking loop block for Glonass L1 C/A to a <a href="#">TrackingInterface</a> . . . . .	2197
<a href="#">glonass_l1_ca_dll_pll_tracking.h</a>	
Interface of an adapter of a DLL+PLL tracking loop block for Glonass L1 C/A to a <a href="#">TrackingInterface</a> . . . . .	2199
<a href="#">glonass_l2_ca_dll_pll_c_aid_tracking.h</a>	
Interface of an adapter of a DLL+PLL tracking loop block for Glonass L2 C/A to a <a href="#">TrackingInterface</a> . . . . .	2201
<a href="#">glonass_l2_ca_dll_pll_tracking.h</a>	
Interface of an adapter of a DLL+PLL tracking loop block for Glonass L2 C/A to a <a href="#">TrackingInterface</a> . . . . .	2203
<a href="#">gps_l1_ca_dll_pll_tracking.h</a>	
Interface of an adapter of a DLL+PLL tracking loop block for GPS L1 C/A to a <a href="#">TrackingInterface</a>	2205
<a href="#">gps_l1_ca_dll_pll_tracking_fpga.h</a>	
Interface of an adapter of a DLL+PLL tracking loop block for GPS L1 C/A to a <a href="#">TrackingInterface</a> for the FPGA . . . . .	2207
<a href="#">gps_l1_ca_dll_pll_tracking_gpu.h</a>	
Implementation of an adapter of a DLL+PLL tracking loop block using GPU accelerated functions for GPS L1 C/A to a <a href="#">TrackingInterface</a> . . . . .	2210
<a href="#">gps_l1_ca_gaussian_tracking.h</a>	
Interface of an adapter of a DLL + Kalman carrier tracking loop block for GPS L1 C/A signals .	2212
<a href="#">gps_l1_ca_kf_tracking.h</a>	
Interface of an adapter of a code + carrier Kalman Filter tracking loop with VTL capabilities block for GPS L1 C/A to a <a href="#">TrackingInterface</a> . . . . .	2214

<a href="#">gps_l1_ca_tcp_connector_tracking.h</a>	
Interface of an adapter of a TCP connector block based on code DLL + carrier PLL for GPS L1 C/A to a <a href="#">TrackingInterface</a>	2215
<a href="#">gps_l2_m_dll_pll_tracking.h</a>	
Interface of an adapter of a DLL+PLL tracking loop block for GPS L1 C/A to a <a href="#">TrackingInterface</a>	2217
<a href="#">gps_l2_m_dll_pll_tracking_fpga.h</a>	
Interface of an adapter of a DLL+PLL tracking loop block for GPS L2C to a <a href="#">TrackingInterface</a> for the FPGA	2219
<a href="#">gps_l5_dll_pll_tracking.h</a>	
Interface of an adapter of a DLL+PLL tracking loop block for GPS L5 to a <a href="#">TrackingInterface</a>	2221
<a href="#">gps_l5_dll_pll_tracking_fpga.h</a>	
Interface of an adapter of a DLL+PLL tracking loop block for GPS L5 to a <a href="#">TrackingInterface</a> for the FPGA	2223
<a href="#">dll_pll_veml_tracking.h</a>	
Implementation of a code DLL + carrier PLL tracking block	2226
<a href="#">dll_pll_veml_tracking_fpga.h</a>	
Implementation of a code DLL + carrier PLL tracking block using an FPGA	2229
<a href="#">galileo_e1_tcp_connector_tracking_cc.h</a>	
Interface of a TCP connector block based on code DLL + carrier PLL VEML (Very Early Minus Late) tracking block for Galileo E1 signals	2233
<a href="#">glonass_l1_ca_dll_pll_c_aid_tracking_cc.h</a>	
Implementation of a code DLL + carrier PLL tracking block	2237
<a href="#">glonass_l1_ca_dll_pll_c_aid_tracking_sc.h</a>	
Implementation of a code DLL + carrier PLL tracking block	2240
<a href="#">glonass_l1_ca_dll_pll_tracking_cc.h</a>	
Implementation of a code DLL + carrier PLL tracking block	2244
<a href="#">glonass_l2_ca_dll_pll_c_aid_tracking_cc.h</a>	
Implementation of a code DLL + carrier PLL tracking block	2248
<a href="#">glonass_l2_ca_dll_pll_c_aid_tracking_sc.h</a>	
Implementation of a code DLL + carrier PLL tracking block	2251
<a href="#">glonass_l2_ca_dll_pll_tracking_cc.h</a>	
Implementation of a code DLL + carrier PLL tracking block	2255
<a href="#">gps_l1_ca_dll_pll_tracking_gpu_cc.h</a>	
Implementation of a code DLL + carrier PLL tracking block, GPU ACCELERATED	2258
<a href="#">gps_l1_ca_gaussian_tracking_cc.h</a>	
Interface of a processing block of a DLL + Kalman carrier tracking loop for GPS L1 C/A signals	2262
<a href="#">gps_l1_ca_tcp_connector_tracking_cc.h</a>	
Interface of a TCP connector block based on code DLL + carrier PLL	2266
<a href="#">kf_tracking.h</a>	2269
<a href="#">bayesian_estimation.h</a>	
Interface of a library with Bayesian noise statistic estimation	2271
<a href="#">cpu_multicorrelator.h</a>	
High optimized CPU vector multiTAP correlator class	2274
<a href="#">cpu_multicorrelator_16sc.h</a>	
Highly optimized CPU vector multiTAP correlator class for lv_16sc_t (short int complex)	2276
<a href="#">cpu_multicorrelator_real_codes.h</a>	
Highly optimized CPU vector multiTAP correlator class using real-valued local codes	2277
<a href="#">cuda_multicorrelator.h</a>	
Highly optimized CUDA GPU vector multiTAP correlator class	2279
<a href="#">dll_pll_conf.h</a>	
Class that contains all the configuration parameters for generic tracking block based on a DLL and a PLL	2283
<a href="#">dll_pll_conf_fpga.h</a>	
Class that contains all the configuration parameters for generic tracking block based on a DLL and a PLL for the FPGA	2285
<a href="#">exponential_smoother.h</a>	
Class that implements an exponential smoother	2287

<a href="#">fpga_multicorrelator.h</a>	
FPGA vector correlator class	2289
<a href="#">kf_conf.h</a>	2293
<a href="#">lock_detectors.h</a>	
Interface of a library with a set of code and carrier phase lock detectors	2294
<a href="#">nonlinear_tracking.h</a>	
Interface of a library for nonlinear tracking algorithms	2296
<a href="#">tcp_communication.h</a>	
Interface of the TCP communication class	2299
<a href="#">tcp_packet_data.h</a>	
Interface of the TCP data packet class	2301
<a href="#">tracking_2nd_DLL_filter.h</a>	
Interface of a 2nd order DLL filter for code tracking loop	2302
<a href="#">tracking_2nd_PLL_filter.h</a>	
Interface of a 2nd order PLL filter for carrier tracking loop	2304
<a href="#">tracking_discriminators.h</a>	
Interface of a library with a set of code tracking and carrier tracking discriminators	2305
<a href="#">tracking_FLL_PLL_filter.h</a>	
Interface of a hybrid FLL and PLL filter for tracking carrier loop	2309
<a href="#">tracking_loop_filter.h</a>	
Generic 1st to 3rd order loop filter implementation	2310
<a href="#">acquisition_interface.h</a>	
Header file of the interface to an acquisition GNSS block	2312
<a href="#">channel_interface.h</a>	
This class represents an interface to a channel GNSS block	2314
<a href="#">configuration_interface.h</a>	
This class represents an interface to configuration parameters	2315
<a href="#">gnss_block_interface.h</a>	
This interface represents a GNSS block	2317
<a href="#">observables_interface.h</a>	
This class represents an interface to an Observables block	2319
<a href="#">pvt_interface.h</a>	
This class represents an interface to a PVT block	2321
<a href="#">signal_source_interface.h</a>	2323
<a href="#">telemetry_decoder_interface.h</a>	
This class represents an interface to a telemetry decoder block	2324
<a href="#">tracking_interface.h</a>	
This class represents an interface to a tracking block	2325
<a href="#">channel_event.h</a>	
Class that defines a channel event	2326
<a href="#">channel_status_msg_receiver.h</a>	
GNU Radio block that receives asynchronous channel messages from acquisition and tracking blocks	2328
<a href="#">command_event.h</a>	
Class that defines a receiver command event	2330
<a href="#">galileo_e6_has_msg_receiver.h</a>	
GNU Radio block that processes Galileo HAS message pages received from Galileo E6B telemetry blocks. After successful decoding, sends the content to the PVT block	2332
<a href="#">galileo_tow_map.h</a>	
GNU Radio block that stores TOW for Galileo channels	2335
<a href="#">gnss_crypto.h</a>	
Class for computing cryptographic functions	2337
<a href="#">gnss_sdr_fpga_sample_counter.h</a>	
Simple block to report the current receiver time based on the output of the tracking or telemetry blocks	2339
<a href="#">gnss_sdr_sample_counter.h</a>	
Simple block to report the current receiver time based on the output of the tracking or telemetry blocks	2341

<a href="#">gnss_sdr_supl_client.h</a>	Class that implements a C++ interface to external Secure User Location Protocol (SUPL) client library . . . . .	2343
<a href="#">gnss_sdr_time_counter.h</a>	Simple block to report the current receiver time based on the output of the tracking or telemetry blocks . . . . .	2347
<a href="#">ini.h</a>	This function parses an INI file into easy-to-access name/value pairs . . . . .	2349
<a href="#">INIReader.h</a>	This class reads an INI file into easy-to-access name/value pairs . . . . .	2351
<a href="#">nav_message_monitor.h</a>	GNU Radio block that processes <a href="#">Nav_Message_Packet</a> received from telemetry blocks and sends them via UDP . . . . .	2353
<a href="#">nav_message_packet.h</a>	Class for storage of decoded navigation messages . . . . .	2354
<a href="#">nav_message_udp_sink.h</a>	Interface of a class that sends serialized <a href="#">Nav_Message_Packet</a> objects over UDP to one or multiple endpoints . . . . .	2356
<a href="#">osnma_helper.h</a>	Class for auxiliary osnma functions . . . . .	2358
<a href="#">osnma_msg_receiver.h</a>	GNU Radio block that processes Galileo OSNMA data received from Galileo E1B telemetry blocks. After successful decoding, sends the content to the PVT block . . . . .	2360
<a href="#">osnma_nav_data_manager.h</a>	Class for Galileo OSNMA navigation data management . . . . .	2364
<a href="#">serdes_nav_message.h</a>	Serialization / Deserialization of <a href="#">Nav_Message_Packet</a> objects using Protocol Buffers . . . . .	2365
<a href="#">string_converter.h</a>	Interface of a class that interprets the contents of a string and converts it into different types . . . . .	2368
<a href="#">uio_fpga.h</a>	This library contains functions to determine the uio device driver file that corresponds to a hardware accelerator device name in the FPGA . . . . .	2370
<a href="#">gnss_synchro_monitor.h</a>	Interface of a receiver monitoring block which allows sending a data stream with the receiver internal parameters ( <a href="#">Gnss_Synchro</a> objects) to local or remote clients over UDP . . . . .	2371
<a href="#">gnss_synchro_udp_sink.h</a>	Interface of a class that sends serialized <a href="#">Gnss_Synchro</a> objects over udp to one or multiple endpoints . . . . .	2373
<a href="#">serdes_gnss_synchro.h</a>	Serialization / Deserialization of <a href="#">Gnss_Synchro</a> objects using Protocol Buffers . . . . .	2375
<a href="#">concurrent_map.h</a>	Interface of a thread-safe std::map . . . . .	2379
<a href="#">concurrent_queue.h</a>	Interface of a thread-safe std::queue . . . . .	2380
<a href="#">control_thread.h</a>	Interface of the receiver control plane . . . . .	2383
<a href="#">file_configuration.h</a>	A <a href="#">ConfigurationInterface</a> that reads the configuration from a file . . . . .	2386
<a href="#">gnss_block_factory.h</a>	Interface of a factory that returns smart pointers to GNSS blocks . . . . .	2388
<a href="#">gnss_flowgraph.h</a>	Interface of a GNSS receiver flow graph . . . . .	2390
<a href="#">in_memory_configuration.h</a>	A <a href="#">ConfigurationInterface</a> for testing purposes . . . . .	2395
<a href="#">tcp_cmd_interface.h</a>	Class that implements a TCP/IP telecommand command line interface for GNSS-SDR . . . . .	2397
<a href="#">agnss_ref_location.h</a>	Interface of an Assisted GNSS REFERENCE LOCATION storage . . . . .	2399

<a href="#">agnss_ref_time.h</a>	Interface of an Assisted GNSS REFERENCE TIME storage . . . . .	2401
<a href="#">Beidou_B1I.h</a>	Defines system parameters for BeiDou B1I signal and DNAV data . . . . .	2402
<a href="#">Beidou_B3I.h</a>	Defines system parameters for BeiDou B3I signal and DNAV data . . . . .	2405
<a href="#">Beidou_DNAV.h</a>	Defines system parameters for BeiDou DNAV data processing . . . . .	2407
<a href="#">beidou_dnav_almanac.h</a>	Interface of a Beidou DNAV Almanac storage . . . . .	2416
<a href="#">beidou_dnav_ephemeris.h</a>	Interface of a BEIDOU EPHEMERIS storage . . . . .	2418
<a href="#">beidou_dnav_iono.h</a>	Interface of a BEIDOU IONOSPHERIC MODEL storage . . . . .	2421
<a href="#">beidou_dnav_navigation_message.h</a>	Interface of a BeiDou DNAV Data message decoder . . . . .	2423
<a href="#">beidou_dnav_utc_model.h</a>	Interface of a BeiDou UTC MODEL storage . . . . .	2428
<a href="#">display.h</a>	Defines useful display constants . . . . .	2431
<a href="#">galileo_almanac.h</a>	Interface of a Galileo ALMANAC storage . . . . .	2433
<a href="#">galileo_almanac_helper.h</a>	Interface of a Galileo ALMANAC storage helper . . . . .	2435
<a href="#">Galileo_CNAV.h</a>	Galileo CNAV message constants. Data from: Galileo High Accuracy Service Signal-In-Space Interface Control Document (HAS SIS ICD) Issue 1.0, May 2022 . . . . .	2437
<a href="#">galileo_cnav_message.h</a>	Implementation of a Galileo CNAV Data message as described in Galileo High Accuracy Service Signal-In-Space Interface Control Document (HAS SIS ICD) Issue 1.0, May 2022 . . . . .	2442
<a href="#">Galileo_E1.h</a>	Defines system parameters for Galileo E1 signal and NAV data . . . . .	2445
<a href="#">Galileo_E5a.h</a>	Defines system parameters for Galileo E5a signal and NAV data . . . . .	2463
<a href="#">Galileo_E5b.h</a>	Defines system parameters for Galileo E5b signal and NAV data . . . . .	2507
<a href="#">Galileo_E6.h</a>	Defines system parameters for Galileo E6 B/C signal, as published at: European Union, E6-B/C Codes Technical Note, Issue 1, January 2019 . . . . .	2551
<a href="#">galileo_ephemeris.h</a>	Interface of a Galileo EPHEMERIS storage . . . . .	2575
<a href="#">Galileo_FNAV.h</a>	Galileo FNAV message constants . . . . .	2577
<a href="#">galileo_fnav_message.h</a>	Implementation of a Galileo F/NAV Data message as described in Galileo OS SIS ICD Issue 1.2 (Nov. 2015) . . . . .	2583
<a href="#">galileo_has_data.h</a>	Class for Galileo HAS message type 1 data storage . . . . .	2588
<a href="#">galileo_has_page.h</a>	Class for Galileo HAS message page storage . . . . .	2591
<a href="#">Galileo_INAV.h</a>	Galileo INAV message constants . . . . .	2593
<a href="#">galileo_inav_message.h</a>	Implementation of a Galileo I/NAV Data message as described in Galileo OS SIS ICD Issue 2.0 (Jan. 2021) . . . . .	2602
<a href="#">galileo_iono.h</a>	Interface of a Galileo Ionospheric Model storage . . . . .	2609

<a href="#">galileo_ism.h</a>	Interface of a Galileo Integrity Support Message . . . . .	2611
<a href="#">Galileo_OSNMA.h</a>	Galileo OSNMA message constants . . . . .	2615
<a href="#">galileo_reduced_ced.h</a>	Galileo Reduced Clock and Ephemeris Data storage class . . . . .	2618
<a href="#">galileo_utc_model.h</a>	Interface of a Galileo UTC MODEL storage . . . . .	2620
<a href="#">glonass_gnav_almanac.h</a>	Interface of a GLONASS GNAV ALMANAC storage . . . . .	2622
<a href="#">glonass_gnav_ephemeris.h</a>	Interface of a GLONASS EPHEMERIS storage . . . . .	2625
<a href="#">glonass_gnav_navigation_message.h</a>	Interface of a GLONASS GNAV Data message decoder as described in GLONASS ICD (Edition 5.1) . . . . .	2629
<a href="#">glonass_gnav_utc_model.h</a>	Interface of a GLONASS GNAV UTC MODEL storage . . . . .	2633
<a href="#">GLONASS_L1_L2_CA.h</a>	Defines system parameters for GLONASS L1 C/A signal and NAV data . . . . .	2636
<a href="#">gnss_almanac.h</a>	Base class for GNSS almanac storage . . . . .	2645
<a href="#">gnss_ephemeris.h</a>	Base class for GNSS Ephemeris . . . . .	2647
<a href="#">gnss_frequencies.h</a>	GNSS Frequencies . . . . .	2650
<a href="#">gnss_obs_codes.h</a>	GNSS Observable codes . . . . .	2652
<a href="#">gnss_satellite.h</a>	Interface of the <a href="#">Gnss_Satellite</a> class . . . . .	2656
<a href="#">gnss_signal.h</a>	Implementation of the <a href="#">Gnss_Signal</a> class . . . . .	2658
<a href="#">gnss_synchro.h</a>	Interface of the <a href="#">Gnss_Synchro</a> class . . . . .	2660
<a href="#">gps_acq_assist.h</a>	Interface of a GPS RRLL ACQUISITION ASSISTACE storage . . . . .	2664
<a href="#">gps_almanac.h</a>	Interface of a GPS ALMANAC storage . . . . .	2666
<a href="#">GPS_CNAV.h</a>	Defines parameters for GPS CNAV . . . . .	2668
<a href="#">gps_cnav_ephemeris.h</a>	Interface of a GPS CNAV EPHEMERIS storage . . . . .	2673
<a href="#">gps_cnav_iono.h</a>	Interface of a GPS CNAV IONOSPHERIC MODEL storage . . . . .	2676
<a href="#">gps_cnav_navigation_message.h</a>	Interface of a GPS CNAV Data message decoder . . . . .	2678
<a href="#">gps_cnav_utc_model.h</a>	Interface of a GPS CNAV UTC MODEL storage . . . . .	2681
<a href="#">gps_ephemeris.h</a>	Interface of a GPS EPHEMERIS storage . . . . .	2682
<a href="#">gps_iono.h</a>	Interface of a GPS IONOSPHERIC MODEL storage . . . . .	2685
<a href="#">GPS_L1_CA.h</a>	Defines system parameters for GPS L1 C/A signal and NAV data . . . . .	2687
<a href="#">GPS_L2C.h</a>	Defines system parameters for GPS L2C signal . . . . .	2695
<a href="#">GPS_L5.h</a>	Defines system parameters for GPS L5 signal . . . . .	2698

<a href="#">gps_navigation_message.h</a>	
Interface of a GPS NAV Data message decoder . . . . .	2701
<a href="#">gps_utc_model.h</a>	
Interface of a GPS UTC MODEL storage . . . . .	2706
<a href="#">MATH_CONSTANTS.h</a>	
Defines useful mathematical constants and their scaled versions . . . . .	2709
<a href="#">osnma_data.h</a>	
Class for Galileo OSNMA data storage . . . . .	2714
<a href="#">osnma_dsm_reader.h</a>	
Class for reading OSNMA DSM messages . . . . .	2718
<a href="#">reed_solomon.h</a>	
Class implementing a Reed-Solomon encoder/decoder for blocks of 255 symbols and symbols of 8 bits . . . . .	2720
<a href="#">sbas_ephemeris.h</a>	
Interface of a SBAS REFERENCE LOCATION storage . . . . .	2723



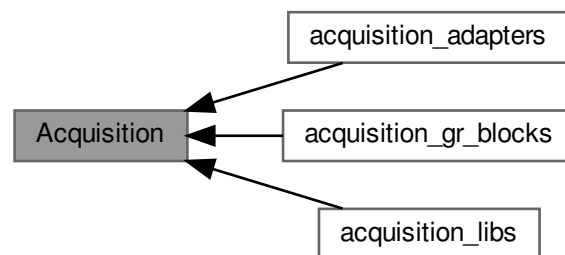


# Chapter 9

## Topic Documentation

### 9.1 Acquisition

Collaboration diagram for Acquisition:



#### Topics

- [acquisition\\_adapters](#)
- [acquisition\\_gr\\_blocks](#)
- [acquisition\\_libs](#)

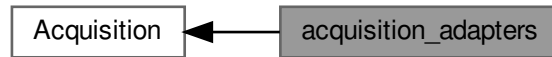
#### 9.1.1 Detailed Description

Classes for GNSS signal acquisition

Classes for GNSS signal acquisition

### 9.1.2 acquisition\_adapters

Collaboration diagram for acquisition\_adapters:



#### Classes

- class [BeidouB1iPcpsAcquisition](#)  
This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.
- class [BeidouB3iPcpsAcquisition](#)  
This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for BeiDou B3I signals.
- class [GalileoE1Pcps8msAmbiguousAcquisition](#)  
Adapts a PCPS 8ms acquisition block to an [AcquisitionInterface](#) for Galileo E1 Signals.
- class [GalileoE1PcpsAmbiguousAcquisition](#)  
This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Galileo E1 Signals.
- class [GalileoE1PcpsAmbiguousAcquisitionFpga](#)  
This class adapts a PCPS acquisition block off-loaded on an FPGA to an [AcquisitionInterface](#) for Galileo E1 Signals.
- class [GalileoE1PcpsCccwsrAmbiguousAcquisition](#)  
Adapts a PCPS CCCWSR acquisition block to an [AcquisitionInterface](#) for Galileo E1 Signals.
- class [GalileoE1PcpsQuickSyncAmbiguousAcquisition](#)  
This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Galileo E1 Signals.
- class [GalileoE1PcpsTongAmbiguousAcquisition](#)  
Adapts a PCPS Tong acquisition block to an [AcquisitionInterface](#) for Galileo E1 Signals.
- class [GalileoE5aNoncoherentIQAcquisitionCaf](#)
- class [GalileoE5aPcpsAcquisition](#)
- class [GalileoE5aPcpsAcquisitionFpga](#)  
This class adapts a PCPS acquisition block off-loaded on an FPGA to an [AcquisitionInterface](#) for Galileo E5a signals.
- class [GalileoE5bPcpsAcquisition](#)
- class [GalileoE5bPcpsAcquisitionFpga](#)  
This class adapts a PCPS acquisition block off-loaded on an FPGA to an [AcquisitionInterface](#) for Galileo E5b signals.
- class [GalileoE6PcpsAcquisition](#)  
This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Galileo E6 Signals.
- class [GlonassL1CaPcpsAcquisition](#)  
This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.
- class [GlonassL2CaPcpsAcquisition](#)  
This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GLONASS L2 C/A signals.
- class [GpsL1CaPcpsAcquisition](#)  
This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.
- class [GpsL1CaPcpsAcquisitionFineDoppler](#)  
This class Adapts a PCPS acquisition block with fine Doppler estimation to an [AcquisitionInterface](#) for GPS L1 C/A signals.
- class [GpsL1CaPcpsAcquisitionFpga](#)

*This class adapts a PCPS acquisition block off-loaded on an FPGA to an [AcquisitionInterface](#) for GPS L1 C/A signals.*

- class [GpsL1CaPcpsAssistedAcquisition](#)

*This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.*

- class [GpsL1CaPcpsOpenCLAcquisition](#)

*This class adapts an OpenCL PCPS acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.*

- class [GpsL1CaPcpsQuickSyncAcquisition](#)

*This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.*

- class [GpsL1CaPcpsTongAcquisition](#)

*This class adapts a PCPS Tong acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.*

- class [GpsL2MPcpsAcquisition](#)

*This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L2 M signals.*

- class [GpsL2MPcpsAcquisitionFpga](#)

*This class adapts a PCPS acquisition block off-loaded on an FPGA to an [AcquisitionInterface](#) for GPS L2 M signals.*

- class [GpsL5iPcpsAcquisition](#)

*This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L5i signals.*

- class [GpsL5iPcpsAcquisitionFpga](#)

*This class adapts a PCPS acquisition block off-loaded on an FPGA to an [AcquisitionInterface](#) for GPS L5i signals.*

## Typedefs

- using [pcps\\_acquisition\\_fine\\_doppler\\_cc\\_sptr](#) = gnss\_shared\_ptr<[pcps\\_acquisition\\_fine\\_doppler\\_cc](#)>

### 9.1.2.1 Detailed Description

Wrap GNU Radio acquisition blocks with an [AcquisitionInterface](#)

### 9.1.2.2 Typedef Documentation

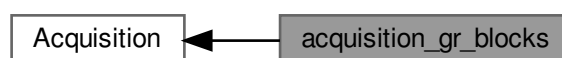
#### 9.1.2.2.1 pcps\_acquisition\_fine\_doppler\_cc\_sptr

```
using pcps_acquisition_fine_doppler_cc_sptr = gnss_shared_ptr<pcps_acquisition_fine_doppler_cc>
```

Definition at line 37 of file [gps\\_l1\\_ca\\_pcps\\_acquisition\\_fine\\_doppler.h](#).

### 9.1.3 acquisition\_gr\_blocks

Collaboration diagram for acquisition\_gr\_blocks:



## Classes

- class [galileo\\_e5a\\_noncoherentIQ\\_acquisition\\_caf\\_cc](#)  
*This class implements a Parallel Code Phase Search Acquisition.*
- class [galileo\\_pcps\\_8ms\\_acquisition\\_cc](#)  
*This class implements a Parallel Code Phase Search Acquisition for Galileo E1 signals with coherent integration time = 8 ms (two codes)*
- class [pcps\\_acquisition](#)  
*This class implements a Parallel Code Phase Search Acquisition.*
- class [pcps\\_acquisition\\_fine\\_doppler\\_cc](#)  
*This class implements a Parallel Code Phase Search Acquisition.*
- class [pcps\\_acquisition\\_fpga](#)  
*This class implements a Parallel Code Phase Search Acquisition that uses the FPGA.*
- class [pcps\\_assisted\\_acquisition\\_cc](#)  
*This class implements a Parallel Code Phase Search Acquisition.*
- class [pcps\\_cccwsr\\_acquisition\\_cc](#)  
*This class implements a Parallel Code Phase Search Acquisition with Coherent [Channel](#) Combining With Sign Recovery scheme.*
- class [pcps\\_opencl\\_acquisition\\_cc](#)  
*This class implements a Parallel Code Phase Search Acquisition.*
- class [pcps\\_quicksync\\_acquisition\\_cc](#)  
*This class implements a Parallel Code Phase Search Acquisition with the implementation of the Sparse QuickSync Algorithm.*
- class [pcps\\_tong\\_acquisition\\_cc](#)  
*This class implements a Parallel Code Phase Search Acquisition with Tong algorithm.*

## Typedefs

- using [galileo\\_e5a\\_noncoherentIQ\\_acquisition\\_caf\\_cc\\_sptr](#) = gnss\_shared\_ptr<[galileo\\_e5a\\_noncoherentIQ\\_acquisition\\_caf\\_cc](#)>
- using [galileo\\_pcps\\_8ms\\_acquisition\\_cc\\_sptr](#) = gnss\_shared\_ptr<[galileo\\_pcps\\_8ms\\_acquisition\\_cc](#)>
- using [pcps\\_acquisition\\_sptr](#) = gnss\_shared\_ptr<[pcps\\_acquisition](#)>
- using [pcps\\_acquisition\\_fine\\_doppler\\_cc\\_sptr](#) = gnss\_shared\_ptr<[pcps\\_acquisition\\_fine\\_doppler\\_cc](#)>
- using [pcps\\_acquisition\\_fpga\\_sptr](#) = std::shared\_ptr<[pcps\\_acquisition\\_fpga](#)>
- using [pcps\\_assisted\\_acquisition\\_cc\\_sptr](#) = gnss\_shared\_ptr<[pcps\\_assisted\\_acquisition\\_cc](#)>
- using [pcps\\_cccwsr\\_acquisition\\_cc\\_sptr](#) = gnss\_shared\_ptr<[pcps\\_cccwsr\\_acquisition\\_cc](#)>
- using [pcps\\_opencl\\_acquisition\\_cc\\_sptr](#) = gnss\_shared\_ptr<[pcps\\_opencl\\_acquisition\\_cc](#)>
- using [pcps\\_quicksync\\_acquisition\\_cc\\_sptr](#) = gnss\_shared\_ptr<[pcps\\_quicksync\\_acquisition\\_cc](#)>
- using [pcps\\_tong\\_acquisition\\_cc\\_sptr](#) = gnss\_shared\_ptr<[pcps\\_tong\\_acquisition\\_cc](#)>

## Functions

- [galileo\\_e5a\\_noncoherentIQ\\_acquisition\\_caf\\_cc\\_sptr](#) **galileo\_e5a\_noncoherentIQ\_make\_acquisition\_caf\_cc** (unsigned int sampled\_ms, unsigned int max\_dwells, unsigned int doppler\_max, int64\_t fs\_in, int samples\_per\_ms, int samples\_per\_code, bool bit\_transition\_flag, bool dump, const std::string &dump\_filename, bool both\_signal\_components\_, int CAF\_window\_hz\_, int Zero\_padding\_, bool enable\_monitor\_output)
- [galileo\\_pcps\\_8ms\\_acquisition\\_cc\\_sptr](#) **galileo\_pcps\_8ms\_make\_acquisition\_cc** (uint32\_t sampled\_ms, uint32\_t max\_dwells, uint32\_t doppler\_max, int64\_t fs\_in, int32\_t samples\_per\_ms, int32\_t samples\_per\_code, bool dump, const std::string &dump\_filename, bool enable\_monitor\_output)
- [pcps\\_acquisition\\_sptr](#) **pcps\_make\_acquisition** (const [Acq\\_Conf](#) &conf\_)
- [pcps\\_acquisition\\_fine\\_doppler\\_cc\\_sptr](#) **pcps\_make\_acquisition\_fine\_doppler\_cc** (const [Acq\\_Conf](#) &conf\_)

- `pcps_acquisition_fpga_sptr` **`pcps_make_acquisition_fpga`** (`Acq_Conf_Fpga` \*conf\_, `uint32_t` acq\_buff\_↵  
num, `std::vector< std::pair< uint32_t, uint32_t > >` &downsampling\_filter\_specs, `uint32_t` &max\_FFT\_size)
- `pcps_assisted_acquisition_cc_sptr` **`pcps_make_assisted_acquisition_cc`** (`int32_t` max\_dwells, `uint32_t` ↵  
sampled\_ms, `int32_t` doppler\_max, `int32_t` doppler\_min, `int64_t` fs\_in, `int32_t` samples\_per\_ms, `bool` dump,  
const `std::string` &dump\_filename, `bool` enable\_monitor\_output)
- `pcps_cccwsr_acquisition_cc_sptr` **`pcps_cccwsr_make_acquisition_cc`** (`uint32_t` sampled\_ms, `uint32_t` ↵  
max\_dwells, `uint32_t` doppler\_max, `int64_t` fs\_in, `int32_t` samples\_per\_ms, `int32_t` samples\_per\_code, `bool`  
dump, const `std::string` &dump\_filename, `bool` enable\_monitor\_output)
- `pcps_openc1_acquisition_cc_sptr` **`pcps_make_openc1_acquisition_cc`** (`uint32_t` sampled\_ms, `uint32_t` ↵  
max\_dwells, `uint32_t` doppler\_max, `int64_t` fs\_in, `int` samples\_per\_ms, `int` samples\_per\_code, `bool` bit\_↵  
\_transition\_flag, `bool` dump, const `std::string` &dump\_filename, `bool` enable\_monitor\_output)
- `pcps_quicksync_acquisition_cc_sptr` **`pcps_quicksync_make_acquisition_cc`** (`uint32_t` folding\_factor,  
`uint32_t` sampled\_ms, `uint32_t` max\_dwells, `uint32_t` doppler\_max, `int64_t` fs\_in, `int32_t` samples\_per\_↵  
\_ms, `int32_t` samples\_per\_code, `bool` bit\_transition\_flag, `bool` dump, const `std::string` &dump\_filename, `bool`  
enable\_monitor\_output)
- `pcps_tong_acquisition_cc_sptr` **`pcps_tong_make_acquisition_cc`** (`uint32_t` sampled\_ms, `uint32_t` ↵  
doppler\_max, `int64_t` fs\_in, `int32_t` samples\_per\_ms, `int32_t` samples\_per\_code, `uint32_t` tong\_init\_↵  
val, `uint32_t` tong\_max\_val, `uint32_t` tong\_max\_dwells, `bool` dump, const `std::string` &dump\_filename, `bool`  
enable\_monitor\_output)

### 9.1.3.1 Detailed Description

GNU Radio processing blocks for GNSS signal acquisition

### 9.1.3.2 Typedef Documentation

#### 9.1.3.2.1 `galileo_e5a_noncoherentIQ_acquisition_caf_cc_sptr`

```
using galileo_e5a_noncoherentIQ_acquisition_caf_cc_sptr = gnss_shared_ptr<galileo_e5a_noncoherentIQ_acquisition_caf_cc>
```

Definition at line 46 of file [galileo\\_e5a\\_noncoherent\\_iq\\_acquisition\\_caf\\_cc.h](#).

#### 9.1.3.2.2 `galileo_pcps_8ms_acquisition_cc_sptr`

```
using galileo_pcps_8ms_acquisition_cc_sptr = gnss_shared_ptr<galileo_pcps_8ms_acquisition_cc>
```

Definition at line 40 of file [galileo\\_pcps\\_8ms\\_acquisition\\_cc.h](#).

#### 9.1.3.2.3 `pcps_acquisition_fine_doppler_cc_sptr`

```
using pcps_acquisition_fine_doppler_cc_sptr = gnss_shared_ptr<pcps_acquisition_fine_doppler_cc>
```

Definition at line 65 of file [pcps\\_acquisition\\_fine\\_doppler\\_cc.h](#).

#### 9.1.3.2.4 `pcps_acquisition_fpga_sptr`

```
using pcps_acquisition_fpga_sptr = std::shared_ptr<pcps_acquisition_fpga>
```

Definition at line 48 of file [pcps\\_acquisition\\_fpga.h](#).

#### 9.1.3.2.5 pcps\_acquisition\_sptr

```
using pcps_acquisition_sptr = gnss_shared_ptr<pcps_acquisition>
```

Definition at line 82 of file [pcps\\_acquisition.h](#).

#### 9.1.3.2.6 pcps\_assisted\_acquisition\_cc\_sptr

```
using pcps_assisted_acquisition_cc_sptr = gnss_shared_ptr<pcps_assisted_acquisition_cc>
```

Definition at line 56 of file [pcps\\_assisted\\_acquisition\\_cc.h](#).

#### 9.1.3.2.7 pcps\_cccwsr\_acquisition\_cc\_sptr

```
using pcps_cccwsr_acquisition_cc_sptr = gnss_shared_ptr<pcps_cccwsr_acquisition_cc>
```

Definition at line 45 of file [pcps\\_cccwsr\\_acquisition\\_cc.h](#).

#### 9.1.3.2.8 pcps\_openc1\_acquisition\_cc\_sptr

```
using pcps_openc1_acquisition_cc_sptr = gnss_shared_ptr<pcps_openc1_acquisition_cc>
```

Definition at line 63 of file [pcps\\_openc1\\_acquisition\\_cc.h](#).

#### 9.1.3.2.9 pcps\_quicksync\_acquisition\_cc\_sptr

```
using pcps_quicksync_acquisition_cc_sptr = gnss_shared_ptr<pcps_quicksync_acquisition_cc>
```

Definition at line 62 of file [pcps\\_quicksync\\_acquisition\\_cc.h](#).

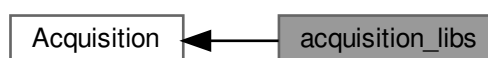
#### 9.1.3.2.10 pcps\_tong\_acquisition\_cc\_sptr

```
using pcps_tong_acquisition_cc_sptr = gnss_shared_ptr<pcps_tong_acquisition_cc>
```

Definition at line 59 of file [pcps\\_tong\\_acquisition\\_cc.h](#).

### 9.1.4 acquisition\_libs

Collaboration diagram for acquisition\_libs:



### Classes

- class [Acq\\_Conf](#)
- class [Acq\\_Conf\\_Fpga](#)
- class [Fpga\\_Acquisition](#)

*Class that implements carrier wipe-off and correlators.*

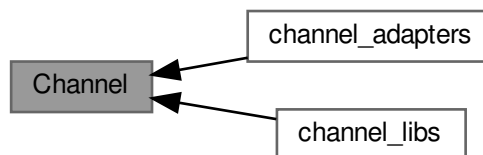
#### 9.1.4.1 Detailed Description

Library with utilities for GNSS signal acquisition

Library with utilities for GNSS signal acquisition

## 9.2 Channel

Collaboration diagram for Channel:



### Topics

- [channel\\_adapters](#)
- [channel\\_libs](#)

#### 9.2.1 Detailed Description

Classes containing a GNSS channel.

#### 9.2.2 channel\_adapters

Collaboration diagram for channel\_adapters:



## Classes

- class [Channel](#)

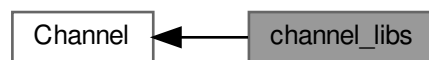
*This class represents a GNSS channel. It wraps an [AcquisitionInterface](#), a [TrackingInterface](#) and a [TelemetryDecoderInterface](#), and handles their interaction through a Finite State Machine.*

### 9.2.2.1 Detailed Description

Classes that wrap an [AcquisitionInterface](#), a [TrackingInterface](#) and a [TelemetryDecoderInterface](#), and handles their interaction.

### 9.2.3 channel\_libs

Collaboration diagram for channel\_libs:



## Classes

- class [ChannelFsm](#)

*This class implements a State Machine for channel.*

- class [channel\\_msg\\_receiver\\_cc](#)

*GNU Radio block that receives asynchronous channel messages from acquisition and tracking blocks.*

## Typedefs

- using [channel\\_msg\\_receiver\\_cc\\_sptr](#) = `gnss_shared_ptr<channel\_msg\_receiver\_cc>`

## Functions

- `channel_msg_receiver_cc_sptr channel_msg_receiver_make_cc (std::shared_ptr< ChannelFsm > channel_fsm, bool repeat)`

### 9.2.3.1 Detailed Description

Library with utilities for a GNSS [Channel](#).



### 9.2.3.2 Typedef Documentation

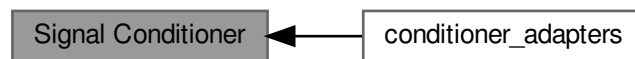
#### 9.2.3.2.1 channel\_msg\_receiver\_cc\_sptr

```
using channel_msg_receiver_cc_sptr = gnss_shared_ptr<channel_msg_receiver_cc>
```

Definition at line 33 of file [channel\\_msg\\_receiver\\_cc.h](#).

## 9.3 Signal Conditioner

Collaboration diagram for Signal Conditioner:



### Topics

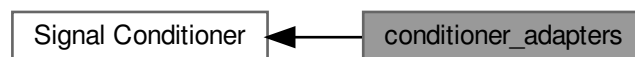
- [conditioner\\_adapters](#)

### 9.3.1 Detailed Description

Signal Conditioner wrapper block

### 9.3.2 conditioner\_adapters

Collaboration diagram for conditioner\_adapters:



## Classes

- class [ArraySignalConditioner](#)

*This class wraps blocks to change data\_type\_adapter, input\_filter and resampler to be applied to the input flow of sampled signal.*

- class [SignalConditioner](#)

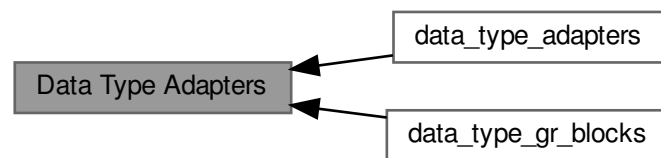
*This class wraps blocks to change data\_type\_adapter, input\_filter and resampler to be applied to the input flow of sampled signal.*

### 9.3.2.1 Detailed Description

Wrap a Signal Conditioner with a [GNSSBlockInterface](#)

## 9.4 Data Type Adapters

Collaboration diagram for Data Type Adapters:



## Topics

- [data\\_type\\_adapters](#)
- [data\\_type\\_gr\\_blocks](#)

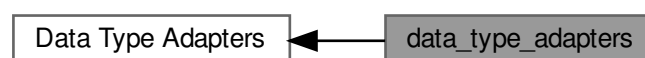
### 9.4.1 Detailed Description

Classes for data type conversion

Classes for data type conversion

### 9.4.2 data\_type\_adapters

Collaboration diagram for data\_type\_adapters:



**Classes**

- class [ByteToShort](#)  
*Adapts an 8-bits sample stream (IF) to a short int stream (IF)*
- class [CshortToGrComplex](#)  
*Adapts an 16-bits complex sample stream to a float complex stream.*
- class [lbyteToCbyte](#)
- class [lbyteToComplex](#)  
*Adapts an I/Q interleaved byte integer sample stream to a gr\_complex (float) stream.*
- class [lbyteToCshort](#)  
*Adapts a short integer (16 bits) interleaved sample stream into a std::complex<short> stream.*
- class [lshortToComplex](#)  
*Adapts an I/Q interleaved short integer sample stream to a gr\_complex (float) stream.*
- class [lshortToCshort](#)  
*Adapts a short integer (16 bits) interleaved sample stream into a std::complex<short> stream.*

**9.4.2.1 Detailed Description**

Wrap GNU Radio data type adapter blocks with a [GNSSBlockInterface](#)

Wrap GNU Radio data type adapter blocks with a [GNSSBlockInterface](#)

**9.4.3 data\_type\_gr\_blocks**

Collaboration diagram for data\_type\_gr\_blocks:

**Classes**

- class [cshort\\_to\\_gr\\_complex](#)  
*This class adapts a short (16-bits) interleaved sample stream into a std::complex<float> stream.*
- class [interleaved\\_byte\\_to\\_complex\\_byte](#)  
*This class adapts an 8-bits interleaved sample stream into a 16-bits complex stream (std::complex<unsigned char>)*
- class [interleaved\\_byte\\_to\\_complex\\_short](#)  
*This class adapts a short (16-bits) interleaved sample stream into a std::complex<short> stream.*
- class [interleaved\\_short\\_to\\_complex\\_short](#)  
*This class adapts a short (16-bits) interleaved sample stream into a std::complex<short> stream.*

## Typedefs

- using `csshort_to_gr_complex_sptr` = `gnss_shared_ptr<csshort_to_gr_complex>`
- using `interleaved_byte_to_complex_byte_sptr` = `gnss_shared_ptr<interleaved_byte_to_complex_byte>`
- using `interleaved_byte_to_complex_short_sptr` = `gnss_shared_ptr<interleaved_byte_to_complex_short>`
- using `interleaved_short_to_complex_short_sptr` = `gnss_shared_ptr<interleaved_short_to_complex_short>`

## Functions

- `csshort_to_gr_complex_sptr` **make\_cshort\_to\_gr\_complex** ()
- `interleaved_byte_to_complex_byte_sptr` **make\_interleaved\_byte\_to\_complex\_byte** ()
- `interleaved_byte_to_complex_short_sptr` **make\_interleaved\_byte\_to\_complex\_short** ()
- `interleaved_short_to_complex_short_sptr` **make\_interleaved\_short\_to\_complex\_short** ()

### 9.4.3.1 Detailed Description

GNU Radio Blocks for data type conversion

### 9.4.3.2 Typedef Documentation

#### 9.4.3.2.1 cshort\_to\_gr\_complex\_sptr

```
using cshort_to_gr_complex_sptr = gnss_shared_ptr<csshort_to_gr_complex>
```

Definition at line 32 of file [csshort\\_to\\_gr\\_complex.h](#).

#### 9.4.3.2.2 interleaved\_byte\_to\_complex\_byte\_sptr

```
using interleaved_byte_to_complex_byte_sptr = gnss_shared_ptr<interleaved_byte_to_complex_byte>
```

Definition at line 32 of file [interleaved\\_byte\\_to\\_complex\\_byte.h](#).

#### 9.4.3.2.3 interleaved\_byte\_to\_complex\_short\_sptr

```
using interleaved_byte_to_complex_short_sptr = gnss_shared_ptr<interleaved_byte_to_complex_short>
```

Definition at line 32 of file [interleaved\\_byte\\_to\\_complex\\_short.h](#).

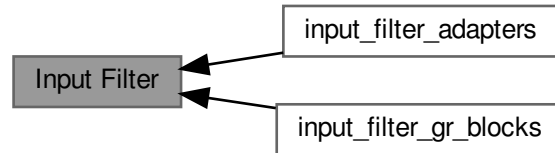
#### 9.4.3.2.4 interleaved\_short\_to\_complex\_short\_sptr

```
using interleaved_short_to_complex_short_sptr = gnss_shared_ptr<interleaved_short_to_complex_short>
```

Definition at line 31 of file [interleaved\\_short\\_to\\_complex\\_short.h](#).

## 9.5 Input Filter

Collaboration diagram for Input Filter:



### Topics

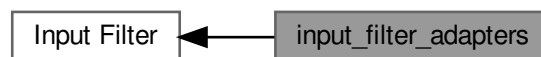
- [input\\_filter\\_adapters](#)
- [input\\_filter\\_gr\\_blocks](#)

### 9.5.1 Detailed Description

Classes for input signal filtering

### 9.5.2 input\_filter\_adapters

Collaboration diagram for input\_filter\_adapters:



### Classes

- class [BeamformerFilter](#)  
*Interface of an adapter of a digital beamformer block to a [GNSSBlockInterface](#).*
- class [FirFilter](#)  
*This class adapts a GNU Radio `gr_fir_filter` designed with `pm_remez`.*
- class [FreqXlatingFirFilter](#)  
*This class adapts a `gnuradio gr_freq_xlating_fir_filter` designed with `pm_remez`.*
- class [NotchFilter](#)
- class [NotchFilterLite](#)
- class [PulseBlankingFilter](#)

### 9.5.2.1 Detailed Description

Classes that wrap GNU Radio input filters with a [GNSSBlockInterface](#)

### 9.5.3 input\_filter\_gr\_blocks

Collaboration diagram for input\_filter\_gr\_blocks:



#### Classes

- class [beamformer](#)  
*This class implements a real-time software-defined spatial filter using the CTTC GNSS experimental antenna array input and a set of dynamically reloadable weights.*
- class [Notch](#)  
*This class implements a real-time software-defined multi state notch filter.*
- class [NotchLite](#)  
*This class implements a real-time software-defined multi state notch filter light version.*
- class [pulse\\_blanking\\_cc](#)

#### Typedefs

- using [beamformer\\_sptr](#) = gnss\_shared\_ptr<[beamformer](#)>
- using [notch\\_sptr](#) = gnss\_shared\_ptr<[Notch](#)>
- using [notch\\_lite\\_sptr](#) = gnss\_shared\_ptr<[NotchLite](#)>
- using [pulse\\_blanking\\_cc\\_sptr](#) = gnss\_shared\_ptr<[pulse\\_blanking\\_cc](#)>

#### Functions

- [beamformer\\_sptr](#) **make\_beamformer\_sptr** ()
- [notch\\_sptr](#) **make\_notch\_filter** (float pfa, float p\_c\_factor, int32\_t length, int32\_t n\_segments\_est, int32\_t n\_segments\_reset)
- [notch\\_lite\\_sptr](#) **make\_notch\_filter\_lite** (float p\_c\_factor, float pfa, int32\_t length, int32\_t n\_segments\_est, int32\_t n\_segments\_reset, int32\_t n\_segments\_coeff)
- [pulse\\_blanking\\_cc\\_sptr](#) **make\_pulse\_blanking\_cc** (float pfa, int32\_t length, int32\_t n\_segments\_est, int32\_t n\_segments\_reset)

#### Variables

- const int [GNSS\\_SDR\\_BEAMFORMER\\_CHANNELS](#) = 8

### 9.5.3.1 Detailed Description

GNU Radio blocks implementing input filters,

### 9.5.3.2 Typedef Documentation

#### 9.5.3.2.1 beamformer\_sptr

```
using beamformer_sptr = gnss_shared_ptr<beamformer>
```

Definition at line 33 of file [beamformer.h](#).

#### 9.5.3.2.2 notch\_lite\_sptr

```
using notch_lite_sptr = gnss_shared_ptr<NotchLite>
```

Definition at line 36 of file [notch\\_lite\\_cc.h](#).

#### 9.5.3.2.3 notch\_sptr

```
using notch_sptr = gnss_shared_ptr<Notch>
```

Definition at line 36 of file [notch\\_cc.h](#).

#### 9.5.3.2.4 pulse\_blanking\_cc\_sptr

```
using pulse_blanking_cc_sptr = gnss_shared_ptr<pulse_blanking_cc>
```

Definition at line 36 of file [pulse\\_blanking\\_cc.h](#).

### 9.5.3.3 Variable Documentation

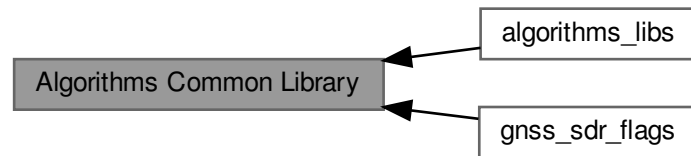
#### 9.5.3.3.1 GNSS\_SDR\_BEAMFORMER\_CHANNELS

```
const int GNSS_SDR_BEAMFORMER_CHANNELS = 8
```

Definition at line 37 of file [beamformer.h](#).

## 9.6 Algorithms Common Library

Collaboration diagram for Algorithms Common Library:



### Topics

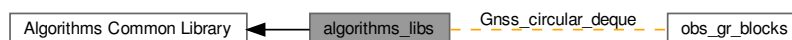
- [algorithms\\_libs](#)
- [gnss\\_sdr\\_flags](#)

### 9.6.1 Detailed Description

Common utilities for the GNSS receiver.

### 9.6.2 algorithms\_libs

Collaboration diagram for algorithms\_libs:



### Classes

- class [byte\\_x2\\_to\\_complex\\_byte](#)  
This class adapts two signed char streams into a `std::complex<signed char>` stream.
- class [complex\\_byte\\_to\\_float\\_x2](#)  
This class adapts a `std::complex<signed char>` stream into two 16-bits (short) streams.
- class [complex\\_float\\_to\\_complex\\_byte](#)  
This class adapts a `gr_complex` stream into a `std::complex<signed char>` stream.
- class [conjugate\\_cc](#)  
This class adapts a `std::complex<short>` stream into two 32-bits (float) streams.
- class [conjugate\\_ic](#)  
This class adapts a `std::complex<short>` stream into two 32-bits (float) streams.



- class `conjugate_sc`  
*This class adapts a `std::complex<short>` stream into two 32-bits (float) streams.*
- class `cshort_to_float_x2`  
*This class adapts a `std::complex<short>` stream into two 32-bits (float) streams.*
- class `Gnss_circular_deque< T >`
- class `Pass_Through`  
*This class implements a block that connects input and output (does nothing)*
- class `short_x2_to_cshort`  
*This class adapts two short streams into a `std::complex<short>` stream.*
- class `TrackingCmd`

## Typedefs

- using `byte_x2_to_complex_byte_sptr` = `gnss_shared_ptr<byte_x2_to_complex_byte>`
- using `complex_byte_to_float_x2_sptr` = `gnss_shared_ptr<complex_byte_to_float_x2>`
- using `complex_float_to_complex_byte_sptr` = `gnss_shared_ptr<complex_float_to_complex_byte>`
- using `conjugate_cc_sptr` = `gnss_shared_ptr<conjugate_cc>`
- using `conjugate_ic_sptr` = `gnss_shared_ptr<conjugate_ic>`
- using `conjugate_sc_sptr` = `gnss_shared_ptr<conjugate_sc>`
- using `cshort_to_float_x2_sptr` = `gnss_shared_ptr<cshort_to_float_x2>`
- using `item_type_converter_t` = `std::function<void(void *, const void *, uint32_t)>`
- using `short_x2_to_cshort_sptr` = `gnss_shared_ptr<short_x2_to_cshort>`

## Functions

- void `beidou_b1i_code_gen_int` (`own::span< int32_t > dest`, `int32_t prn`, `uint32_t chip_shift`)  
*Generates int32\_t GPS L1 C/A code for the desired SV ID and code shift.*
- void `beidou_b1i_code_gen_float` (`own::span< float > dest`, `int32_t prn`, `uint32_t chip_shift`)  
*Generates float GPS L1 C/A code for the desired SV ID and code shift.*
- void `beidou_b1i_code_gen_complex` (`own::span< std::complex< float > > dest`, `int32_t prn`, `uint32_t chip_shift`)  
*Generates complex GPS L1 C/A code for the desired SV ID and code shift.*
- void `beidou_b1i_code_gen_complex_sampled` (`own::span< std::complex< float > > dest`, `uint32_t prn`, `int32_t sampling_freq`, `uint32_t chip_shift`)  
*Generates complex GPS L1 C/A code for the desired SV ID and code shift, and sampled to specific sampling frequency.*
- void `beidou_b3i_code_gen_int` (`own::span< int > dest`, `int32_t prn`, `uint32_t chip_shift`)  
*Generates int BeiDou B3I code for the desired SV ID and code shift.*
- void `beidou_b3i_code_gen_float` (`own::span< float > dest`, `int32_t prn`, `uint32_t chip_shift`)  
*Generates float BeiDou B3I code for the desired SV ID and code shift.*
- void `beidou_b3i_code_gen_complex` (`own::span< std::complex< float > > dest`, `int32_t prn`, `uint32_t chip_shift`)  
*Generates complex BeiDou B3I code for the desired SV ID and code shift.*
- void `beidou_b3i_code_gen_complex_sampled` (`own::span< std::complex< float > > dest`, `uint32_t prn`, `int sampling_freq`, `uint32_t chip_shift`)  
*Generates complex BeiDou B3I code for the desired SV ID and code shift, and sampled to specific sampling frequency.*
- `byte_x2_to_complex_byte_sptr` `make_byte_x2_to_complex_byte` ()
- `complex_byte_to_float_x2_sptr` `make_complex_byte_to_float_x2` ()
- `complex_float_to_complex_byte_sptr` `make_complex_float_to_complex_byte` ()
- `conjugate_cc_sptr` `make_conjugate_cc` ()
- `conjugate_ic_sptr` `make_conjugate_ic` ()

- conjugate\_sc\_sptr **make\_conjugate\_sc** ()
- cshort\_to\_float\_x2\_sptr **make\_cshort\_to\_float\_x2** ()
- void [galileo\\_e1\\_code\\_gen\\_sinboc11\\_float](#) (own::span< float > dest, const std::array< char, 3 > &signal\_id, uint32\_t prn)  
*This function generates Galileo E1 code (can select E1B or E1C sinboc).*
- void [galileo\\_e1\\_code\\_gen\\_float\\_sampled](#) (own::span< float > dest, const std::array< char, 3 > &signal\_id, bool cboc, uint32\_t prn, int32\_t sampling\_freq, uint32\_t chip\_shift, bool secondary\_flag)  
*This function generates Galileo E1 code (can select E1B or E1C, cboc or sinboc and the sample frequency sampling\_freq).*
- void [galileo\\_e1\\_code\\_gen\\_float\\_sampled](#) (own::span< float > dest, const std::array< char, 3 > &signal\_id, bool cboc, uint32\_t prn, int32\_t sampling\_freq, uint32\_t chip\_shift)  
*This function generates Galileo E1 code (can select E1B or E1C, cboc or sinboc and the sample frequency sampling\_freq).*
- void [galileo\\_e1\\_code\\_gen\\_complex\\_sampled](#) (own::span< std::complex< float > > dest, const std::array< char, 3 > &signal\_id, bool cboc, uint32\_t prn, int32\_t sampling\_freq, uint32\_t chip\_shift, bool secondary\_flag)  
*This function generates Galileo E1 code (can select E1B or E1C, cboc or sinboc and the sample frequency sampling\_freq).*
- void [galileo\\_e1\\_code\\_gen\\_complex\\_sampled](#) (own::span< std::complex< float > > dest, const std::array< char, 3 > &signal\_id, bool cboc, uint32\_t prn, int32\_t sampling\_freq, uint32\_t chip\_shift)  
*galileo\_e1\_code\_gen\_complex\_sampled without secondary\_flag for backward compatibility.*
- void [galileo\\_e5\\_a\\_code\\_gen\\_complex\\_primary](#) (own::span< std::complex< float > > dest, int32\_t prn, const std::array< char, 3 > &signal\_id)  
*Generates Galileo E5a code at 1 sample/chip.*
- void [galileo\\_e5\\_a\\_code\\_gen\\_complex\\_sampled](#) (own::span< std::complex< float > > dest, uint32\_t prn, const std::array< char, 3 > &signal\_id, int32\_t sampling\_freq, uint32\_t chip\_shift)  
*Generates Galileo E5a complex code, shifted to the desired chip and sampled at a frequency sampling\_freq.*
- void [galileo\\_e5\\_b\\_code\\_gen\\_complex\\_primary](#) (own::span< std::complex< float > > dest, int32\_t prn, const std::array< char, 3 > &signal\_id)  
*Generates Galileo E5b code at 1 sample/chip.*
- void [galileo\\_e5\\_b\\_code\\_gen\\_complex\\_sampled](#) (own::span< std::complex< float > > dest, uint32\_t prn, const std::array< char, 3 > &signal\_id, int32\_t sampling\_freq, uint32\_t chip\_shift)  
*Generates Galileo E5b complex code, shifted to the desired chip and sampled at a frequency sampling\_freq.*
- void [galileo\\_e6\\_b\\_code\\_gen\\_complex\\_primary](#) (own::span< std::complex< float > > dest, int32\_t prn)  
*Generates Galileo E6B code at 1 sample/chip.*
- void [galileo\\_e6\\_b\\_code\\_gen\\_float\\_primary](#) (own::span< float > dest, int32\_t prn)  
*Generates Galileo E6B code at 1 sample/chip.*
- void [galileo\\_e6\\_b\\_code\\_gen\\_complex\\_sampled](#) (own::span< std::complex< float > > dest, uint32\_t prn, int32\_t sampling\_freq, uint32\_t chip\_shift)  
*Generates Galileo E6B complex code, shifted to the desired chip and sampled at a frequency sampling\_freq.*
- void [galileo\\_e6\\_c\\_code\\_gen\\_complex\\_primary](#) (own::span< std::complex< float > > dest, int32\_t prn)  
*Generates Galileo E6C codes at 1 sample/chip.*
- void [galileo\\_e6\\_c\\_code\\_gen\\_float\\_primary](#) (own::span< float > dest, int32\_t prn)  
*Generates Galileo E6C codes at 1 sample/chip.*
- void [galileo\\_e6\\_c\\_code\\_gen\\_complex\\_sampled](#) (own::span< std::complex< float > > dest, uint32\_t prn, int32\_t sampling\_freq, uint32\_t chip\_shift)  
*Generates Galileo E6C complex codes, shifted to the desired chip and sampled at a frequency sampling\_freq.*
- void [galileo\\_e6\\_c\\_secondary\\_code\\_gen\\_complex](#) (own::span< std::complex< float > > dest, int32\_t prn)  
*Generates Galileo E6C secondary codes at 1 sample/chip.*
- void [galileo\\_e6\\_c\\_secondary\\_code\\_gen\\_float](#) (own::span< float > dest, int32\_t prn)  
*Generates Galileo E6C secondary codes at 1 sample/chip.*
- std::string [galileo\\_e6\\_c\\_secondary\\_code](#) (int32\_t prn)  
*Generates a string with Galileo E6C secondary codes at 1 sample/chip.*
- arma::mat [Skew\\_symmetric](#) (const arma::vec &a)

- Calculates skew-symmetric matrix.*
- double **WGS84\_g0** (double Lat\_rad)
- double **WGS84\_geocentric\_radius** (double Lat\_geodetic\_rad)
- int **topocent** (double \*Az, double \*El, double \*D, const arma::vec &x, const arma::vec &dx)
  - Transformation of vector dx into topocentric coordinate system with origin at x Inputs: x - vector origin coordinates (in ECEF system [X; Y; Z;]) dx - vector ([dX; dY; dZ;]).*
- int **togeod** (double \*dphi, double \*dlambda, double \*h, double a, double finv, double X, double Y, double Z)
  - Subroutine to calculate geodetic coordinates latitude, longitude, height given Cartesian coordinates X,Y,Z, and reference ellipsoid values semi-major axis (a) and the inverse of flattening (finv).*
- arma::vec **Gravity\_ECEF** (const arma::vec &r\_eb\_e)
  - Calculates acceleration due to gravity resolved about ECEF-frame.*
- arma::vec **cart2geo** (const arma::vec &XYZ, int ellipsoid\_selection)
  - Conversion of Cartesian coordinates (X,Y,Z) to geographical coordinates (latitude, longitude, h) on a selected reference ellipsoid.*
- arma::vec **LLH\_to\_deg** (const arma::vec &LLH)
- double **degtorad** (double angleInDegrees)
- double **radtodeg** (double angleInRadians)
- double **mstoknotsh** (double MetersPerSeconds)
- double **mstokph** (double MetersPerSeconds)
- arma::vec **CTM\_to\_Euler** (const arma::mat &C)
- arma::mat **Euler\_to\_CTM** (const arma::vec &eul)
- void **ECEF\_to\_Geo** (const arma::vec &r\_eb\_e, const arma::vec &v\_eb\_e, const arma::mat &C\_b\_e, arma::vec &LLH, arma::vec &v\_eb\_n, arma::mat &C\_b\_n)
- void **Geo\_to\_ECEF** (const arma::vec &LLH, const arma::vec &v\_eb\_n, const arma::mat &C\_b\_n, arma::vec &r\_eb\_e, arma::vec &v\_eb\_e, arma::mat &C\_b\_e)
  - From Geographic to ECEF coordinates.*
- void **pv\_Geo\_to\_ECEF** (double L\_b, double lambda\_b, double h\_b, const arma::vec &v\_eb\_n, arma::vec &r\_eb\_e, arma::vec &v\_eb\_e)
  - Converts curvilinear to Cartesian position and velocity resolving axes from NED to ECEF This function created 11/4/2012 by Paul Groves.*
- double **great\_circle\_distance** (double lat1, double lon1, double lat2, double lon2)
  - The Haversine formula determines the great-circle distance between two points on a sphere given their longitudes and latitudes.*
- void **cart2utm** (const arma::vec &r\_eb\_e, int zone, arma::vec &r\_enu)
  - Transformation of ECEF (X,Y,Z) to (E,N,U) in UTM, zone 'zone'.*
- int **findUtmZone** (double latitude\_deg, double longitude\_deg)
  - Function finds the UTM zone number for given longitude and latitude.*
- double **clsin** (const arma::colvec &ar, int degree, double argument)
  - Clenshaw summation of sinus of argument.*
- void **clklsin** (const arma::colvec &ar, int degree, double arg\_real, double arg\_imag, double \*re, double \*im)
  - Clenshaw summation of sinus with complex argument.*
- void **glonass\_l1\_ca\_code\_gen\_complex** (own::span< std::complex< float > > dest, uint32\_t chip\_shift)
  - Generates complex GLONASS L1 C/A code for the desired SV ID and code shift.*
- void **glonass\_l1\_ca\_code\_gen\_complex\_sampled** (own::span< std::complex< float > > dest, int32\_t sampling\_freq, uint32\_t chip\_shift)
  - Generates complex GLONASS L1 C/A code for the desired SV ID and code shift, and sampled to specific sampling frequency.*
- void **glonass\_l2\_ca\_code\_gen\_complex** (own::span< std::complex< float > > dest, uint32\_t chip\_shift)
  - Generates complex GLONASS L2 C/A code for the desired SV ID and code shift.*
- void **glonass\_l2\_ca\_code\_gen\_complex\_sampled** (own::span< std::complex< float > > dest, int32\_t sampling\_freq, uint32\_t chip\_shift)
  - Generates complex GLONASS L2 C/A code for the desired SV ID and code shift, and sampled to specific sampling frequency.*

- bool **gnss\_sdr\_create\_directory** (const std::string &foldername)
- void **complex\_exp\_gen** (own::span< std::complex< float > > dest, double freq, double sampling\_freq)  
*This function generates a complex exponential in dest.*
- void **complex\_exp\_gen\_conj** (own::span< std::complex< float > > dest, double freq, double sampling\_freq)  
*This function generates a conjugate complex exponential in dest.*
- void **hex\_to\_binary\_converter** (own::span< int32\_t > dest, char from)  
*This function makes a conversion from hex (the input is a char) to binary (the output are 4 ints with +1 or -1 values).*
- std::string **hex\_to\_binary\_string** (char from)  
*This function makes a conversion from hex (the input is a char) to binary (the output is a string of 4 char with 0 or 1 values).*
- void **resampler** (const own::span< float > from, own::span< float > dest, float fs\_in, float fs\_out)  
*This function resamples a sequence of float values.*
- void **resampler** (own::span< const std::complex< float > > from, own::span< std::complex< float > > dest, float fs\_in, float fs\_out)  
*This function resamples a sequence of complex values.*
- void **gps\_l2c\_m\_code\_gen\_complex** (own::span< std::complex< float > > dest, uint32\_t prn)  
*Generates complex GPS L2C M code for the desired SV ID.*
- void **gps\_l2c\_m\_code\_gen\_float** (own::span< float > dest, uint32\_t prn)  
*Generates float GPS L2C M code for the desired SV ID.*
- void **gps\_l2c\_m\_code\_gen\_complex\_sampled** (own::span< std::complex< float > > dest, uint32\_t prn, int32\_t sampling\_freq)  
*Generates complex GPS L2C M code for the desired SV ID, and sampled to specific sampling frequency.*
- void **gps\_l5i\_code\_gen\_complex** (own::span< std::complex< float > > dest, uint32\_t prn)  
*Generates complex GPS L5I code for the desired SV ID.*
- void **gps\_l5i\_code\_gen\_float** (own::span< float > dest, uint32\_t prn)  
*Generates real GPS L5I code for the desired SV ID.*
- void **gps\_l5q\_code\_gen\_complex** (own::span< std::complex< float > > dest, uint32\_t prn)  
*Generates complex GPS L5Q code for the desired SV ID.*
- void **gps\_l5q\_code\_gen\_float** (own::span< float > dest, uint32\_t prn)  
*Generates real GPS L5Q code for the desired SV ID.*
- void **gps\_l5i\_code\_gen\_complex\_sampled** (own::span< std::complex< float > > dest, uint32\_t prn, int32\_t sampling\_freq)  
*Generates complex GPS L5I code for the desired SV ID, and sampled to specific sampling frequency.*
- void **gps\_l5q\_code\_gen\_complex\_sampled** (own::span< std::complex< float > > dest, uint32\_t prn, int32\_t sampling\_freq)  
*Generates complex GPS L5Q code for the desired SV ID, and sampled to specific sampling frequency.*
- void **gps\_l1\_ca\_code\_gen\_int** (own::span< int32\_t > dest, int32\_t prn, uint32\_t chip\_shift)  
*Generates int GPS L1 C/A code for the desired SV ID and code shift.*
- void **gps\_l1\_ca\_code\_gen\_float** (own::span< float > dest, int32\_t prn, uint32\_t chip\_shift)  
*Generates float GPS L1 C/A code for the desired SV ID and code shift.*
- void **gps\_l1\_ca\_code\_gen\_complex** (own::span< std::complex< float > > dest, int32\_t prn, uint32\_t chip\_shift)  
*Generates complex GPS L1 C/A code for the desired SV ID and code shift.*
- void **gps\_l1\_ca\_code\_gen\_complex\_sampled** (own::span< std::complex< float > > dest, uint32\_t prn, int32\_t sampling\_freq, uint32\_t chip\_shift)  
*Generates complex GPS L1 C/A code for the desired SV ID and code shift, and sampled to specific sampling frequency.*
- bool **item\_type\_valid** (const std::string &item\_type)  
*Check if a string is a valid item type.*
- size\_t **item\_type\_size** (const std::string &item\_type)  
*Return the size of the given item type, or zero if unknown.*
- bool **item\_type\_is\_complex** (const std::string &item\_type)

- Determine if an item\_type is complex.*
- `item_type_converter_t make_vector_converter` (const std::string &input\_type, const std::string &output\_type)  
*Create a function to convert an array of input\_type to an array of output\_type.*
- `short_x2_to_cshort_sptr make_short_x2_to_cshort` ()
- `Gnss_circular_deque< T >::Gnss_circular_deque` ()  
*Default constructor.*
- `Gnss_circular_deque< T >::Gnss_circular_deque` (unsigned int max\_size, unsigned int nchann)  
*nchann = number of channels; max\_size = channel capacity*
- unsigned int `Gnss_circular_deque< T >::size` (unsigned int ch) const  
*Returns the number of available elements in a channel.*
- T & `Gnss_circular_deque< T >::back` (unsigned int ch)  
*Returns a reference to the last element in the deque.*
- T & `Gnss_circular_deque< T >::front` (unsigned int ch)  
*Returns a reference to the first element in the deque.*
- T & `Gnss_circular_deque< T >::at` (unsigned int ch, unsigned int pos)  
*Returns a reference to an element with bound checking.*
- const T & `Gnss_circular_deque< T >::get` (unsigned int ch, unsigned int pos) const  
*Returns a const reference to an element without bound checking.*
- void `Gnss_circular_deque< T >::clear` (unsigned int ch)  
*Removes all the elements of the deque (Sets size to 0). Capacity is not modified.*
- void `Gnss_circular_deque< T >::reset` (unsigned int max\_size, unsigned int nchann)  
*Removes all the elements in all the channels. Re-sets the number of channels and their capacity.*
- void `Gnss_circular_deque< T >::reset` ()  
*Removes all the channels (Sets nchann to 0)*
- void `Gnss_circular_deque< T >::pop_front` (unsigned int ch)  
*Removes the first element of the deque.*
- void `Gnss_circular_deque< T >::push_back` (unsigned int ch, const T &new\_data)  
*Inserts an element at the end of the deque.*

### 9.6.2.1 Detailed Description

Common utilities for GNSS algorithms.

### 9.6.2.2 Typedef Documentation

#### 9.6.2.2.1 byte\_x2\_to\_complex\_byte\_sptr

```
using byte_x2_to_complex_byte_sptr = gnss_shared_ptr<byte_x2_to_complex_byte>
```

Definition at line 33 of file [byte\\_x2\\_to\\_complex\\_byte.h](#).

#### 9.6.2.2.2 complex\_byte\_to\_float\_x2\_sptr

```
using complex_byte_to_float_x2_sptr = gnss_shared_ptr<complex_byte_to_float_x2>
```

Definition at line 33 of file [complex\\_byte\\_to\\_float\\_x2.h](#).

#### 9.6.2.2.3 `complex_float_to_complex_byte_sptr`

```
using complex_float_to_complex_byte_sptr = gnss_shared_ptr<complex_float_to_complex_byte>
```

Definition at line 33 of file [complex\\_float\\_to\\_complex\\_byte.h](#).

#### 9.6.2.2.4 `conjugate_cc_sptr`

```
using conjugate_cc_sptr = gnss_shared_ptr<conjugate_cc>
```

Definition at line 33 of file [conjugate\\_cc.h](#).

#### 9.6.2.2.5 `conjugate_ic_sptr`

```
using conjugate_ic_sptr = gnss_shared_ptr<conjugate_ic>
```

Definition at line 33 of file [conjugate\\_ic.h](#).

#### 9.6.2.2.6 `conjugate_sc_sptr`

```
using conjugate_sc_sptr = gnss_shared_ptr<conjugate_sc>
```

Definition at line 33 of file [conjugate\\_sc.h](#).

#### 9.6.2.2.7 `cshort_to_float_x2_sptr`

```
using cshort_to_float_x2_sptr = gnss_shared_ptr<cshort_to_float_x2>
```

Definition at line 33 of file [cshort\\_to\\_float\\_x2.h](#).

#### 9.6.2.2.8 `item_type_converter_t`

```
using item_type_converter_t = std::function<void(void *, const void *, uint32_t)>
```

Definition at line 34 of file [item\\_type\\_helpers.h](#).

#### 9.6.2.2.9 `short_x2_to_cshort_sptr`

```
using short_x2_to_cshort_sptr = gnss_shared_ptr<short_x2_to_cshort>
```

Definition at line 33 of file [short\\_x2\\_to\\_cshort.h](#).

### 9.6.2.3 Function Documentation

#### 9.6.2.3.1 at()

```
template<class T>
T & Gnss_circular_deque< T >::at (
    unsigned int ch,
    unsigned int pos)
```

Returns a reference to an element with bound checking.

Definition at line 88 of file [gnss\\_circular\\_deque.h](#).

#### 9.6.2.3.2 back()

```
template<class T>
T & Gnss_circular_deque< T >::back (
    unsigned int ch)
```

Returns a reference to the last element in the deque.

Definition at line 74 of file [gnss\\_circular\\_deque.h](#).

#### 9.6.2.3.3 beidou\_b1i\_code\_gen\_complex()

```
void beidou_b1i_code_gen_complex (
    own::span< std::complex< float > > dest,
    int32_t prn,
    uint32_t chip_shift)
```

Generates complex GPS L1 C/A code for the desired SV ID and code shift.

#### 9.6.2.3.4 beidou\_b1i\_code\_gen\_complex\_sampled()

```
void beidou_b1i_code_gen_complex_sampled (
    own::span< std::complex< float > > dest,
    uint32_t prn,
    int32_t sampling_freq,
    uint32_t chip_shift)
```

Generates complex GPS L1 C/A code for the desired SV ID and code shift, and sampled to specific sampling frequency.

#### 9.6.2.3.5 beidou\_b1i\_code\_gen\_float()

```
void beidou_b1i_code_gen_float (
    own::span< float > dest,
    int32_t prn,
    uint32_t chip_shift)
```

Generates float GPS L1 C/A code for the desired SV ID and code shift.

#### 9.6.2.3.6 beidou\_b1i\_code\_gen\_int()

```
void beidou_b1i_code_gen_int (
    own::span< int32_t > dest,
    int32_t prn,
    uint32_t chip_shift)
```

Generates int32\_t GPS L1 C/A code for the desired SV ID and code shift.

#### 9.6.2.3.7 beidou\_b3i\_code\_gen\_complex()

```
void beidou_b3i_code_gen_complex (
    own::span< std::complex< float > > dest,
    int32_t prn,
    uint32_t chip_shift)
```

Generates complex BeiDou B3I code for the desired SV ID and code shift.

#### 9.6.2.3.8 beidou\_b3i\_code\_gen\_complex\_sampled()

```
void beidou_b3i_code_gen_complex_sampled (
    own::span< std::complex< float > > dest,
    uint32_t prn,
    int sampling_freq,
    uint32_t chip_shift)
```

Generates complex BeiDou B3I code for the desired SV ID and code shift, and sampled to specific sampling frequency.

#### 9.6.2.3.9 beidou\_b3i\_code\_gen\_float()

```
void beidou_b3i_code_gen_float (
    own::span< float > dest,
    int32_t prn,
    uint32_t chip_shift)
```

Generates float BeiDou B3I code for the desired SV ID and code shift.

#### 9.6.2.3.10 beidou\_b3i\_code\_gen\_int()

```
void beidou_b3i_code_gen_int (
    own::span< int > dest,
    int32_t prn,
    uint32_t chip_shift)
```

Generates int BeiDou B3I code for the desired SV ID and code shift.



**9.6.2.3.11 cart2geo()**

```
arma::vec cart2geo (
    const arma::vec & XYZ,
    int ellipsoid_selection)
```

Conversion of Cartesian coordinates (X,Y,Z) to geographical coordinates (latitude, longitude, h) on a selected reference ellipsoid.

Choices of Reference Ellipsoid for Geographical Coordinates

1. International Ellipsoid 1924
2. International Ellipsoid 1967
3. World Geodetic System 1972
4. Geodetic Reference System 1980
5. World Geodetic System 1984

**9.6.2.3.12 cart2utm()**

```
void cart2utm (
    const arma::vec & r_eb_e,
    int zone,
    arma::vec & r_enu)
```

Transformation of ECEF (X,Y,Z) to (E,N,U) in UTM, zone 'zone'.

**9.6.2.3.13 clear()**

```
template<class T>
void Gnss_circular_deque< T >::clear (
    unsigned int ch)
```

Removes all the elements of the deque (Sets size to 0). Capacity is not modified.

Definition at line 102 of file [gnss\\_circular\\_deque.h](#).

**9.6.2.3.14 clkasin()**

```
void clkasin (
    const arma::colvec & ar,
    int degree,
    double arg_real,
    double arg_imag,
    double * re,
    double * im)
```

Clenshaw summation of sinus with complex argument.

#### 9.6.2.3.15 `clsin()`

```
double clsin (
    const arma::colvec & ar,
    int degree,
    double argument)
```

Clenshaw summation of sinus of argument.

#### 9.6.2.3.16 `complex_exp_gen()`

```
void complex_exp_gen (
    own::span< std::complex< float > > dest,
    double freq,
    double sampling_freq)
```

This function generates a complex exponential in dest.

#### 9.6.2.3.17 `complex_exp_gen_conj()`

```
void complex_exp_gen_conj (
    own::span< std::complex< float > > dest,
    double freq,
    double sampling_freq)
```

This function generates a conjugate complex exponential in dest.

#### 9.6.2.3.18 `findUtmZone()`

```
int findUtmZone (
    double latitude_deg,
    double longitude_deg)
```

Function finds the UTM zone number for given longitude and latitude.

#### 9.6.2.3.19 `front()`

```
template<class T>
T & Gnss_circular_deque< T >::front (
    unsigned int ch)
```

Returns a reference to the first element in the deque.

Definition at line 81 of file [gnss\\_circular\\_deque.h](#).

**9.6.2.3.20 galileo\_e1\_code\_gen\_complex\_sampled() [1/2]**

```
void galileo_e1_code_gen_complex_sampled (
    own::span< std::complex< float > > dest,
    const std::array< char, 3 > & signal_id,
    bool cboc,
    uint32_t prn,
    int32_t sampling_freq,
    uint32_t chip_shift)
```

galileo\_e1\_code\_gen\_complex\_sampled without secondary\_flag for backward compatibility.

**9.6.2.3.21 galileo\_e1\_code\_gen\_complex\_sampled() [2/2]**

```
void galileo_e1_code_gen_complex_sampled (
    own::span< std::complex< float > > dest,
    const std::array< char, 3 > & signal_id,
    bool cboc,
    uint32_t prn,
    int32_t sampling_freq,
    uint32_t chip_shift,
    bool secondary_flag)
```

This function generates Galileo E1 code (can select E1B or E1C, cboc or sinboc and the sample frequency sampling\_freq).

**9.6.2.3.22 galileo\_e1\_code\_gen\_float\_sampled() [1/2]**

```
void galileo_e1_code_gen_float_sampled (
    own::span< float > dest,
    const std::array< char, 3 > & signal_id,
    bool cboc,
    uint32_t prn,
    int32_t sampling_freq,
    uint32_t chip_shift)
```

This function generates Galileo E1 code (can select E1B or E1C, cboc or sinboc and the sample frequency sampling\_freq).

**9.6.2.3.23 galileo\_e1\_code\_gen\_float\_sampled() [2/2]**

```
void galileo_e1_code_gen_float_sampled (
    own::span< float > dest,
    const std::array< char, 3 > & signal_id,
    bool cboc,
    uint32_t prn,
    int32_t sampling_freq,
    uint32_t chip_shift,
    bool secondary_flag)
```

This function generates Galileo E1 code (can select E1B or E1C, cboc or sinboc and the sample frequency sampling\_freq).

**9.6.2.3.24 galileo\_e1\_code\_gen\_sinboc11\_float()**

```
void galileo_e1_code_gen_sinboc11_float (
    own::span< float > dest,
    const std::array< char, 3 > & signal_id,
    uint32_t prn)
```

This function generates Galileo E1 code (can select E1B or E1C sinboc).

**9.6.2.3.25 galileo\_e5\_a\_code\_gen\_complex\_primary()**

```
void galileo_e5_a_code_gen_complex_primary (
    own::span< std::complex< float > > dest,
    int32_t prn,
    const std::array< char, 3 > & signal_id)
```

Generates Galileo E5a code at 1 sample/chip.

**9.6.2.3.26 galileo\_e5\_a\_code\_gen\_complex\_sampled()**

```
void galileo_e5_a_code_gen_complex_sampled (
    own::span< std::complex< float > > dest,
    uint32_t prn,
    const std::array< char, 3 > & signal_id,
    int32_t sampling_freq,
    uint32_t chip_shift)
```

Generates Galileo E5a complex code, shifted to the desired chip and sampled at a frequency sampling\_freq.

**9.6.2.3.27 galileo\_e5\_b\_code\_gen\_complex\_primary()**

```
void galileo_e5_b_code_gen_complex_primary (
    own::span< std::complex< float > > dest,
    int32_t prn,
    const std::array< char, 3 > & signal_id)
```

Generates Galileo E5b code at 1 sample/chip.

**9.6.2.3.28 galileo\_e5\_b\_code\_gen\_complex\_sampled()**

```
void galileo_e5_b_code_gen_complex_sampled (
    own::span< std::complex< float > > dest,
    uint32_t prn,
    const std::array< char, 3 > & signal_id,
    int32_t sampling_freq,
    uint32_t chip_shift)
```

Generates Galileo E5b complex code, shifted to the desired chip and sampled at a frequency sampling\_freq.

**9.6.2.3.29 galileo\_e6\_b\_code\_gen\_complex\_primary()**

```
void galileo_e6_b_code_gen_complex_primary (
    own::span< std::complex< float > > dest,
    int32_t prn)
```

Generates Galileo E6B code at 1 sample/chip.

**9.6.2.3.30 galileo\_e6\_b\_code\_gen\_complex\_sampled()**

```
void galileo_e6_b_code_gen_complex_sampled (
    own::span< std::complex< float > > dest,
    uint32_t prn,
    int32_t sampling_freq,
    uint32_t chip_shift)
```

Generates Galileo E6B complex code, shifted to the desired chip and sampled at a frequency `sampling_freq`.

**9.6.2.3.31 galileo\_e6\_b\_code\_gen\_float\_primary()**

```
void galileo_e6_b_code_gen_float_primary (
    own::span< float > dest,
    int32_t prn)
```

Generates Galileo E6B code at 1 sample/chip.

**9.6.2.3.32 galileo\_e6\_c\_code\_gen\_complex\_primary()**

```
void galileo_e6_c_code_gen_complex_primary (
    own::span< std::complex< float > > dest,
    int32_t prn)
```

Generates Galileo E6C codes at 1 sample/chip.

**9.6.2.3.33 galileo\_e6\_c\_code\_gen\_complex\_sampled()**

```
void galileo_e6_c_code_gen_complex_sampled (
    own::span< std::complex< float > > dest,
    uint32_t prn,
    int32_t sampling_freq,
    uint32_t chip_shift)
```

Generates Galileo E6C complex codes, shifted to the desired chip and sampled at a frequency `sampling_freq`.

**9.6.2.3.34 galileo\_e6\_c\_code\_gen\_float\_primary()**

```
void galileo_e6_c_code_gen_float_primary (
    own::span< float > dest,
    int32_t prn)
```

Generates Galileo E6C codes at 1 sample/chip.

**9.6.2.3.35 galileo\_e6\_c\_secondary\_code()**

```
std::string galileo_e6_c_secondary_code (
    int32_t prn)
```

Generates a string with Galileo E6C secondary codes at 1 sample/chip.

**9.6.2.3.36 galileo\_e6\_c\_secondary\_code\_gen\_complex()**

```
void galileo_e6_c_secondary_code_gen_complex (
    own::span< std::complex< float > > dest,
    int32_t prn)
```

Generates Galileo E6C secondary codes at 1 sample/chip.

**9.6.2.3.37 galileo\_e6\_c\_secondary\_code\_gen\_float()**

```
void galileo_e6_c_secondary_code_gen_float (
    own::span< float > dest,
    int32_t prn)
```

Generates Galileo E6C secondary codes at 1 sample/chip.

**9.6.2.3.38 Geo\_to\_ECEF()**

```
void Geo_to_ECEF (
    const arma::vec & LLH,
    const arma::vec & v_eb_n,
    const arma::mat & C_b_n,
    arma::vec & r_eb_e,
    arma::vec & v_eb_e,
    arma::mat & C_b_e)
```

From Geographic to ECEF coordinates.

Inputs: LLH latitude (rad), longitude (rad), height (m)  $v_{eb\_n}$  velocity of body frame w.r.t. ECEF frame, resolved along north, east, and down (m/s)  $C_{b\_n}$  body-to-NED coordinate transformation matrix

Outputs:  $r_{eb\_e}$  Cartesian position of body frame w.r.t. ECEF frame, resolved along ECEF-frame axes (m)  $v_{eb\_e}$  velocity of body frame w.r.t. ECEF frame, resolved along ECEF-frame axes (m/s)  $C_{b\_e}$  body-to-ECEF-frame coordinate transformation matrix

**9.6.2.3.39 get()**

```
template<class T>
const T & Gnss_circular_deque< T >::get (
    unsigned int ch,
    unsigned int pos) const
```

Returns a const reference to an element without bound checking.

Definition at line 95 of file [gnss\\_circular\\_deque.h](#).

**9.6.2.3.40 glonass\_l1\_ca\_code\_gen\_complex()**

```
void glonass_l1_ca_code_gen_complex (
    own::span< std::complex< float > > dest,
    uint32_t chip_shift)
```

Generates complex GLONASS L1 C/A code for the desired SV ID and code shift.

**9.6.2.3.41 glonass\_l1\_ca\_code\_gen\_complex\_sampled()**

```
void glonass_l1_ca_code_gen_complex_sampled (
    own::span< std::complex< float > > dest,
    int32_t sampling_freq,
    uint32_t chip_shift)
```

Generates complex GLONASS L1 C/A code for the desired SV ID and code shift, and sampled to specific sampling frequency.

**9.6.2.3.42 glonass\_l2\_ca\_code\_gen\_complex()**

```
void glonass_l2_ca_code_gen_complex (
    own::span< std::complex< float > > dest,
    uint32_t chip_shift)
```

Generates complex GLONASS L2 C/A code for the desired SV ID and code shift.

**9.6.2.3.43 glonass\_l2\_ca\_code\_gen\_complex\_sampled()**

```
void glonass_l2_ca_code_gen_complex_sampled (
    own::span< std::complex< float > > dest,
    int32_t sampling_freq,
    uint32_t chip_shift)
```

Generates complex GLONASS L2 C/A code for the desired SV ID and code shift, and sampled to specific sampling frequency.

**9.6.2.3.44 Gnss\_circular\_deque() [1/2]**

```
template<class T>
Gnss_circular_deque< T >::Gnss_circular_deque ()
```

Default constructor.

Definition at line 53 of file [gnss\\_circular\\_deque.h](#).

References [reset\(\)](#).

Here is the call graph for this function:



#### 9.6.2.3.45 Gnss\_circular\_deque() [2/2]

```
template<class T>
Gnss_circular_deque< T >::Gnss_circular_deque (
    unsigned int max_size,
    unsigned int nchann)
```

nchann = number of channels; max\_size = channel capacity

Definition at line 60 of file [gnss\\_circular\\_deque.h](#).

References [reset\(\)](#).

Here is the call graph for this function:



#### 9.6.2.3.46 gps\_l1\_ca\_code\_gen\_complex()

```
void gps_l1_ca_code_gen_complex (
    own::span< std::complex< float > > dest,
    int32_t prn,
    uint32_t chip_shift)
```

Generates complex GPS L1 C/A code for the desired SV ID and code shift.

#### 9.6.2.3.47 gps\_l1\_ca\_code\_gen\_complex\_sampled()

```
void gps_l1_ca_code_gen_complex_sampled (
    own::span< std::complex< float > > dest,
    uint32_t prn,
    int32_t sampling_freq,
    uint32_t chip_shift)
```

Generates complex GPS L1 C/A code for the desired SV ID and code shift, and sampled to specific sampling frequency.

#### 9.6.2.3.48 gps\_l1\_ca\_code\_gen\_float()

```
void gps_l1_ca_code_gen_float (
    own::span< float > dest,
    int32_t prn,
    uint32_t chip_shift)
```

Generates float GPS L1 C/A code for the desired SV ID and code shift.



**9.6.2.3.49 `gps_l1_ca_code_gen_int()`**

```
void gps_l1_ca_code_gen_int (
    own::span< int32_t > dest,
    int32_t prn,
    uint32_t chip_shift)
```

Generates int GPS L1 C/A code for the desired SV ID and code shift.

**9.6.2.3.50 `gps_l2c_m_code_gen_complex()`**

```
void gps_l2c_m_code_gen_complex (
    own::span< std::complex< float > > dest,
    uint32_t prn)
```

Generates complex GPS L2C M code for the desired SV ID.

**9.6.2.3.51 `gps_l2c_m_code_gen_complex_sampled()`**

```
void gps_l2c_m_code_gen_complex_sampled (
    own::span< std::complex< float > > dest,
    uint32_t prn,
    int32_t sampling_freq)
```

Generates complex GPS L2C M code for the desired SV ID, and sampled to specific sampling frequency.

**9.6.2.3.52 `gps_l2c_m_code_gen_float()`**

```
void gps_l2c_m_code_gen_float (
    own::span< float > dest,
    uint32_t prn)
```

Generates float GPS L2C M code for the desired SV ID.

**9.6.2.3.53 `gps_l5i_code_gen_complex()`**

```
void gps_l5i_code_gen_complex (
    own::span< std::complex< float > > dest,
    uint32_t prn)
```

Generates complex GPS L5I code for the desired SV ID.

**9.6.2.3.54 `gps_l5i_code_gen_complex_sampled()`**

```
void gps_l5i_code_gen_complex_sampled (
    own::span< std::complex< float > > dest,
    uint32_t prn,
    int32_t sampling_freq)
```

Generates complex GPS L5I code for the desired SV ID, and sampled to specific sampling frequency.

**9.6.2.3.55 gps\_l5i\_code\_gen\_float()**

```
void gps_l5i_code_gen_float (
    own::span< float > dest,
    uint32_t prn)
```

Generates real GPS L5I code for the desired SV ID.

**9.6.2.3.56 gps\_l5q\_code\_gen\_complex()**

```
void gps_l5q_code_gen_complex (
    own::span< std::complex< float > > dest,
    uint32_t prn)
```

Generates complex GPS L5Q code for the desired SV ID.

**9.6.2.3.57 gps\_l5q\_code\_gen\_complex\_sampled()**

```
void gps_l5q_code_gen_complex_sampled (
    own::span< std::complex< float > > dest,
    uint32_t prn,
    int32_t sampling_freq)
```

Generates complex GPS L5Q code for the desired SV ID, and sampled to specific sampling frequency.

**9.6.2.3.58 gps\_l5q\_code\_gen\_float()**

```
void gps_l5q_code_gen_float (
    own::span< float > dest,
    uint32_t prn)
```

Generates real GPS L5Q code for the desired SV ID.

**9.6.2.3.59 Gravity\_ECEF()**

```
arma::vec Gravity_ECEF (
    const arma::vec & r_eb_e)
```

Calculates acceleration due to gravity resolved about ECEF-frame.

**9.6.2.3.60 great\_circle\_distance()**

```
double great_circle_distance (
    double lat1,
    double lon1,
    double lat2,
    double lon2)
```

The Haversine formula determines the great-circle distance between two points on a sphere given their longitudes and latitudes.

**9.6.2.3.61 hex\_to\_binary\_converter()**

```
void hex_to_binary_converter (
    own::span< int32_t > dest,
    char from)
```

This function makes a conversion from hex (the input is a char) to binary (the output are 4 ints with +1 or -1 values).

**9.6.2.3.62 hex\_to\_binary\_string()**

```
std::string hex_to_binary_string (
    char from)
```

This function makes a conversion from hex (the input is a char) to binary (the output is a string of 4 char with 0 or 1 values).

**9.6.2.3.63 item\_type\_is\_complex()**

```
bool item_type_is_complex (
    const std::string & item_type)
```

Determine if an item\_type is complex.

**9.6.2.3.64 item\_type\_size()**

```
size_t item_type_size (
    const std::string & item_type)
```

Return the size of the given item type, or zero if unknown.

**9.6.2.3.65 item\_type\_valid()**

```
bool item_type_valid (
    const std::string & item_type)
```

Check if a string is a valid item type.

\description Valid item types include: "byte", "short", "float", "ibyte", "ishort", "cbyte", "cshort", "gr\_complex"

**9.6.2.3.66 make\_vector\_converter()**

```
item_type_converter_t make_vector_converter (
    const std::string & input_type,
    const std::string & output_type)
```

Create a function to convert an array of input\_type to an array of output\_type.

\description Provides a generic interface to generate conversion functions for mapping arrays of items.

**Parameters**

<i>input_type</i>	- String representation of the input item type
<i>output_type</i>	- String representation of the output item type

The item types accepted are:

1. "byte" for 8 bit integers
2. "cbyte" for complex (interleaved) 8 bit integers
3. "ibyte" for complex (interleaved) 8 bit integers
  
1. "short" for 16 bit integers
2. "cshort" for complex (interleaved) 16 bit integers
3. "ishort" for complex (interleaved) 16 bit integers
4. "float" for 32 bit floating point values
5. "gr\_complex" for complex (interleaved) 32 bit floating point values

**Returns**

A function object with the following prototype: `void convert_fun( void *dest, void *src, int num_items );`

**9.6.2.3.67 pop\_front()**

```
template<class T>
void Gnss_circular_deque< T >::pop_front (
    unsigned int ch)
```

Removes the first element of the deque.

Definition at line 130 of file [gnss\\_circular\\_deque.h](#).

**9.6.2.3.68 push\_back()**

```
template<class T>
void Gnss_circular_deque< T >::push_back (
    unsigned int ch,
    const T & new_data)
```

Inserts an element at the end of the deque.

Definition at line 137 of file [gnss\\_circular\\_deque.h](#).

**9.6.2.3.69 pv\_Geo\_to\_ECEF()**

```
void pv_Geo_to_ECEF (
    double L_b,
    double lambda_b,
    double h_b,
    const arma::vec & v_eb_n,
    arma::vec & r_eb_e,
    arma::vec & v_eb_e)
```

Converts curvilinear to Cartesian position and velocity resolving axes from NED to ECEF This function created 11/4/2012 by Paul Groves.

Inputs: L\_b latitude (rad) lambda\_b longitude (rad) h\_b height (m) v\_eb\_n velocity of body frame w.r.t. ECEF frame, resolved along north, east, and down (m/s)

Outputs: r\_eb\_e Cartesian position of body frame w.r.t. ECEF frame, resolved along ECEF-frame axes (m) v\_eb\_e velocity of body frame w.r.t. ECEF frame, resolved along ECEF-frame axes (m/s)

**9.6.2.3.70 resampler() [1/2]**

```
void resampler (
    const own::span< float > from,
    own::span< float > dest,
    float fs_in,
    float fs_out)
```

This function resamples a sequence of float values.

**9.6.2.3.71 resampler() [2/2]**

```
void resampler (
    own::span< const std::complex< float > > from,
    own::span< std::complex< float > > dest,
    float fs_in,
    float fs_out)
```

This function resamples a sequence of complex values.

**9.6.2.3.72 reset() [1/2]**

```
template<class T>
void Gnss_circular_deque< T >::reset ()
```

Removes all the channels (Sets nchann to 0)

Definition at line 123 of file [gnss\\_circular\\_deque.h](#).

**9.6.2.3.73 reset() [2/2]**

```
template<class T>
void Gnss_circular_deque< T >::reset (
    unsigned int max_size,
    unsigned int nchann)
```

Removes all the elements in all the channels. Re-sets the number of channels and their capacity.

Definition at line 109 of file [gnss\\_circular\\_deque.h](#).

Referenced by [Gnss\\_circular\\_deque\(\)](#), and [Gnss\\_circular\\_deque\(\)](#).

**9.6.2.3.74 size()**

```
template<class T>
unsigned int Gnss_circular_deque< T >::size (
    unsigned int ch) const
```

Returns the number of available elements in a channel.

Definition at line 67 of file [gnss\\_circular\\_deque.h](#).

**9.6.2.3.75 Skew\_symmetric()**

```
arma::mat Skew_symmetric (
    const arma::vec & a)
```

Calculates skew-symmetric matrix.

**9.6.2.3.76 togeod()**

```
int togeod (
    double * dphi,
    double * dlambd,
    double * h,
    double a,
    double finv,
    double X,
    double Y,
    double Z)
```

Subroutine to calculate geodetic coordinates latitude, longitude, height given Cartesian coordinates X,Y,Z, and reference ellipsoid values semi-major axis (a) and the inverse of flattening (finv).

The output units of angular quantities will be in decimal degrees (15.5 degrees not 15 deg 30 min). The output units of h will be the same as the units of X,Y,Z,a.

Inputs:

a	- semi-major axis of the reference ellipsoid
finv	- inverse of flattening of the reference ellipsoid
X,Y,Z	- Cartesian coordinates

Outputs:

dphi	- latitude
dlambda	- longitude
h	- height above reference ellipsoid

Based in a Matlab function by Kai Borre

### 9.6.2.3.77 topocent()

```
int topocent (
    double * Az,
    double * El,
    double * D,
    const arma::vec & x,
    const arma::vec & dx)
```

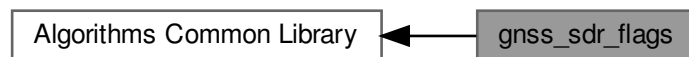
Transformation of vector dx into topocentric coordinate system with origin at x Inputs: x - vector origin coordinates (in ECEF system [X; Y; Z;]) dx - vector ([dX; dY; dZ;]).

Outputs: D - vector length. Units like the input Az - azimuth from north positive clockwise, degrees El - elevation angle, degrees

Based on a Matlab function by Kai Borre

## 9.6.3 gnss\_sdr\_flags

Collaboration diagram for gnss\_sdr\_flags:



### Functions

- [ABSL\\_DECLARE\\_FLAG](#) (std::string, c)  
*Path to the configuration file.*
- [ABSL\\_DECLARE\\_FLAG](#) (std::string, config\_file)  
*Path to the configuration file.*
- [ABSL\\_DECLARE\\_FLAG](#) (std::string, log\_dir)  
*Path to the folder in which logging will be stored.*
- [ABSL\\_DECLARE\\_FLAG](#) (std::string, s)  
*Path to the file containing the signal samples.*
- [ABSL\\_DECLARE\\_FLAG](#) (std::string, signal\_source)  
*Path to the file containing the signal samples.*
- [ABSL\\_DECLARE\\_FLAG](#) (std::string, timestamp\_source)  
*Path to the file containing the signal samples.*
- [ABSL\\_DECLARE\\_FLAG](#) (bool, rf\_shutdown)  
*Shutdown RF when program exits.*
- [ABSL\\_DECLARE\\_FLAG](#) (int32\_t, doppler\_max)  
*If defined, maximum Doppler value in the search grid, in Hz (overrides the configuration file).*
- [ABSL\\_DECLARE\\_FLAG](#) (int32\_t, doppler\_step)  
*If defined, sets the frequency step in the search grid, in Hz, in Hz (overrides the configuration file).*
- [ABSL\\_DECLARE\\_FLAG](#) (int32\_t, cn0\_samples)

- *Number of correlator outputs used for CN0 estimation.*  
 • `ABSL_DECLARE_FLAG` (int32\_t, cn0\_min)  
*Minimum valid CN0 (in dB-Hz).*
- `ABSL_DECLARE_FLAG` (int32\_t, max\_lock\_fail)  
*Maximum number of code lock failures before dropping a satellite.*
- `ABSL_DECLARE_FLAG` (int32\_t, max\_carrier\_lock\_fail)  
*Maximum number of carrier lock failures before dropping a satellite.*
- `ABSL_DECLARE_FLAG` (double, carrier\_lock\_th)  
*Carrier lock threshold (in rad).*
- `ABSL_DECLARE_FLAG` (double, dll\_bw\_hz)  
*Bandwidth of the DLL low pass filter, in Hz (overrides the configuration file).*
- `ABSL_DECLARE_FLAG` (double, pll\_bw\_hz)  
*Bandwidth of the PLL low pass filter, in Hz (overrides the configuration file).*
- `ABSL_DECLARE_FLAG` (int32\_t, carrier\_smoothing\_factor)  
*Sets carrier smoothing factor M (overrides the configuration file).*
- `ABSL_DECLARE_FLAG` (std::string, RINEX\_version)  
*If defined, specifies the RINEX version (2.11 or 3.02). Overrides the configuration file.*
- `ABSL_DECLARE_FLAG` (std::string, RINEX\_name)  
*If defined, specifies the RINEX files base name.*
- `ABSL_DECLARE_FLAG` (bool, keyboard)  
*If set to false, disables the keyboard listener. Only for debug purposes (e.g. ASAN mode termination)*
- bool `ValidateFlags` ()

## Variables

- const int32\_t `DEFAULT_CARRIER_SMOOTHING_FACTOR` = 200

### 9.6.3.1 Detailed Description

Library for command-line handling.

### 9.6.3.2 Function Documentation

#### 9.6.3.2.1 `ABSL_DECLARE_FLAG()` [1/20]

```
ABSL_DECLARE_FLAG (
    bool ,
    keyboard )
```

If set to false, disables the keyboard listener. Only for debug purposes (e.g. ASAN mode termination)

#### 9.6.3.2.2 `ABSL_DECLARE_FLAG()` [2/20]

```
ABSL_DECLARE_FLAG (
    bool ,
    rf_shutdown )
```

Shutdown RF when program exits.



**9.6.3.2.3 ABSL\_DECLARE\_FLAG() [3/20]**

```
ABSL_DECLARE_FLAG (
    double ,
    carrier_lock_th )
```

Carrier lock threshold (in rad).

**9.6.3.2.4 ABSL\_DECLARE\_FLAG() [4/20]**

```
ABSL_DECLARE_FLAG (
    double ,
    dll_bw_hz )
```

Bandwidth of the DLL low pass filter, in Hz (overrides the configuration file).

**9.6.3.2.5 ABSL\_DECLARE\_FLAG() [5/20]**

```
ABSL_DECLARE_FLAG (
    double ,
    pll_bw_hz )
```

Bandwidth of the PLL low pass filter, in Hz (overrides the configuration file).

**9.6.3.2.6 ABSL\_DECLARE\_FLAG() [6/20]**

```
ABSL_DECLARE_FLAG (
    int32_t ,
    carrier_smoothing_factor )
```

Sets carrier smoothing factor M (overrides the configuration file).

**9.6.3.2.7 ABSL\_DECLARE\_FLAG() [7/20]**

```
ABSL_DECLARE_FLAG (
    int32_t ,
    cn0_min )
```

Minimum valid CN0 (in dB-Hz).

**9.6.3.2.8 ABSL\_DECLARE\_FLAG() [8/20]**

```
ABSL_DECLARE_FLAG (
    int32_t ,
    cn0_samples )
```

Number of correlator outputs used for CN0 estimation.

**9.6.3.2.9 ABSL\_DECLARE\_FLAG()** [9/20]

```
ABSL_DECLARE_FLAG (
    int32_t ,
    doppler_max )
```

If defined, maximum Doppler value in the search grid, in Hz (overrides the configuration file).

**9.6.3.2.10 ABSL\_DECLARE\_FLAG()** [10/20]

```
ABSL_DECLARE_FLAG (
    int32_t ,
    doppler_step )
```

If defined, sets the frequency step in the search grid, in Hz, in Hz (overrides the configuration file).

**9.6.3.2.11 ABSL\_DECLARE\_FLAG()** [11/20]

```
ABSL_DECLARE_FLAG (
    int32_t ,
    max_carrier_lock_fail )
```

Maximum number of carrier lock failures before dropping a satellite.

**9.6.3.2.12 ABSL\_DECLARE\_FLAG()** [12/20]

```
ABSL_DECLARE_FLAG (
    int32_t ,
    max_lock_fail )
```

Maximum number of code lock failures before dropping a satellite.

**9.6.3.2.13 ABSL\_DECLARE\_FLAG()** [13/20]

```
ABSL_DECLARE_FLAG (
    std::string ,
    c )
```

Path to the configuration file.

**9.6.3.2.14 ABSL\_DECLARE\_FLAG()** [14/20]

```
ABSL_DECLARE_FLAG (
    std::string ,
    config_file )
```

Path to the configuration file.

**9.6.3.2.15 ABSL\_DECLARE\_FLAG()** [15/20]

```
ABSL_DECLARE_FLAG (  
    std::string ,  
    log_dir )
```

Path to the folder in which logging will be stored.

**9.6.3.2.16 ABSL\_DECLARE\_FLAG()** [16/20]

```
ABSL_DECLARE_FLAG (  
    std::string ,  
    RINEX_name )
```

If defined, specifies the RINEX files base name.

**9.6.3.2.17 ABSL\_DECLARE\_FLAG()** [17/20]

```
ABSL_DECLARE_FLAG (  
    std::string ,  
    RINEX_version )
```

If defined, specifies the RINEX version (2.11 or 3.02). Overrides the configuration file.

**9.6.3.2.18 ABSL\_DECLARE\_FLAG()** [18/20]

```
ABSL_DECLARE_FLAG (  
    std::string ,  
    s )
```

Path to the file containing the signal samples.

**9.6.3.2.19 ABSL\_DECLARE\_FLAG()** [19/20]

```
ABSL_DECLARE_FLAG (  
    std::string ,  
    signal_source )
```

Path to the file containing the signal samples.

**9.6.3.2.20 ABSL\_DECLARE\_FLAG()** [20/20]

```
ABSL_DECLARE_FLAG (  
    std::string ,  
    timestamp_source )
```

Path to the file containing the signal samples.

### 9.6.3.3 Variable Documentation

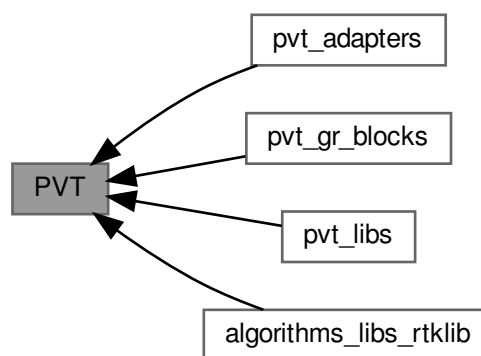
#### 9.6.3.3.1 DEFAULT\_CARRIER\_SMOOTHING\_FACTOR

```
const int32_t DEFAULT_CARRIER_SMOOTHING_FACTOR = 200
```

Definition at line 103 of file [gnss\\_sdr\\_flags.h](#).

## 9.7 PVT

Collaboration diagram for PVT:



### Topics

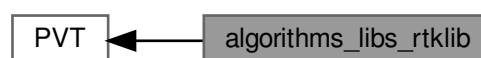
- [algorithms\\_libs\\_rtklib](#)
- [pvt\\_adapters](#)
- [pvt\\_gr\\_blocks](#)
- [pvt\\_libs](#)

### 9.7.1 Detailed Description

Computation of Position, Velocity and Time from GNSS observables.

### 9.7.2 algorithms\_libs\_rtklib

Collaboration diagram for algorithms\_libs\_rtklib:



## Classes

- struct [gtime\\_t](#)
- struct [obsd\\_t](#)
- struct [obs\\_t](#)
- struct [erpd\\_t](#)
- struct [erp\\_t](#)
- struct [pcv\\_t](#)
- struct [pcvs\\_t](#)
- struct [alm\\_t](#)
- struct [eph\\_t](#)
- struct [geph\\_t](#)
- struct [peph\\_t](#)
- struct [pclk\\_t](#)
- struct [seph\\_t](#)
- struct [tled\\_t](#)
- struct [tle\\_t](#)
- struct [tec\\_t](#)
- struct [fcbd\\_t](#)
- struct [sbsmsg\\_t](#)
- struct [sbs\\_t](#)
- struct [sbsfcrr\\_t](#)
- struct [sbslcorr\\_t](#)
- struct [sbssatp\\_t](#)
- struct [sbssat\\_t](#)
- struct [sbsigp\\_t](#)
- struct [sbsigpband\\_t](#)
- struct [sbsion\\_t](#)
- struct [dgps\\_t](#)
- struct [ssr\\_t](#)
- struct [lexmsg\\_t](#)
- struct [lex\\_t](#)
- struct [lexeph\\_t](#)
- struct [lexion\\_t](#)
- struct [stec\\_t](#)
- struct [trop\\_t](#)
- struct [pppcorr\\_t](#)
- struct [nav\\_t](#)
- struct [sta\\_t](#)
- struct [sol\\_t](#)
- struct [solbuf\\_t](#)
- struct [solstat\\_t](#)
- struct [solstatbuf\\_t](#)
- struct [rtcm\\_t](#)
- struct [url\\_t](#)
- struct [opt\\_t](#)
- struct [exterr\\_t](#)
- struct [snrmask\\_t](#)
- struct [prcopt\\_t](#)
- struct [solopt\\_t](#)
- struct [ssat\\_t](#)
- struct [ambc\\_t](#)
- struct [rtk\\_t](#)
- struct [half\\_cyc\\_tag](#)
- struct [stream\\_t](#)

- struct [serial\\_t](#)
- struct [file\\_t](#)
- struct [tcp\\_t](#)
- struct [tcpsvr\\_t](#)
- struct [tcpcli\\_t](#)
- struct [ntrip\\_t](#)
- struct [ftp\\_t](#)
- struct [raw\\_t](#)
- struct [rtksvr\\_t](#)
- struct [msm\\_h\\_t](#)
- class [HAS\\_clock\\_corrections](#)
- class [HAS\\_orbit\\_corrections](#)
- class [HAS\\_obs\\_corrections](#)

## Macros

- #define [dev\\_t](#) int
- #define [socket\\_t](#) int
- #define [closesocket](#) close
- #define [lock\\_t](#) pthread\_mutex\_t
- #define [initlock](#)(f)
- #define [rtk\\_lock](#)(f)
- #define [rtk\\_unlock](#)(f)
- #define [VER\\_RTKLIB](#) "2.4.2"
- #define [NTRIP\\_AGENT](#) "RTKLIB/" VER\_RTKLIB
- #define [NTRIP\\_CLI\\_PORT](#) 2101 /\* default ntrip-client connection port \*/
- #define [NTRIP\\_SVR\\_PORT](#) 80 /\* default ntrip-server connection port \*/
- #define [NTRIP\\_MAXRSP](#) 32768 /\* max size of ntrip response \*/
- #define [NTRIP\\_MAXSTR](#) 256 /\* max length of mountpoint string \*/
- #define [NTRIP\\_RSP\\_OK\\_CLI](#) "ICY 200 OK\r\n" /\* ntrip response: client \*/
- #define [NTRIP\\_RSP\\_OK\\_SVR](#) "OK\r\n" /\* ntrip response: server \*/
- #define [NTRIP\\_RSP\\_SRCTBL](#) "SOURCETABLE 200 OK\r\n" /\* ntrip response: source table \*/
- #define [NTRIP\\_RSP\\_TBLEND](#) "ENDSOURCETABLE"
- #define [NTRIP\\_RSP\\_HTTP](#) "HTTP/" /\* ntrip response: http \*/
- #define [NTRIP\\_RSP\\_ERROR](#) "ERROR" /\* ntrip response: error \*/
- #define [FTP\\_CMD](#) "wget" /\* ftp/http command \*/
- #define [ENAGLO](#)
- #define [ENABDS](#)
- #define [STR\\_MODE\\_R](#) 0x1 /\* stream mode: read \*/
- #define [STR\\_MODE\\_W](#) 0x2 /\* stream mode: write \*/
- #define [STR\\_MODE\\_RW](#) 0x3 /\* stream mode: read/write \*/
- #define [STR\\_NONE](#) 0 /\* stream type: none \*/
- #define [STR\\_SERIAL](#) 1 /\* stream type: serial \*/
- #define [STR\\_FILE](#) 2 /\* stream type: file \*/
- #define [STR\\_TCPSVR](#) 3 /\* stream type: TCP server \*/
- #define [STR\\_TCPCLI](#) 4 /\* stream type: TCP client \*/
- #define [STR\\_UDP](#) 5 /\* stream type: UDP stream \*/
- #define [STR\\_NTRIPSVR](#) 6 /\* stream type: NTRIP server \*/
- #define [STR\\_NTRIPCLI](#) 7 /\* stream type: NTRIP client \*/
- #define [STR\\_FTP](#) 8 /\* stream type: ftp \*/
- #define [STR\\_HTTP](#) 9 /\* stream type: http \*/
- #define [NP\\_PPP](#)(opt)
- #define [IC\\_PPP](#)(s, opt)
- #define [IT\\_PPP](#)(opt)

- `#define NR_PPP(opt)`
- `#define IB_PPP(s, opt)`
- `#define NX_PPP(opt)`
- `#define NF_RTK(opt)`
- `#define NP_RTK(opt)`
- `#define NI_RTK(opt)`
- `#define NT_RTK(opt)`
- `#define NL_RTK(opt)`
- `#define NB_RTK(opt)`
- `#define NR_RTK(opt)`
- `#define NX_RTK(opt)`
- `#define II_RTK(s, opt)`
- `#define IT_RTK(r, opt)`
- `#define IL_RTK(f, opt)`
- `#define IB_RTK(s, f, opt)`
- `#define COMMENTH "%" /* comment line indicator for solution */`
- `#define MSG_DISCONN "$_DISCONNECT\r\n" /* disconnect message */`

### Typedefs

- using `fatalfunc_t` = `void(const char *)`  
*fatal callback function type*
- typedef struct `half_cyc_tag` `half_cyc_t`

### Functions

- `eph_t eph_to_rtklib` (const `Galileo_Ephemeris` &gal\_eph)
- `eph_t eph_to_rtklib` (const `Galileo_Ephemeris` &gal\_eph, const std::map< int, `HAS_orbit_corrections` > &orbit\_correction\_map, const std::map< int, `HAS_clock_corrections` > &clock\_correction\_map)
- `eph_t eph_to_rtklib` (const `Gps_Ephemeris` &gps\_eph, bool pre\_2009\_file=false)
- `eph_t eph_to_rtklib` (const `Gps_Ephemeris` &gps\_eph, const std::map< int, `HAS_orbit_corrections` > &orbit\_correction\_map, const std::map< int, `HAS_clock_corrections` > &clock\_correction\_map, bool pre\_2009\_file=false)
- `eph_t eph_to_rtklib` (const `Gps_CNAV_Ephemeris` &gps\_cnav\_eph)
- `eph_t eph_to_rtklib` (const `Beidou_Dnav_Ephemeris` &bei\_eph)
- `alm_t alm_to_rtklib` (const `Gps_Almanac` &gps\_alm)
- `alm_t alm_to_rtklib` (const `Galileo_Almanac` &gal\_alm)
- `geph_t eph_to_rtklib` (const `Glonass_Gnav_Ephemeris` &glonass\_gnav\_eph, const `Glonass_Gnav_Utc_Model` &gnav\_clock\_model)  
*Transforms a `Glonass_Gnav_Ephemeris` to its RTKLIB counterpart.*
- `obsd_t insert_obs_to_rtklib` (`obsd_t` &rtklib\_obs, const `Gnss_Synchro` &gnss\_synchro, const std::map< std::string, std::map< int, `HAS_obs_corrections` > > &has\_obs\_corr, int week, int band, bool pre\_2009\_file=false)
- `obsd_t insert_obs_to_rtklib` (`obsd_t` &rtklib\_obs, const `Gnss_Synchro` &gnss\_synchro, int week, int band, bool pre\_2009\_file=false)
- int `rtkopenstat` (const char \*file, int level)
- void `rtkclosestat` ()
- void `rtkoutstat` (`rtk_t` \*rtk)
- void `swapsolstat` ()
- void `outsolstat` (`rtk_t` \*rtk)
- void `errmsg` (`rtk_t` \*rtk, const char \*format,...)
- double `sdocs` (const `obsd_t` \*obs, int i, int j, int f)
- double `globs_L1L2` (const `obsd_t` \*obs, int i, int j, const double \*lam)

- double **gfobs\_L1L5** (const [obsd\\_t](#) \*obs, int i, int j, const double \*lam)
- double **varerr** (int sat, int sys, double el, double bl, double dt, int f, const [prcopt\\_t](#) \*opt)
- double **baseline** (const double \*ru, const double \*rb, double \*dr)
- void **initx\_rtk** ([rtk\\_t](#) \*rtk, double xi, double var, int i)
- int **selsat** (const [obsd\\_t](#) \*obs, const double \*azel, int nu, int nr, const [prcopt\\_t](#) \*opt, int \*sat, int \*iu, int \*ir)
- void **udpos** ([rtk\\_t](#) \*rtk, double tt)
- void **udion** ([rtk\\_t](#) \*rtk, double tt, double bl, const int \*sat, int ns)
- void **udtrop** ([rtk\\_t](#) \*rtk, double tt, double bl)
- void **udrcvbias** ([rtk\\_t](#) \*rtk, double tt)
- void **detslp\_ll** ([rtk\\_t](#) \*rtk, const [obsd\\_t](#) \*obs, int i, int rcv)
- void **detslp\_gf\_L1L2** ([rtk\\_t](#) \*rtk, const [obsd\\_t](#) \*obs, int i, int j, const [nav\\_t](#) \*nav)
- void **detslp\_gf\_L1L5** ([rtk\\_t](#) \*rtk, const [obsd\\_t](#) \*obs, int i, int j, const [nav\\_t](#) \*nav)
- void **detslp\_dop** ([rtk\\_t](#) \*rtk, const [obsd\\_t](#) \*obs, int i, int rcv, const [nav\\_t](#) \*nav)
- void **udbias** ([rtk\\_t](#) \*rtk, double tt, const [obsd\\_t](#) \*obs, const int \*sat, const int \*iu, const int \*ir, int ns, const [nav\\_t](#) \*nav)
- void **udstate** ([rtk\\_t](#) \*rtk, const [obsd\\_t](#) \*obs, const int \*sat, const int \*iu, const int \*ir, int ns, const [nav\\_t](#) \*nav)
- void **zdres\_sat** (int base, double r, const [obsd\\_t](#) \*obs, const [nav\\_t](#) \*nav, const double \*azel, const double \*dant, const [prcopt\\_t](#) \*opt, double \*y)
- int **zdres** (int base, const [obsd\\_t](#) \*obs, int n, const double \*rs, const double \*dts, const int \*svh, const [nav\\_t](#) \*nav, const double \*rr, const [prcopt\\_t](#) \*opt, int index, double \*y, double \*e, double \*azel)
- int **validobs** (int i, int j, int f, int nf, const double \*y)
- void **ddcov** (const int \*nb, int n, const double \*Ri, const double \*Rj, int nv, double \*R)
- int **constbl** ([rtk\\_t](#) \*rtk, const double \*x, const double \*P, double \*v, double \*H, double \*Ri, double \*Rj, int index)
- double **prectrop** ([gtime\\_t](#) time, const double \*pos, int r, const double \*azel, const [prcopt\\_t](#) \*opt, const double \*x, double \*dtdx)
- double **gloicbcorr** (int sat1, int sat2, const [prcopt\\_t](#) \*opt, double lam1, double lam2, int f)
- int **test\_sys** (int sys, int m)
- int **ddres** ([rtk\\_t](#) \*rtk, const [nav\\_t](#) \*nav, double dt, const double \*x, const double \*P, const int \*sat, double \*y, const double \*e, double \*azel, const int \*iu, const int \*ir, int ns, double \*v, double \*H, double \*R, int \*vflg)
- double **intpres** ([gtime\\_t](#) time, const [obsd\\_t](#) \*obs, int n, const [nav\\_t](#) \*nav, [rtk\\_t](#) \*rtk, double \*y)
- int **ddmat** ([rtk\\_t](#) \*rtk, double \*D)
- void **restamb** ([rtk\\_t](#) \*rtk, const double \*bias, int nb, double \*xa)
- void **holdamb** ([rtk\\_t](#) \*rtk, const double \*xa)
- int **resamb\_LAMBDA** ([rtk\\_t](#) \*rtk, double \*bias, double \*xa)
- int **valpos** ([rtk\\_t](#) \*rtk, const double \*v, const double \*R, const int \*vflg, int nv, double thres)
- int **relpos** ([rtk\\_t](#) \*rtk, const [obsd\\_t](#) \*obs, int nu, int nr, const [nav\\_t](#) \*nav)
- void **rtkinit** ([rtk\\_t](#) \*rtk, const [prcopt\\_t](#) \*opt)
- void **rtkfree** ([rtk\\_t](#) \*rtk)
- int **rtkpos** ([rtk\\_t](#) \*rtk, const [obsd\\_t](#) \*obs, int n, const [nav\\_t](#) \*nav)
- const char \* **opt2sep** (const [solopt\\_t](#) \*opt)
- int **tonum** (char \*buff, const char \*sep, double \*v)
- double **sqvar** (double covar)
- double **dmm2deg** (double dmm)
- void **septime** (double t, double \*t1, double \*t2, double \*t3)
- void **soltocov** (const [sol\\_t](#) \*sol, double \*P)
- void **covtosol** (const double \*P, [sol\\_t](#) \*sol)
- int **decode\_nmearmc** (char \*\*val, int n, [sol\\_t](#) \*sol)
- int **decode\_nmeagga** (char \*\*val, int n, [sol\\_t](#) \*sol)
- int **decode\_nmea** (char \*buff, [sol\\_t](#) \*sol)
- char \* **decode\_soltime** (char \*buff, const [solopt\\_t](#) \*opt, [gtime\\_t](#) \*time)
- int **decode\_solxyz** (char \*buff, const [solopt\\_t](#) \*opt, [sol\\_t](#) \*sol)
- int **decode\_solllh** (char \*buff, const [solopt\\_t](#) \*opt, [sol\\_t](#) \*sol)
- int **decode\_solenu** (char \*buff, const [solopt\\_t](#) \*opt, [sol\\_t](#) \*sol)
- int **decode\_solgsi** (char \*buff, const [solopt\\_t](#) \*opt, [sol\\_t](#) \*sol)



- int **decode\_solpos** (char \*buff, const [solopt\\_t](#) \*opt, [sol\\_t](#) \*sol)
- void **decode\_refpos** (char \*buff, const [solopt\\_t](#) \*opt, double \*rb)
- int **decode\_sol** (char \*buff, const [solopt\\_t](#) \*opt, [sol\\_t](#) \*sol, double \*rb)
- void **decode\_solopt** (char \*buff, [solopt\\_t](#) \*opt)
- void **readsolopt** (FILE \*fp, [solopt\\_t](#) \*opt)
- int **inputsol** (unsigned char data, [gtime\\_t](#) ts, [gtime\\_t](#) te, double tint, int qflag, const [solopt\\_t](#) \*opt, [solbuf\\_t](#) \*solbuf)
- int **readsoldata** (FILE \*fp, [gtime\\_t](#) ts, [gtime\\_t](#) te, double tint, int qflag, const [solopt\\_t](#) \*opt, [solbuf\\_t](#) \*solbuf)
- int **cmpsol** (const void \*p1, const void \*p2)
- int **sort\_solbuf** ([solbuf\\_t](#) \*solbuf)
- int **readsolt** (char \*files[], int nfile, [gtime\\_t](#) ts, [gtime\\_t](#) te, double tint, int qflag, [solbuf\\_t](#) \*solbuf)
- int **readsol** (char \*files[], int nfile, [solbuf\\_t](#) \*sol)
- int **addsol** ([solbuf\\_t](#) \*solbuf, const [sol\\_t](#) \*sol)
- [sol\\_t](#) \* **getsol** ([solbuf\\_t](#) \*solbuf, int index)
- void **initsolbuf** ([solbuf\\_t](#) \*solbuf, int cyclic, int nmax)
- void **freesolbuf** ([solbuf\\_t](#) \*solbuf)
- void **freesolstatbuf** ([solstatbuf\\_t](#) \*solstatbuf)
- int **cmpsolstat** (const void \*p1, const void \*p2)
- int **sort\_solstat** ([solstatbuf\\_t](#) \*statbuf)
- int **decode\_solstat** (char \*buff, [solstat\\_t](#) \*stat)
- void **addsolstat** ([solstatbuf\\_t](#) \*statbuf, const [solstat\\_t](#) \*stat)
- int **readsolstatdata** (FILE \*fp, [gtime\\_t](#) ts, [gtime\\_t](#) te, double tint, [solstatbuf\\_t](#) \*statbuf)
- int **readsolstatt** (char \*files[], int nfile, [gtime\\_t](#) ts, [gtime\\_t](#) te, double tint, [solstatbuf\\_t](#) \*statbuf)
- int **readsolstat** (char \*files[], int nfile, [solstatbuf\\_t](#) \*statbuf)
- int **outecef** (unsigned char \*buff, const char \*s, const [sol\\_t](#) \*sol, const [solopt\\_t](#) \*opt)
- int **outpos** (unsigned char \*buff, const char \*s, const [sol\\_t](#) \*sol, const [solopt\\_t](#) \*opt)
- int **outenu** (unsigned char \*buff, const char \*s, const [sol\\_t](#) \*sol, const double \*rb, const [solopt\\_t](#) \*opt)
- int **outnmea\_rmc** (unsigned char \*buff, const [sol\\_t](#) \*sol)
- int **outnmea\_gga** (unsigned char \*buff, const [sol\\_t](#) \*sol)
- int **outnmea\_gsa** (unsigned char \*buff, const [sol\\_t](#) \*sol, const [ssat\\_t](#) \*ssat)
- int **outnmea\_gsv** (unsigned char \*buff, const [sol\\_t](#) \*sol, const [ssat\\_t](#) \*ssat)
- int **outprcopts** (unsigned char \*buff, const [prcopt\\_t](#) \*opt)
- int **outsolheads** (unsigned char \*buff, const [solopt\\_t](#) \*opt)
- int **outsols** (unsigned char \*buff, const [sol\\_t](#) \*sol, const double \*rb, const [solopt\\_t](#) \*opt)
- int **outsolsexs** (unsigned char \*buff, const [sol\\_t](#) \*sol, const [ssat\\_t](#) \*ssat, const [solopt\\_t](#) \*opt)
- void **outprcopt** (FILE \*fp, const [prcopt\\_t](#) \*opt)
- void **outsolhead** (FILE \*fp, const [solopt\\_t](#) \*opt)
- void **outsol** (FILE \*fp, const [sol\\_t](#) \*sol, const double \*rb, const [solopt\\_t](#) \*opt)
- void **outsolsex** (FILE \*fp, const [sol\\_t](#) \*sol, const [ssat\\_t](#) \*ssat, const [solopt\\_t](#) \*opt)

## Variables

- const int **TINTACT** = 200  
*period for stream active (ms)*
- const int **SERIBUFFSIZE** = 4096  
*serial buffer size (bytes)*
- const int **TIMETAGH\_LEN** = 64  
*time tag file header length*
- const int **MAXCLI** = 32  
*max client connection for tcp svr*
- const int **MAXSTATMSG** = 32  
*max length of status message*
- const int **FTP\_TIMEOUT** = 30

- ftp/http timeout (s)*
- const int **MAXRAWLEN** = 4096  
*max length of receiver raw message*
- const int **MAXSOLBUF** = 256  
*max number of solution buffer*
- const int **MAXSBSMSG** = 32  
*max number of SBAS msg in RTK server*
- const int **MAXOBSBUF** = 128  
*max number of observation data buffer*
- const int **FILEPATHSEP** = '/'
- const double **RE\_WGS84** = 6378137.0  
*earth semimajor axis (WGS84) (m)*
- const double **FE\_WGS84** = (1.0 / 298.257223563)  
*earth flattening (WGS84)*
- const double **HION** = 350000.0  
*ionosphere height (m)*
- const double **PRN\_HWBIAS** = 1e-6  
*process noise of h/w bias (m/MHz/sqrt(s))*
- const double **INT\_SWAP\_STAT** = 86400.0  
*swap interval of solution status file (s)*
- const double **INT\_SWAP\_TRAC** = 86400.0  
*swap interval of trace file (s)*
- const unsigned int **POLYCRC32** = 0xEDB88320u  
*CRC32 polynomial.*
- const unsigned int **POLYCRC24Q** = 0x1864CFBu  
*CRC24Q polynomial.*
- const int **PMODE\_SINGLE** = 0  
*positioning mode: single*
- const int **PMODE\_DGPS** = 1  
*positioning mode: DGPS/DGNSS*
- const int **PMODE\_KINEMA** = 2  
*positioning mode: kinematic*
- const int **PMODE\_STATIC** = 3  
*positioning mode: static*
- const int **PMODE\_MOVEB** = 4  
*positioning mode: moving-base*
- const int **PMODE\_FIXED** = 5  
*positioning mode: fixed*
- const int **PMODE\_PPP\_KINEMA** = 6  
*positioning mode: PPP-kinematic*
- const int **PMODE\_PPP\_STATIC** = 7  
*positioning mode: PPP-static*
- const int **PMODE\_PPP\_FIXED** = 8  
*positioning mode: PPP-fixed*
- const int **SOLF\_LLH** = 0  
*solution format: lat/lon/height*
- const int **SOLF\_XYZ** = 1  
*solution format: x/y/z-ecef*
- const int **SOLF\_ENU** = 2  
*solution format: e/n/u-baseline*
- const int **SOLF\_NMEA** = 3

- solution format: NMEA-183*
- const int SOLF\_STAT = 4
  - solution format: solution status*
- const int SOLF\_GSIF = 5
  - solution format: GSI F1/F2*
- const int SOLQ\_NONE = 0
  - solution status: no solution*
- const int SOLQ\_FIX = 1
  - solution status: fix*
- const int SOLQ\_FLOAT = 2
  - solution status: float*
- const int SOLQ\_SBAS = 3
  - solution status: SBAS*
- const int SOLQ\_DGPS = 4
  - solution status: DGPS/DGNSS*
- const int SOLQ\_SINGLE = 5
  - solution status: single*
- const int SOLQ\_PPP = 6
  - solution status: PPP*
- const int SOLQ\_DR = 7
  - solution status: dead reckoning*
- const int MAXSOLQ = 7
  - max number of solution status*
- const int TIMES\_GPST = 0
  - time system: gps time*
- const int TIMES\_UTC = 1
  - time system: utc*
- const int TIMES\_JST = 2
  - time system: jst*
- const double ERR\_SAAS = 0.3
  - saastamoinen model error std (m)*
- const double ERR\_BRDCI = 0.5
  - broadcast iono model error factor*
- const double ERR\_CBIAS = 0.3
  - code bias error std (m)*
- const double REL\_HUMI = 0.7
  - relative humidity for saastamoinen model*
- const double GAP\_RESION = 120
  - default gap to reset ionos parameters (ep)*
- const int MAXFREQ = 7
  - max NFREQ*
- const int MAXLEAPS = 64
  - max number of leap seconds table*
- const double DTTOL = 0.005
  - tolerance of time difference (s)*
- const int NFREQ = 3
  - number of carrier frequencies*
- const int NFREQGLO = 2
  - number of carrier frequencies of GLONASS*
- const int NEXOBS = 0
  - number of extended obs codes*

- const int `MAXANT` = 64  
*max length of station name/antenna type*
- const int `MINPRNGPS` = 1  
*min satellite PRN number of GPS*
- const int `MAXPRNGPS` = 32  
*max satellite PRN number of GPS*
- const int `NSATGPS` = (`MAXPRNGPS` - `MINPRNGPS` + 1)  
*number of GPS satellites*
- const int `NSYSGPS` = 1
- const int `SYS_NONE` = 0x00  
*navigation system: none*
- const int `SYS_GPS` = 0x01  
*navigation system: GPS*
- const int `SYS_SBS` = 0x02  
*navigation system: SBAS*
- const int `SYS_GLO` = 0x04  
*navigation system: GLONASS*
- const int `SYS_GAL` = 0x08  
*navigation system: Galileo*
- const int `SYS_QZS` = 0x10  
*navigation system: QZSS*
- const int `SYS_BDS` = 0x20  
*navigation system: BeiDou*
- const int `SYS_IRN` = 0x40  
*navigation system: IRNS*
- const int `SYS_LEO` = 0x80  
*navigation system: LEO*
- const int `SYS_ALL` = 0xFF  
*navigation system: all*
- const int `MINPRNGLO` = 1  
*min satellite slot number of GLONASS*
- const int `MAXPRNGLO` = 27  
*max satellite slot number of GLONASS*
- const int `NSATGLO` = (`MAXPRNGLO` - `MINPRNGLO` + 1)  
*number of GLONASS satellites*
- const int `NSYSGLO` = 1
- const int `MINPRNGAL` = 1  
*min satellite PRN number of Galileo*
- const int `MAXPRNGAL` = 36  
*max satellite PRN number of Galileo*
- const int `NSATGAL` = (`MAXPRNGAL` - `MINPRNGAL` + 1)  
*number of Galileo satellites*
- const int `NSYSGAL` = 1
- const int `MAXPRNQZS` = 199  
*max satellite PRN number of QZSS*
- const int `MINPRNQZS` = 193  
*min satellite PRN number of QZSS*
- const int `MINPRNQZS_S` = 0
- const int `NSATQZS` = 0
- const int `NSYSQZS` = 0
- const int `MINPRNBDS` = 1

```

    min satellite sat number of BeiDou
• const int MAXPRNBDS = 63
    max satellite sat number of BeiDou
• const int NSATBDS = (MAXPRNBDS - MINPRNBDS + 1)
    number of BeiDou satellites
• const int NSYSBDS = 1
• const int MINPRNIRN = 1
    min satellite sat number of IRNSS
• const int MAXPRNIRN = 7
    max satellite sat number of IRNSS
• const int NSATIRN = 0
• const int NSYSIRN = 0
• const int MINPRNLEO = 1
    min satellite sat number of LEO
• const int MAXPRNLEO = 10
    max satellite sat number of LEO */
• const int NSATLEO = 0
• const int NSYSLEO = 0
• const int NSYS = (NSYSGPS + NSYSGLO + NSYSGAL + NSYSQZS + NSYSBDS + NSYSIRN + NSYSLEO)
    number of systems
• const int MINPRNSBS = 120
    min satellite PRN number of SBAS
• const int MAXPRNSBS = 142
    max satellite PRN number of SBAS
• const int NSATSBS = (MAXPRNSBS - MINPRNSBS + 1)
    number of SBAS satellites
• const int MAXSAT = (NSATGPS + NSATGLO + NSATGAL + NSATQZS + NSATBDS + NSATIRN + NSATSBS
+ NSATLEO)
• const int MAXSTA = 255
• const int MAXOBS = 64
    max number of obs in an epoch
• const int MAXRCV = 64
    max receiver number (1 to MAXRCV)
• const int MAXOBSTYPE = 64
    max number of obs type in RINEX
• const double MAXDTOE = 7200.0
    max time difference to GPS Toe (s)
• const double MAXDTOE_QZS = 7200.0
    max time difference to QZSS Toe (s)
• const double MAXDTOE_GAL = 10800.0
    max time difference to Galileo Toe (s)
• const double MAXDTOE_BDS = 21600.0
    max time difference to BeiDou Toe (s)
• const double MAXDTOE_GLO = 1800.0
    max time difference to GLONASS Toe (s)
• const double MAXDTOE_SBS = 360.0
    max time difference to SBAS Toe (s)
• const double MAXDTOE_S = 86400.0
    max time difference to ephemeris toe (s) for other
• const double MAXGDOP = 300.0
    max GDOP

```

- const int [MAXSBSURA](#) = 8  
*max URA of SBAS satellite*
- const int [MAXBAND](#) = 10  
*max SBAS band of IGP*
- const int [MAXNIGP](#) = 201  
*max number of IGP in SBAS band*
- const int [MAXNGEO](#) = 4  
*max number of GEO satellites*
- const int [MAXSOLMSG](#) = 8191  
*max length of solution message*
- const int [MAXERRMSG](#) = 4096  
*max length of error/warning message*
- const int [IONOOPT\\_OFF](#) = 0  
*ionosphere option: correction off*
- const int [IONOOPT\\_BRDC](#) = 1  
*ionosphere option: broadcast model*
- const int [IONOOPT\\_SBAS](#) = 2  
*ionosphere option: SBAS model*
- const int [IONOOPT\\_IFLC](#) = 3  
*ionosphere option: L1/L2 or L1/L5 iono-free LC*
- const int [IONOOPT\\_EST](#) = 4  
*ionosphere option: estimation*
- const int [IONOOPT\\_TEC](#) = 5  
*ionosphere option: IONEX TEC model*
- const int [IONOOPT\\_QZS](#) = 6  
*ionosphere option: QZSS broadcast model*
- const int [IONOOPT\\_LEX](#) = 7  
*ionosphere option: QZSS LEX ionosphere*
- const int [IONOOPT\\_STEC](#) = 8  
*ionosphere option: SLANT TEC model*
- const int [TROPOPT\\_OFF](#) = 0  
*troposphere option: correction off*
- const int [TROPOPT\\_SAAS](#) = 1  
*troposphere option: Saastamoinen model*
- const int [TROPOPT\\_SBAS](#) = 2  
*troposphere option: SBAS model*
- const int [TROPOPT\\_EST](#) = 3  
*troposphere option: ZTD estimation*
- const int [TROPOPT\\_ESTG](#) = 4  
*troposphere option: ZTD+grad estimation*
- const int [TROPOPT\\_COR](#) = 5  
*troposphere option: ZTD correction*
- const int [TROPOPT\\_CORG](#) = 6  
*troposphere option: ZTD+grad correction*
- const int [EPHOPT\\_BRDC](#) = 0  
*ephemeris option: broadcast ephemeris*
- const int [EPHOPT\\_PREC](#) = 1  
*ephemeris option: precise ephemeris*
- const int [EPHOPT\\_SBAS](#) = 2  
*ephemeris option: broadcast + SBAS*
- const int [EPHOPT\\_SSRAPC](#) = 3

- ephemeris option: broadcast + SSR\_APC*
- const int `EPHOPT_SSR`COM = 4
- ephemeris option: broadcast + SSR\_COM*
- const int `EPHOPT_LEX` = 5
- ephemeris option: QZSS LEX ephemeris*
- const double `EFACT_GPS` = 1.0
- error factor: GPS*
- const double `EFACT_GLO` = 1.5
- error factor: GLONASS*
- const double `EFACT_GAL` = 1.0
- error factor: Galileo*
- const double `EFACT_QZS` = 1.0
- error factor: QZSS*
- const double `EFACT_BDS` = 1.0
- error factor: BeiDou*
- const double `EFACT_IRN` = 1.5
- error factor: IRNSS*
- const double `EFACT_SBS` = 3.0
- error factor: SBAS*
- const int `MAXEXFILE` = 1024
- max number of expanded files*
- const double `MAXSBSAGEF` = 30.0
- max age of SBAS fast correction (s)*
- const double `MAXSBSAGEL` = 1800.0
- max age of SBAS long term corr (s)*
- const int `ARMODE_OFF` = 0
- AR mode: off.*
- const int `ARMODE_CONT` = 1
- AR mode: continuous.*
- const int `ARMODE_INST` = 2
- AR mode: instantaneous.*
- const int `ARMODE_FIXHOLD` = 3
- AR mode: fix and hold.*
- const int `ARMODE_PPPAR` = 4
- AR mode: PPP-AR.*
- const int `ARMODE_PPPAR_ILS` = 5
- AR mode: AR mode: PPP-AR ILS.*
- const int `ARMODE_WLNL` = 6
- const int `ARMODE_TCAR` = 7
- const int `POSOPT_RINEX` = 3
- pos option: rinex header pos*
- const int `MAXSTRPATH` = 1024
- max length of stream path*
- const int `MAXSTRMSG` = 1024
- max length of stream message*
- const double `CHISQR` [100]
- const double `LAM_CARR` [`MAXFREQ`]
- const int `STRFMT_RTCM2` = 0
- const int `STRFMT_RTCM3` = 1
- const int `STRFMT_SP3` = 16
- const int `STRFMT_RNXCLK` = 17

- const int STRFMT\_SBAS = 18
- const int STRFMT\_NMEA = 19
- const int MAXSTRRTK = 8
- const double VAR\_POS = std::pow(30.0, 2.0)
- const double VAR\_VEL = std::pow(10.0, 2.0)
- const double VAR\_ACC = std::pow(10.0, 2.0)
- const double VAR\_HWBIAS = std::pow(1.0, 2.0)
- const double VAR\_GRA = std::pow(0.001, 2.0)
- const double INIT\_ZWD = 0.15
- const double PRN\_HWBIA = 1E-6
- const double MAXAC = 30.0
- const double VAR\_HOLDAMB = 0.001
- const double TTOL\_MOVEB = (1.0 + 2 \* DTTOL)

### 9.7.2.1 Detailed Description

Our version of the RTKLIB core library (see <http://www.rtklib.com/>)

### 9.7.2.2 Macro Definition Documentation

#### 9.7.2.2.1 closesocket

```
#define closesocket close
```

Definition at line 59 of file [rtklib.h](#).

#### 9.7.2.2.2 COMMENTH

```
#define COMMENTH "%" /* comment line indicator for solution */
```

Definition at line 43 of file [rtklib\\_solution.h](#).

#### 9.7.2.2.3 dev\_t

```
#define dev_t int
```

Definition at line 57 of file [rtklib.h](#).

#### 9.7.2.2.4 ENABDS

```
#define ENABDS
```

Definition at line 202 of file [rtklib.h](#).



#### 9.7.2.2.5 ENAGLO

```
#define ENAGLO
```

Definition at line 170 of file [rtklib.h](#).

#### 9.7.2.2.6 FTP\_CMD

```
#define FTP_CMD "wget" /* ftp/http command */
```

Definition at line 78 of file [rtklib.h](#).

#### 9.7.2.2.7 IB\_PPP

```
#define IB_PPP(  
    s,  
    opt)
```

##### Value:

```
(NR_PPP(opt) + (s)-1) /* state index of  
    phase bias */
```

Definition at line 342 of file [rtklib.h](#).

#### 9.7.2.2.8 IB\_RTK

```
#define IB_RTK(  
    s,  
    f,  
    opt)
```

##### Value:

```
(NR_RTK(opt) + MAXSAT * (f) + (s)-1) /* phase bias (s:satno,f:freq) */
```

Definition at line 68 of file [rtklib\\_rtkpos.h](#).

#### 9.7.2.2.9 IC\_PPP

```
#define IC_PPP(  
    s,  
    opt)
```

##### Value:

```
(NP_PPP(opt) + (s)) /* state index of  
    clocks (s=0:gps,1:glo) */
```

Definition at line 339 of file [rtklib.h](#).

**9.7.2.2.10 II\_RTK**

```
#define II_RTK(
    s,
    opt)
```

**Value:**

```
(NP_RTK(opt) + (s)-1) /* ionos (s:satellite no) */
```

Definition at line 65 of file [rtklib\\_rtkpos.h](#).

**9.7.2.2.11 IL\_RTK**

```
#define IL_RTK(
    f,
    opt)
```

**Value:**

```
(NP_RTK(opt) + NI_RTK(opt) + NT_RTK(opt) + (f)) /* receiver h/w bias */
```

Definition at line 67 of file [rtklib\\_rtkpos.h](#).

**9.7.2.2.12 initlock**

```
#define initlock(
    f)
```

**Value:**

```
pthread_mutex_init(f, NULL)
```

Definition at line 61 of file [rtklib.h](#).

**9.7.2.2.13 IT\_PPP**

```
#define IT_PPP(
    opt)
```

**Value:**

```
(IC_PPP(0, opt) + NSYS) /* state index
of tropos */
```

Definition at line 340 of file [rtklib.h](#).

**9.7.2.2.14 IT\_RTK**

```
#define IT_RTK(
    r,
    opt)
```

**Value:**

```
(NP_RTK(opt) + NI_RTK(opt) + NT_RTK(opt) / 2 * (r)) /* tropos (r:0=rov,l:ref) */
```

Definition at line 66 of file [rtklib\\_rtkpos.h](#).

**9.7.2.2.15 lock\_t**

```
#define lock_t pthread_mutex_t
```

Definition at line 60 of file [rtklib.h](#).

**9.7.2.2.16 MSG\_DISCONN**

```
#define MSG_DISCONN "$_DISCONNECT\r\n" /* disconnect message */
```

Definition at line 44 of file [rtklib\\_solution.h](#).

**9.7.2.2.17 NB\_RTK**

```
#define NB_RTK(  
    opt)
```

**Value:**

```
((opt)->mode <= PMODE_DGPS ? 0 : MAXSAT * NB_RTK(opt))
```

Definition at line 351 of file [rtklib.h](#).

**9.7.2.2.18 NF\_RTK**

```
#define NF_RTK(  
    opt)
```

**Value:**

```
((opt)->ionoopt == IONOOPT_IFLC ? 1 : (opt)->nf)
```

Definition at line 346 of file [rtklib.h](#).

**9.7.2.2.19 NI\_RTK**

```
#define NI_RTK(  
    opt)
```

**Value:**

```
((opt)->ionoopt != IONOOPT_EST ? 0 : MAXSAT)
```

Definition at line 348 of file [rtklib.h](#).

**9.7.2.2.20 NL\_RTK**

```
#define NL_RTK(  
    opt)
```

**Value:**

```
((opt)->glomodear != 2 ? 0 : NFREQGLO)
```

Definition at line 350 of file [rtklib.h](#).

**9.7.2.2.21 NP\_PPP**

```
#define NP_PPP(  
    opt)
```

**Value:**

```
((opt)->dynamics ? 9 : 3) /* number of  
    pos solution */
```

Definition at line 338 of file [rtklib.h](#).

**9.7.2.2.22 NP\_RTK**

```
#define NP_RTK(  
    opt)
```

**Value:**

```
((opt)->dynamics == 0 ? 3 : 9)
```

Definition at line 347 of file [rtklib.h](#).

**9.7.2.2.23 NR\_PPP**

```
#define NR_PPP(  
    opt)
```

**Value:**

```
(IT_PPP(opt) + ((opt)->tropopt < TROPOPT_EST ? 0 : ((opt)->tropopt == TROPOPT_EST ? 1 : 3)) /* number of  
    solutions */
```

Definition at line 341 of file [rtklib.h](#).

**9.7.2.2.24 NR\_RTK**

```
#define NR_RTK(  
    opt)
```

**Value:**

```
(NP_RTK(opt) + NI_RTK(opt) + NT_RTK(opt) + NL_RTK(opt))
```

Definition at line 352 of file [rtklib.h](#).

**9.7.2.2.25 NT\_RTK**

```
#define NT_RTK(  
    opt)
```

**Value:**

```
((opt)->tropopt < TROPOPT_EST ? 0 : ((opt)->tropopt < TROPOPT_ESTG ? 2 : 6))
```

Definition at line 349 of file [rtklib.h](#).

**9.7.2.2.26 NTRIP\_AGENT**

```
#define NTRIP_AGENT "RTKLIB/" VER_RTKLIB
```

Definition at line 66 of file [rtklib.h](#).

**9.7.2.2.27 NTRIP\_CLI\_PORT**

```
#define NTRIP_CLI_PORT 2101 /* default ntrip-client connection port */
```

Definition at line 67 of file [rtklib.h](#).

**9.7.2.2.28 NTRIP\_MAXRSP**

```
#define NTRIP_MAXRSP 32768 /* max size of ntrip response */
```

Definition at line 69 of file [rtklib.h](#).

**9.7.2.2.29 NTRIP\_MAXSTR**

```
#define NTRIP_MAXSTR 256 /* max length of mountpoint string */
```

Definition at line 70 of file [rtklib.h](#).

**9.7.2.2.30 NTRIP\_RSP\_ERROR**

```
#define NTRIP_RSP_ERROR "ERROR" /* ntrip response: error */
```

Definition at line 76 of file [rtklib.h](#).

**9.7.2.2.31 NTRIP\_RSP\_HTTP**

```
#define NTRIP_RSP_HTTP "HTTP/" /* ntrip response: http */
```

Definition at line 75 of file [rtklib.h](#).

**9.7.2.2.32 NTRIP\_RSP\_OK\_CLI**

```
#define NTRIP_RSP_OK_CLI "ICY 200 OK\r\n" /* ntrip response: client */
```

Definition at line 71 of file [rtklib.h](#).

**9.7.2.2.33 NTRIP\_RSP\_OK\_SVR**

```
#define NTRIP_RSP_OK_SVR "OK\r\n" /* ntrip response: server */
```

Definition at line 72 of file [rtklib.h](#).

**9.7.2.2.34 NTRIP\_RSP\_SRCTBL**

```
#define NTRIP_RSP_SRCTBL "SOURCETABLE 200 OK\r\n" /* ntrip response: source table */
```

Definition at line 73 of file [rtklib.h](#).

**9.7.2.2.35 NTRIP\_RSP\_TBLEND**

```
#define NTRIP_RSP_TBLEND "ENDSOURCETABLE"
```

Definition at line 74 of file [rtklib.h](#).

**9.7.2.2.36 NTRIP\_SVR\_PORT**

```
#define NTRIP_SVR_PORT 80 /* default ntrip-server connection port */
```

Definition at line 68 of file [rtklib.h](#).

**9.7.2.2.37 NX\_PPP**

```
#define NX_PPP(  
    opt)
```

**Value:**

```
(IB_PPP(MAXSAT, opt) + 1) /* number of  
    estimated states */
```

Definition at line 343 of file [rtklib.h](#).

**9.7.2.2.38 NX\_RTK**

```
#define NX_RTK(  
    opt)
```

**Value:**

```
(NR_RTK(opt) + NB_RTK(opt))
```

Definition at line 353 of file [rtklib.h](#).

**9.7.2.2.39 rtk\_lock**

```
#define rtk_lock(  
    f)
```

**Value:**

```
pthread_mutex_lock(f)
```

Definition at line 62 of file [rtklib.h](#).

#### 9.7.2.2.40 rtk\_unlock

```
#define rtk_unlock(  
    f)
```

**Value:**

```
pthread_mutex_unlock(f)
```

Definition at line 63 of file [rtklib.h](#).

#### 9.7.2.2.41 socket\_t

```
#define socket_t int
```

Definition at line 58 of file [rtklib.h](#).

#### 9.7.2.2.42 STR\_FILE

```
#define STR_FILE 2 /* stream type:  file */
```

Definition at line 329 of file [rtklib.h](#).

#### 9.7.2.2.43 STR\_FTP

```
#define STR_FTP 8 /* stream type:  ftp */
```

Definition at line 335 of file [rtklib.h](#).

#### 9.7.2.2.44 STR\_HTTP

```
#define STR_HTTP 9 /* stream type:  http */
```

Definition at line 336 of file [rtklib.h](#).

#### 9.7.2.2.45 STR\_MODE\_R

```
#define STR_MODE_R 0x1 /* stream mode:  read */
```

Definition at line 323 of file [rtklib.h](#).

#### 9.7.2.2.46 STR\_MODE\_RW

```
#define STR_MODE_RW 0x3 /* stream mode:  read/write */
```

Definition at line 325 of file [rtklib.h](#).

#### 9.7.2.2.47 STR\_MODE\_W

```
#define STR_MODE_W 0x2 /* stream mode:  write */
```

Definition at line 324 of file [rtklib.h](#).

#### 9.7.2.2.48 STR\_NONE

```
#define STR_NONE 0 /* stream type:  none */
```

Definition at line 327 of file [rtklib.h](#).

#### 9.7.2.2.49 STR\_NTRIPCLI

```
#define STR_NTRIPCLI 7 /* stream type:  NTRIP client */
```

Definition at line 334 of file [rtklib.h](#).

#### 9.7.2.2.50 STR\_NTRIPSVR

```
#define STR_NTRIPSVR 6 /* stream type:  NTRIP server */
```

Definition at line 333 of file [rtklib.h](#).

#### 9.7.2.2.51 STR\_SERIAL

```
#define STR_SERIAL 1 /* stream type:  serial */
```

Definition at line 328 of file [rtklib.h](#).

#### 9.7.2.2.52 STR\_TCPCLI

```
#define STR_TCPCLI 4 /* stream type:  TCP client */
```

Definition at line 331 of file [rtklib.h](#).

#### 9.7.2.2.53 STR\_TCPSVR

```
#define STR_TCPSVR 3 /* stream type:  TCP server */
```

Definition at line 330 of file [rtklib.h](#).

#### 9.7.2.2.54 STR\_UDP

```
#define STR_UDP 5 /* stream type:  UDP stream */
```

Definition at line 332 of file [rtklib.h](#).



### 9.7.2.2.55 VER\_RTKLIB

```
#define VER_RTKLIB "2.4.2"
```

Definition at line 65 of file [rtklib.h](#).

## 9.7.2.3 Typedef Documentation

### 9.7.2.3.1 fatalfunc\_t

```
using fatalfunc_t = void(const char *)
```

fatal callback function type

Definition at line 320 of file [rtklib.h](#).

## 9.7.2.4 Function Documentation

### 9.7.2.4.1 eph\_to\_rtklib()

```
geph_t eph_to_rtklib (
    const Glonass_Gnav_Ephemeris & glonass_gnav_eph,
    const Glonass_Gnav_Utc_Model & gnav_clock_model)
```

Transforms a [Glonass\\_Gnav\\_Ephemeris](#) to its RTKLIB counterpart.

#### Parameters

<i>glonass_gnav_eph</i>	GLONASS GNAV Ephemeris structure
-------------------------	----------------------------------

#### Returns

Ephemeris structure for RTKLIB parsing

## 9.7.2.5 Variable Documentation

### 9.7.2.5.1 ARMODE\_CONT

```
const int ARMODE_CONT = 1
```

AR mode: continuous.

Definition at line 307 of file [rtklib.h](#).

### 9.7.2.5.2 ARMODE\_FIXHOLD

```
const int ARMODE_FIXHOLD = 3
```

AR mode: fix and hold.

Definition at line 309 of file [rtklib.h](#).

#### 9.7.2.5.3 ARMODE\_INST

```
const int ARMODE_INST = 2
```

AR mode: instantaneous.

Definition at line 308 of file [rtklib.h](#).

#### 9.7.2.5.4 ARMODE\_OFF

```
const int ARMODE_OFF = 0
```

AR mode: off.

Definition at line 306 of file [rtklib.h](#).

#### 9.7.2.5.5 ARMODE\_PPPAR

```
const int ARMODE_PPPAR = 4
```

AR mode: PPP-AR.

Definition at line 310 of file [rtklib.h](#).

#### 9.7.2.5.6 ARMODE\_PPPAR\_ILS

```
const int ARMODE_PPPAR_ILS = 5
```

AR mode: AR mode: PPP-AR ILS.

Definition at line 311 of file [rtklib.h](#).

#### 9.7.2.5.7 ARMODE\_TCAR

```
const int ARMODE_TCAR = 7
```

Definition at line 313 of file [rtklib.h](#).

#### 9.7.2.5.8 ARMODE\_WLNL

```
const int ARMODE_WLNL = 6
```

Definition at line 312 of file [rtklib.h](#).

#### 9.7.2.5.9 CHISQR

```
const double CHISQR[100]
```

**Initial value:**

```
= {  
    10.8, 13.8, 16.3, 18.5, 20.5, 22.5, 24.3, 26.1, 27.9, 29.6,  
    31.3, 32.9, 34.5, 36.1, 37.7, 39.3, 40.8, 42.3, 43.8, 45.3,  
    46.8, 48.3, 49.7, 51.2, 52.6, 54.1, 55.5, 56.9, 58.3, 59.7,  
    61.1, 62.5, 63.9, 65.2, 66.6, 68.0, 69.3, 70.7, 72.1, 73.4,  
    74.7, 76.0, 77.3, 78.6, 80.0, 81.3, 82.6, 84.0, 85.4, 86.7,  
    88.0, 89.3, 90.6, 91.9, 93.3, 94.7, 96.0, 97.4, 98.7, 100,  
    101, 102, 103, 104, 105, 107, 108, 109, 110, 112,  
    113, 114, 115, 116, 118, 119, 120, 122, 123, 125,  
    126, 127, 128, 129, 131, 132, 133, 134, 135, 137,  
    138, 139, 140, 142, 143, 144, 145, 147, 148, 149}
```

Definition at line 1291 of file [rtklib.h](#).

#### 9.7.2.5.10 DTTOL

```
const double DTTOL = 0.005
```

tolerance of time difference (s)

Definition at line 146 of file [rtklib.h](#).

#### 9.7.2.5.11 EFACT\_BDS

```
const double EFACT_BDS = 1.0
```

error factor: BeiDou

Definition at line 298 of file [rtklib.h](#).

#### 9.7.2.5.12 EFACT\_GAL

```
const double EFACT_GAL = 1.0
```

error factor: Galileo

Definition at line 296 of file [rtklib.h](#).

#### 9.7.2.5.13 EFACT\_GLO

```
const double EFACT_GLO = 1.5
```

error factor: GLONASS

Definition at line 295 of file [rtklib.h](#).

#### 9.7.2.5.14 EFACT\_GPS

```
const double EFACT_GPS = 1.0
```

error factor: GPS

Definition at line 294 of file [rtklib.h](#).

#### 9.7.2.5.15 EFACT\_IRN

```
const double EFACT_IRN = 1.5
```

error factor: IRNSS

Definition at line 299 of file [rtklib.h](#).

#### 9.7.2.5.16 EFACT\_QZS

```
const double EFACT_QZS = 1.0
```

error factor: QZSS

Definition at line 297 of file [rtklib.h](#).

#### 9.7.2.5.17 EFACT\_SBS

```
const double EFACT_SBS = 3.0
```

error factor: SBAS

Definition at line 300 of file [rtklib.h](#).

#### 9.7.2.5.18 EPHOPT\_BRDC

```
const int EPHOPT_BRDC = 0
```

ephemeris option: broadcast ephemeris

Definition at line 287 of file [rtklib.h](#).

#### 9.7.2.5.19 EPHOPT\_LEX

```
const int EPHOPT_LEX = 5
```

ephemeris option: QZSS LEX ephemeris

Definition at line 292 of file [rtklib.h](#).

#### 9.7.2.5.20 EPHOPT\_PREC

```
const int EPHOPT_PREC = 1
```

ephemeris option: precise ephemeris

Definition at line 288 of file [rtklib.h](#).

#### 9.7.2.5.21 EPHOPT\_SBAS

```
const int EPHOPT_SBAS = 2
```

ephemeris option: broadcast + SBAS

Definition at line 289 of file [rtklib.h](#).

#### 9.7.2.5.22 EPHOPT\_SSRAPC

```
const int EPHOPT_SSRAPC = 3
```

ephemeris option: broadcast + SSR\_APC

Definition at line 290 of file [rtklib.h](#).

#### 9.7.2.5.23 EPHOPT\_SSRCOM

```
const int EPHOPT_SSRCOM = 4
```

ephemeris option: broadcast + SSR\_COM

Definition at line 291 of file [rtklib.h](#).

#### 9.7.2.5.24 ERR\_BRDCI

```
const double ERR_BRDCI = 0.5
```

broadcast iono model error factor

Definition at line 138 of file [rtklib.h](#).

#### 9.7.2.5.25 ERR\_CBIAS

```
const double ERR_CBIAS = 0.3
```

code bias error std (m)

Definition at line 139 of file [rtklib.h](#).

#### 9.7.2.5.26 ERR\_SAAS

```
const double ERR_SAAS = 0.3
```

saastamoinen model error std (m)

Definition at line 137 of file [rtklib.h](#).

#### 9.7.2.5.27 FE\_WGS84

```
const double FE_WGS84 = (1.0 / 298.257223563)
```

earth flattening (WGS84)

Definition at line 94 of file [rtklib.h](#).

#### 9.7.2.5.28 FILEPATHSEP

```
const int FILEPATHSEP = '/'
```

Definition at line 92 of file [rtklib.h](#).

#### 9.7.2.5.29 FTP\_TIMEOUT

```
const int FTP_TIMEOUT = 30
```

ftp/http timeout (s)

Definition at line 86 of file [rtklib.h](#).

#### 9.7.2.5.30 GAP\_RESION

```
const double GAP_RESION = 120
```

default gap to reset ionos parameters (ep)

Definition at line 141 of file [rtklib.h](#).

#### 9.7.2.5.31 HION

```
const double HION = 350000.0
```

ionosphere height (m)

Definition at line 96 of file [rtklib.h](#).

#### 9.7.2.5.32 INIT\_ZWD

```
const double INIT_ZWD = 0.15
```

Definition at line 50 of file [rtklib\\_rtkpos.h](#).

#### 9.7.2.5.33 INT\_SWAP\_STAT

```
const double INT_SWAP_STAT = 86400.0
```

swap interval of solution status file (s)

Definition at line 99 of file [rtklib.h](#).

#### 9.7.2.5.34 INT\_SWAP\_TRAC

```
const double INT_SWAP_TRAC = 86400.0
```

swap interval of trace file (s)

Definition at line 100 of file [rtklib.h](#).

#### 9.7.2.5.35 IONOOPT\_BRDC

```
const int IONOOPT_BRDC = 1
```

ionosphere option: broadcast model

Definition at line 269 of file [rtklib.h](#).

#### 9.7.2.5.36 IONOOPT\_EST

```
const int IONOOPT_EST = 4
```

ionosphere option: estimation

Definition at line 272 of file [rtklib.h](#).

#### 9.7.2.5.37 IONOOPT\_IFLC

```
const int IONOOPT_IFLC = 3
```

ionosphere option: L1/L2 or L1/L5 iono-free LC

Definition at line 271 of file [rtklib.h](#).

#### 9.7.2.5.38 IONOOPT\_LEX

```
const int IONOOPT_LEX = 7
```

ionosphere option: QZSS LEX ionosphere

Definition at line 275 of file [rtklib.h](#).

#### 9.7.2.5.39 IONOOPT\_OFF

```
const int IONOOPT_OFF = 0
```

ionosphere option: correction off

Definition at line 268 of file [rtklib.h](#).

#### 9.7.2.5.40 IONOOPT\_QZS

```
const int IONOOPT_QZS = 6
```

ionosphere option: QZSS broadcast model

Definition at line 274 of file [rtklib.h](#).

#### 9.7.2.5.41 IONOOPT\_SBAS

```
const int IONOOPT_SBAS = 2
```

ionosphere option: SBAS model

Definition at line 270 of file [rtklib.h](#).

#### 9.7.2.5.42 IONOOPT\_STEC

```
const int IONOOPT_STEC = 8
```

ionosphere option: SLANT TEC model

Definition at line 276 of file [rtklib.h](#).

#### 9.7.2.5.43 IONOOPT\_TEC

```
const int IONOOPT_TEC = 5
```

ionosphere option: IONEX TEC model

Definition at line 273 of file [rtklib.h](#).



#### 9.7.2.5.44 LAM\_CARR

```
const double LAM_CARR[MAXFREQ]
```

##### Initial value:

```
= {  
    SPEED_OF_LIGHT_M_S / FREQ1, SPEED_OF_LIGHT_M_S / FREQ2, SPEED_OF_LIGHT_M_S / FREQ5, SPEED_OF_LIGHT_M_S /  
    FREQ6, SPEED_OF_LIGHT_M_S / FREQ7,  
    SPEED_OF_LIGHT_M_S / FREQ8, SPEED_OF_LIGHT_M_S / FREQ9}
```

Definition at line 1304 of file [rtklib.h](#).

#### 9.7.2.5.45 MAXAC

```
const double MAXAC = 30.0
```

Definition at line 53 of file [rtklib\\_rtkpos.h](#).

#### 9.7.2.5.46 MAXANT

```
const int MAXANT = 64
```

max length of station name/antenna type

Definition at line 151 of file [rtklib.h](#).

#### 9.7.2.5.47 MAXBAND

```
const int MAXBAND = 10
```

max SBAS band of IGP

Definition at line 261 of file [rtklib.h](#).

#### 9.7.2.5.48 MAXCLI

```
const int MAXCLI = 32
```

max client connection for tcp svr

Definition at line 83 of file [rtklib.h](#).

#### 9.7.2.5.49 MAXDToe

```
const double MAXDToe = 7200.0
```

max time difference to GPS Toe (s)

Definition at line 251 of file [rtklib.h](#).

#### 9.7.2.5.50 MAXDToe\_BDS

```
const double MAXDToe_BDS = 21600.0
```

max time difference to BeiDou Toe (s)

Definition at line 254 of file [rtklib.h](#).

#### 9.7.2.5.51 MAXDToe\_GAL

```
const double MAXDToe_GAL = 10800.0
```

max time difference to Galileo Toe (s)

Definition at line 253 of file [rtklib.h](#).

#### 9.7.2.5.52 MAXDToe\_GLO

```
const double MAXDToe_GLO = 1800.0
```

max time difference to GLONASS Toe (s)

Definition at line 255 of file [rtklib.h](#).

#### 9.7.2.5.53 MAXDToe\_QZS

```
const double MAXDToe_QZS = 7200.0
```

max time difference to QZSS Toe (s)

Definition at line 252 of file [rtklib.h](#).

#### 9.7.2.5.54 MAXDToe\_S

```
const double MAXDToe_S = 86400.0
```

max time difference to ephemeris toe (s) for other

Definition at line 257 of file [rtklib.h](#).

#### 9.7.2.5.55 MAXDToe\_SBS

```
const double MAXDToe_SBS = 360.0
```

max time difference to SBAS Toe (s)

Definition at line 256 of file [rtklib.h](#).

**9.7.2.5.56 MAXERRMSG**

```
const int MAXERRMSG = 4096
```

max length of error/warning message

Definition at line 266 of file [rtklib.h](#).

**9.7.2.5.57 MAXEXFILE**

```
const int MAXEXFILE = 1024
```

max number of expanded files

Definition at line 302 of file [rtklib.h](#).

**9.7.2.5.58 MAXFREQ**

```
const int MAXFREQ = 7
```

max NFREQ

Definition at line 143 of file [rtklib.h](#).

**9.7.2.5.59 MAXGDOP**

```
const double MAXGDOP = 300.0
```

max GDOP

Definition at line 258 of file [rtklib.h](#).

**9.7.2.5.60 MAXLEAPS**

```
const int MAXLEAPS = 64
```

max number of leap seconds table

Definition at line 145 of file [rtklib.h](#).

**9.7.2.5.61 MAXNGEO**

```
const int MAXNGEO = 4
```

max number of GEO satellites

Definition at line 263 of file [rtklib.h](#).

#### 9.7.2.5.62 MAXNIGP

```
const int MAXNIGP = 201
```

max number of IGP in SBAS band

Definition at line 262 of file [rtklib.h](#).

#### 9.7.2.5.63 MAXOBS

```
const int MAXOBS = 64
```

max number of obs in an epoch

Definition at line 246 of file [rtklib.h](#).

#### 9.7.2.5.64 MAXOBSBUF

```
const int MAXOBSBUF = 128
```

max number of observation data buffer

Definition at line 90 of file [rtklib.h](#).

#### 9.7.2.5.65 MAXOBSTYPE

```
const int MAXOBSTYPE = 64
```

max number of obs type in RINEX

Definition at line 250 of file [rtklib.h](#).

#### 9.7.2.5.66 MAXPRNBDS

```
const int MAXPRNBDS = 63
```

max satellite sat number of BeiDou

Definition at line 205 of file [rtklib.h](#).

#### 9.7.2.5.67 MAXPRNGAL

```
const int MAXPRNGAL = 36
```

max satellite PRN number of Galileo

Definition at line 185 of file [rtklib.h](#).

**9.7.2.5.68 MAXPRNGLO**

```
const int MAXPRNGLO = 27
```

max satellite slot number of GLONASS

Definition at line 173 of file [rtklib.h](#).

**9.7.2.5.69 MAXPRNGPS**

```
const int MAXPRNGPS = 32
```

max satellite PRN number of GPS

Definition at line 154 of file [rtklib.h](#).

**9.7.2.5.70 MAXPRNIRN**

```
const int MAXPRNIRN = 7
```

max satellite sat number of IRNSS

Definition at line 216 of file [rtklib.h](#).

**9.7.2.5.71 MAXPRNLEO**

```
const int MAXPRNLEO = 10
```

max satellite sat number of LEO \*/

Definition at line 226 of file [rtklib.h](#).

**9.7.2.5.72 MAXPRNQZS**

```
const int MAXPRNQZS = 199
```

max satellite PRN number of QZSS

Definition at line 189 of file [rtklib.h](#).

**9.7.2.5.73 MAXPRNSBS**

```
const int MAXPRNSBS = 142
```

max satellite PRN number of SBAS

Definition at line 238 of file [rtklib.h](#).

#### 9.7.2.5.74 MAXRAWLEN

```
const int MAXRAWLEN = 4096
```

max length of receiver raw message

Definition at line 87 of file [rtklib.h](#).

#### 9.7.2.5.75 MAXRCV

```
const int MAXRCV = 64
```

max receiver number (1 to MAXRCV)

Definition at line 249 of file [rtklib.h](#).

#### 9.7.2.5.76 MAXSAT

```
const int MAXSAT = (NSATGPS + NSATGLO + NSATGAL + NSATQZS + NSATBDS + NSATIRN + NSATSBS +  
NSATLEO)
```

Definition at line 241 of file [rtklib.h](#).

#### 9.7.2.5.77 MAXSBSAGEF

```
const double MAXSBSAGEF = 30.0
```

max age of SBAS fast correction (s)

Definition at line 303 of file [rtklib.h](#).

#### 9.7.2.5.78 MAXSBSAGEL

```
const double MAXSBSAGEL = 1800.0
```

max age of SBAS long term corr (s)

Definition at line 304 of file [rtklib.h](#).

#### 9.7.2.5.79 MAXSBSMSG

```
const int MAXSBSMSG = 32
```

max number of SBAS msg in RTK server

Definition at line 89 of file [rtklib.h](#).

**9.7.2.5.80 MAXSBSURA**

```
const int MAXSBSURA = 8
```

max URA of SBAS satellite

Definition at line 260 of file [rtklib.h](#).

**9.7.2.5.81 MAXSOLBUF**

```
const int MAXSOLBUF = 256
```

max number of solution buffer

Definition at line 88 of file [rtklib.h](#).

**9.7.2.5.82 MAXSOLMSG**

```
const int MAXSOLMSG = 8191
```

max length of solution message

Definition at line 265 of file [rtklib.h](#).

**9.7.2.5.83 MAXSOLQ**

```
const int MAXSOLQ = 7
```

max number of solution status

Definition at line 130 of file [rtklib.h](#).

**9.7.2.5.84 MAXSTA**

```
const int MAXSTA = 255
```

Definition at line 243 of file [rtklib.h](#).

**9.7.2.5.85 MAXSTATMSG**

```
const int MAXSTATMSG = 32
```

max length of status message

Definition at line 84 of file [rtklib.h](#).

#### 9.7.2.5.86 MAXSTRMSG

```
const int MAXSTRMSG = 1024
```

max length of stream message

Definition at line 318 of file [rtklib.h](#).

#### 9.7.2.5.87 MAXSTRPATH

```
const int MAXSTRPATH = 1024
```

max length of stream path

Definition at line 317 of file [rtklib.h](#).

#### 9.7.2.5.88 MAXSTRRTK

```
const int MAXSTRRTK = 8
```

Definition at line 1316 of file [rtklib.h](#).

#### 9.7.2.5.89 MINPRNBDS

```
const int MINPRNBDS = 1
```

min satellite sat number of BeiDou

Definition at line 204 of file [rtklib.h](#).

#### 9.7.2.5.90 MINPRNGAL

```
const int MINPRNGAL = 1
```

min satellite PRN number of Galileo

Definition at line 184 of file [rtklib.h](#).

#### 9.7.2.5.91 MINPRNGLO

```
const int MINPRNGLO = 1
```

min satellite slot number of GLONASS

Definition at line 172 of file [rtklib.h](#).



**9.7.2.5.92 MINPRNGPS**

```
const int MINPRNGPS = 1
```

min satellite PRN number of GPS

Definition at line 153 of file [rtklib.h](#).

**9.7.2.5.93 MINPRNIRN**

```
const int MINPRNIRN = 1
```

min satellite sat number of IRNSS

Definition at line 215 of file [rtklib.h](#).

**9.7.2.5.94 MINPRNLEO**

```
const int MINPRNLEO = 1
```

min satellite sat number of LEO

Definition at line 225 of file [rtklib.h](#).

**9.7.2.5.95 MINPRNQZS**

```
const int MINPRNQZS = 193
```

min satellite PRN number of QZSS

Definition at line 190 of file [rtklib.h](#).

**9.7.2.5.96 MINPRNQZS\_S**

```
const int MINPRNQZS_S = 0
```

Definition at line 197 of file [rtklib.h](#).

**9.7.2.5.97 MINPRNSBS**

```
const int MINPRNSBS = 120
```

min satellite PRN number of SBAS

Definition at line 237 of file [rtklib.h](#).

#### 9.7.2.5.98 NEXOBS

```
const int NEXOBS = 0
```

number of extended obs codes

Definition at line 150 of file [rtklib.h](#).

#### 9.7.2.5.99 NFREQ

```
const int NFREQ = 3
```

number of carrier frequencies

Definition at line 148 of file [rtklib.h](#).

#### 9.7.2.5.100 NFREQGLO

```
const int NFREQGLO = 2
```

number of carrier frequencies of GLONASS

Definition at line 149 of file [rtklib.h](#).

#### 9.7.2.5.101 NSATBDS

```
const int NSATBDS = (MAXPRNBDS - MINPRNBDS + 1)
```

number of BeiDou satellites

Definition at line 206 of file [rtklib.h](#).

#### 9.7.2.5.102 NSATGAL

```
const int NSATGAL = (MAXPRNGAL - MINPRNGAL + 1)
```

number of Galileo satellites

Definition at line 186 of file [rtklib.h](#).

#### 9.7.2.5.103 NSATGLO

```
const int NSATGLO = (MAXPRNGLO - MINPRNGLO + 1)
```

number of GLONASS satellites

Definition at line 174 of file [rtklib.h](#).

#### 9.7.2.5.104 NSATGPS

```
const int NSATGPS = (MAXPRNGPS - MINPRNGPS + 1)
```

number of GPS satellites

Definition at line 155 of file [rtklib.h](#).

#### 9.7.2.5.105 NSATIRN

```
const int NSATIRN = 0
```

Definition at line 221 of file [rtklib.h](#).

#### 9.7.2.5.106 NSATLEO

```
const int NSATLEO = 0
```

Definition at line 231 of file [rtklib.h](#).

#### 9.7.2.5.107 NSATQZS

```
const int NSATQZS = 0
```

Definition at line 198 of file [rtklib.h](#).

#### 9.7.2.5.108 NSATSBS

```
const int NSATSBS = (MAXPRNSBS - MINPRNSBS + 1)
```

number of SBAS satellites

Definition at line 239 of file [rtklib.h](#).

#### 9.7.2.5.109 NSYS

```
const int NSYS = (NSYSGPS + NSYSGLO + NSYSGAL + NSYSQZS + NSYSBDS + NSYSIRN + NSYSLEO)
```

number of systems

Definition at line 235 of file [rtklib.h](#).

#### 9.7.2.5.110 NSYSBDS

```
const int NSYSBDS = 1
```

Definition at line 207 of file [rtklib.h](#).

#### 9.7.2.5.111 NSYSGAL

```
const int NSYSGAL = 1
```

Definition at line 187 of file [rtklib.h](#).

#### 9.7.2.5.112 NSYSGLO

```
const int NSYSGLO = 1
```

Definition at line 175 of file [rtklib.h](#).

#### 9.7.2.5.113 NSYSGPS

```
const int NSYSGPS = 1
```

Definition at line 156 of file [rtklib.h](#).

#### 9.7.2.5.114 NSYSIRN

```
const int NSYSIRN = 0
```

Definition at line 222 of file [rtklib.h](#).

#### 9.7.2.5.115 NSYSLEO

```
const int NSYSLEO = 0
```

Definition at line 232 of file [rtklib.h](#).

#### 9.7.2.5.116 NSYSQZS

```
const int NSYSQZS = 0
```

Definition at line 199 of file [rtklib.h](#).

#### 9.7.2.5.117 PMODE\_DGPS

```
const int PMODE_DGPS = 1
```

positioning mode: DGPS/DGNSS

Definition at line 106 of file [rtklib.h](#).

**9.7.2.5.118 PMODE\_FIXED**

```
const int PMODE_FIXED = 5
```

positioning mode: fixed

Definition at line 110 of file [rtklib.h](#).

**9.7.2.5.119 PMODE\_KINEMA**

```
const int PMODE_KINEMA = 2
```

positioning mode: kinematic

Definition at line 107 of file [rtklib.h](#).

**9.7.2.5.120 PMODE\_MOVEB**

```
const int PMODE_MOVEB = 4
```

positioning mode: moving-base

Definition at line 109 of file [rtklib.h](#).

**9.7.2.5.121 PMODE\_PPP\_FIXED**

```
const int PMODE_PPP_FIXED = 8
```

positioning mode: PPP-fixed

Definition at line 113 of file [rtklib.h](#).

**9.7.2.5.122 PMODE\_PPP\_KINEMA**

```
const int PMODE_PPP_KINEMA = 6
```

positioning mode: PPP-kinematic

Definition at line 111 of file [rtklib.h](#).

**9.7.2.5.123 PMODE\_PPP\_STATIC**

```
const int PMODE_PPP_STATIC = 7
```

positioning mode: PPP-static

Definition at line 112 of file [rtklib.h](#).

#### 9.7.2.5.124 PMODE\_SINGLE

```
const int PMODE_SINGLE = 0
```

positioning mode: single

Definition at line 105 of file [rtklib.h](#).

#### 9.7.2.5.125 PMODE\_STATIC

```
const int PMODE_STATIC = 3
```

positioning mode: static

Definition at line 108 of file [rtklib.h](#).

#### 9.7.2.5.126 POLYCRC24Q

```
const unsigned int POLYCRC24Q = 0x1864CFBu
```

CRC24Q polynomial.

Definition at line 103 of file [rtklib.h](#).

#### 9.7.2.5.127 POLYCRC32

```
const unsigned int POLYCRC32 = 0xEDB88320u
```

CRC32 polynomial.

Definition at line 102 of file [rtklib.h](#).

#### 9.7.2.5.128 POSOPT\_RINEX

```
const int POSOPT_RINEX = 3
```

pos option: rinex header pos

Definition at line 316 of file [rtklib.h](#).

#### 9.7.2.5.129 PRN\_HWBIA

```
const double PRN_HWBIA = 1E-6
```

Definition at line 52 of file [rtklib\\_rtkpos.h](#).

**9.7.2.5.130 PRN\_HWBIAS**

```
const double PRN_HWBIAS = 1e-6
```

process noise of h/w bias (m/MHz/sqrt(s))

Definition at line 97 of file [rtklib.h](#).

**9.7.2.5.131 RE\_WGS84**

```
const double RE_WGS84 = 6378137.0
```

earth semimajor axis (WGS84) (m)

Definition at line 93 of file [rtklib.h](#).

**9.7.2.5.132 REL\_HUMI**

```
const double REL_HUMI = 0.7
```

relative humidity for saastamoinen model

Definition at line 140 of file [rtklib.h](#).

**9.7.2.5.133 SERIBUFFSIZE**

```
const int SERIBUFFSIZE = 4096
```

serial buffer size (bytes)

Definition at line 81 of file [rtklib.h](#).

**9.7.2.5.134 SOLF\_ENU**

```
const int SOLF_ENU = 2
```

solution format: e/n/u-baseline

Definition at line 117 of file [rtklib.h](#).

**9.7.2.5.135 SOLF\_GSIF**

```
const int SOLF_GSIF = 5
```

solution format: GSI F1/F2

Definition at line 120 of file [rtklib.h](#).

#### 9.7.2.5.136 SOLF\_LLH

```
const int SOLF_LLH = 0
```

solution format: lat/lon/height

Definition at line 115 of file [rtklib.h](#).

#### 9.7.2.5.137 SOLF\_NMEA

```
const int SOLF_NMEA = 3
```

solution format: NMEA-183

Definition at line 118 of file [rtklib.h](#).

#### 9.7.2.5.138 SOLF\_STAT

```
const int SOLF_STAT = 4
```

solution format: solution status

Definition at line 119 of file [rtklib.h](#).

#### 9.7.2.5.139 SOLF\_XYZ

```
const int SOLF_XYZ = 1
```

solution format: x/y/z-ecef

Definition at line 116 of file [rtklib.h](#).

#### 9.7.2.5.140 SOLQ\_DGPS

```
const int SOLQ_DGPS = 4
```

solution status: DGPS/DGNSS

Definition at line 126 of file [rtklib.h](#).

#### 9.7.2.5.141 SOLQ\_DR

```
const int SOLQ_DR = 7
```

solution status: dead reckoning

Definition at line 129 of file [rtklib.h](#).



**9.7.2.5.142 SOLQ\_FIX**

```
const int SOLQ_FIX = 1
```

solution status: fix

Definition at line 123 of file [rtklib.h](#).

**9.7.2.5.143 SOLQ\_FLOAT**

```
const int SOLQ_FLOAT = 2
```

solution status: float

Definition at line 124 of file [rtklib.h](#).

**9.7.2.5.144 SOLQ\_NONE**

```
const int SOLQ_NONE = 0
```

solution status: no solution

Definition at line 122 of file [rtklib.h](#).

**9.7.2.5.145 SOLQ\_PPP**

```
const int SOLQ_PPP = 6
```

solution status: PPP

Definition at line 128 of file [rtklib.h](#).

**9.7.2.5.146 SOLQ\_SBAS**

```
const int SOLQ_SBAS = 3
```

solution status: SBAS

Definition at line 125 of file [rtklib.h](#).

**9.7.2.5.147 SOLQ\_SINGLE**

```
const int SOLQ_SINGLE = 5
```

solution status: single

Definition at line 127 of file [rtklib.h](#).

#### 9.7.2.5.148 STRFMT\_NMEA

```
const int STRFMT_NMEA = 19
```

Definition at line 1313 of file [rtklib.h](#).

#### 9.7.2.5.149 STRFMT\_RNXCLK

```
const int STRFMT_RNXCLK = 17
```

Definition at line 1311 of file [rtklib.h](#).

#### 9.7.2.5.150 STRFMT\_RTCM2

```
const int STRFMT_RTCM2 = 0
```

Definition at line 1308 of file [rtklib.h](#).

#### 9.7.2.5.151 STRFMT\_RTCM3

```
const int STRFMT_RTCM3 = 1
```

Definition at line 1309 of file [rtklib.h](#).

#### 9.7.2.5.152 STRFMT\_SBAS

```
const int STRFMT_SBAS = 18
```

Definition at line 1312 of file [rtklib.h](#).

#### 9.7.2.5.153 STRFMT\_SP3

```
const int STRFMT_SP3 = 16
```

Definition at line 1310 of file [rtklib.h](#).

#### 9.7.2.5.154 SYS\_ALL

```
const int SYS_ALL = 0xFF
```

navigation system: all

Definition at line 167 of file [rtklib.h](#).

**9.7.2.5.155 SYS\_BDS**

```
const int SYS_BDS = 0x20
```

navigation system: BeiDou

Definition at line 164 of file [rtklib.h](#).

**9.7.2.5.156 SYS\_GAL**

```
const int SYS_GAL = 0x08
```

navigation system: Galileo

Definition at line 162 of file [rtklib.h](#).

**9.7.2.5.157 SYS\_GLO**

```
const int SYS_GLO = 0x04
```

navigation system: GLONASS

Definition at line 161 of file [rtklib.h](#).

**9.7.2.5.158 SYS\_GPS**

```
const int SYS_GPS = 0x01
```

navigation system: GPS

Definition at line 159 of file [rtklib.h](#).

**9.7.2.5.159 SYS\_IRN**

```
const int SYS_IRN = 0x40
```

navigation system: IRNS

Definition at line 165 of file [rtklib.h](#).

**9.7.2.5.160 SYS\_LEO**

```
const int SYS_LEO = 0x80
```

navigation system: LEO

Definition at line 166 of file [rtklib.h](#).

#### 9.7.2.5.161 SYS\_NONE

```
const int SYS_NONE = 0x00
```

navigation system: none

Definition at line 158 of file [rtklib.h](#).

#### 9.7.2.5.162 SYS\_QZS

```
const int SYS_QZS = 0x10
```

navigation system: QZSS

Definition at line 163 of file [rtklib.h](#).

#### 9.7.2.5.163 SYS\_SBS

```
const int SYS_SBS = 0x02
```

navigation system: SBAS

Definition at line 160 of file [rtklib.h](#).

#### 9.7.2.5.164 TIMES\_GPST

```
const int TIMES_GPST = 0
```

time system: gps time

Definition at line 132 of file [rtklib.h](#).

#### 9.7.2.5.165 TIMES\_JST

```
const int TIMES_JST = 2
```

time system: jst

Definition at line 134 of file [rtklib.h](#).

#### 9.7.2.5.166 TIMES\_UTC

```
const int TIMES_UTC = 1
```

time system: utc

Definition at line 133 of file [rtklib.h](#).

**9.7.2.5.167 TIMETAGH\_LEN**

```
const int TIMETAGH_LEN = 64
```

time tag file header length

Definition at line 82 of file [rtklib.h](#).

**9.7.2.5.168 TINTACT**

```
const int TINTACT = 200
```

period for stream active (ms)

Definition at line 80 of file [rtklib.h](#).

**9.7.2.5.169 TROPOPT\_COR**

```
const int TROPOPT_COR = 5
```

troposphere option: ZTD correction

Definition at line 283 of file [rtklib.h](#).

**9.7.2.5.170 TROPOPT\_CORG**

```
const int TROPOPT_CORG = 6
```

troposphere option: ZTD+grad correction

Definition at line 284 of file [rtklib.h](#).

**9.7.2.5.171 TROPOPT\_EST**

```
const int TROPOPT_EST = 3
```

troposphere option: ZTD estimation

Definition at line 281 of file [rtklib.h](#).

**9.7.2.5.172 TROPOPT\_ESTG**

```
const int TROPOPT_ESTG = 4
```

troposphere option: ZTD+grad estimation

Definition at line 282 of file [rtklib.h](#).

#### 9.7.2.5.173 TROPOPT\_OFF

```
const int TROPOPT_OFF = 0
```

troposphere option: correction off

Definition at line 278 of file [rtklib.h](#).

#### 9.7.2.5.174 TROPOPT\_SAAS

```
const int TROPOPT_SAAS = 1
```

troposphere option: Saastamoinen model

Definition at line 279 of file [rtklib.h](#).

#### 9.7.2.5.175 TROPOPT\_SBAS

```
const int TROPOPT_SBAS = 2
```

troposphere option: SBAS model

Definition at line 280 of file [rtklib.h](#).

#### 9.7.2.5.176 TTOL\_MOVEB

```
const double TTOL_MOVEB = (1.0 + 2 * DTTOL)
```

Definition at line 57 of file [rtklib\\_rtkpos.h](#).

#### 9.7.2.5.177 VAR\_ACC

```
const double VAR_ACC = std::pow(10.0, 2.0)
```

Definition at line 47 of file [rtklib\\_rtkpos.h](#).

#### 9.7.2.5.178 VAR\_GRA

```
const double VAR_GRA = std::pow(0.001, 2.0)
```

Definition at line 49 of file [rtklib\\_rtkpos.h](#).

#### 9.7.2.5.179 VAR\_HOLDAMB

```
const double VAR_HOLDAMB = 0.001
```

Definition at line 55 of file [rtklib\\_rtkpos.h](#).

#### 9.7.2.5.180 VAR\_HWBIAS

```
const double VAR_HWBIAS = std::pow(1.0, 2.0)
```

Definition at line 48 of file [rtklib\\_rtkpos.h](#).

#### 9.7.2.5.181 VAR\_POS

```
const double VAR_POS = std::pow(30.0, 2.0)
```

Definition at line 45 of file [rtklib\\_rtkpos.h](#).

#### 9.7.2.5.182 VAR\_VEL

```
const double VAR_VEL = std::pow(10.0, 2.0)
```

Definition at line 46 of file [rtklib\\_rtkpos.h](#).

### 9.7.3 pvt\_adapters

Collaboration diagram for pvt\_adapters:



#### Classes

- class [Rtklib\\_Pvt](#)

*This class implements a [PvtInterface](#) for the RTKLIB PVT block.*

#### 9.7.3.1 Detailed Description

Wrap GNU Radio PVT solvers with a [PvtInterface](#)

### 9.7.4 pvt\_gr\_blocks

Collaboration diagram for pvt\_gr\_blocks:



#### Classes

- class [rtklib\\_pvt\\_gs](#)

*This class implements a block that computes the PVT solution using the RTKLIB integrated library.*

#### Typedefs

- using [rtklib\\_pvt\\_gs\\_sptr](#) = `gnss_shared_ptr<rtklib\_pvt\_gs>`

#### Functions

- `rtklib_pvt_gs_sptr rtklib\_make\_pvt\_gs (uint32_t nchannels, const Pvt\_Conf &conf_, const rtk\_t &rtk)`

#### 9.7.4.1 Detailed Description

GNU Radio blocks for the computation of PVT solutions.

#### 9.7.4.2 Typedef Documentation

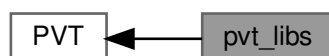
##### 9.7.4.2.1 rtklib\_pvt\_gs\_sptr

```
using rtklib_pvt_gs_sptr = gnss_shared_ptr<rtklib\_pvt\_gs>
```

Definition at line 73 of file [rtklib\\_pvt\\_gs.h](#).

### 9.7.5 pvt\_libs

Collaboration diagram for pvt\_libs:





## Classes

- struct [sdr\\_gnss\\_packet\\_t](#)
- struct [an\\_packet\\_t](#)
- class [An\\_Packet\\_Printer](#)

*class that prints PVT solutions in a serial device following a custom version of the Advanced Navigation Packet Protocol.*
- class [Geohash](#)

*Class for geohash encoding / decoding See <https://en.wikipedia.org/wiki/Geohash>.*
- class [GeoJSON\\_Printer](#)

*Prints PVT solutions in GeoJSON format file.*
- class [Gpx\\_Printer](#)

*Prints PVT information to GPX format file.*
- class [Has\\_Simple\\_Printer](#)

*Prints HAS messages content in a txt file. See HAS-SIS-ICD for a message description.*
- class [Kml\\_Printer](#)

*Prints PVT information to OGC KML format file (can be viewed with Google Earth)*
- class [Monitor\\_Ephemeris\\_Udp\\_Sink](#)
- class [Monitor\\_Pvt](#)

*This class contains parameters and outputs of the PVT block.*
- class [Monitor\\_Pvt\\_Udp\\_Sink](#)
- class [Nmea\\_Printer](#)

*This class provides a implementation of a subset of the NMEA-0183 standard for interfacing marine electronic devices as defined by the National Marine Electronics Association (NMEA).*
- class [Pvt\\_Conf](#)
- class [Pvt\\_Kf](#)

*Kalman Filter for Position and Velocity.*
- class [Pvt\\_Solution](#)

*Base class for a PVT solution.*
- class [Rinex\\_Printer](#)

*Class that handles the generation of Receiver INdependent EXchange format (RINEX) files.*
- class [Rtcm](#)

*This class implements the generation and reading of some Message Types defined in the RTCM 3.2 Standard, plus some utilities to handle messages.*
- class [Rtcm\\_Printer](#)

*This class provides a implementation of a subset of the RTCM Standard 10403.2 messages.*
- class [Rtklib\\_Solver](#)

*This class implements a PVT solution based on RTKLIB.*
- class [Serdes\\_Galileo\\_Eph](#)

*This class implements serialization and deserialization of [Galileo\\_Ephemeris](#) using Protocol Buffers.*
- class [Serdes\\_Gps\\_Eph](#)

*This class implements serialization and deserialization of [Gps\\_Ephemeris](#) objects using Protocol Buffers.*
- class [Serdes\\_Monitor\\_Pvt](#)

*This class implements serialization and deserialization of [Monitor\\_Pvt](#) objects using Protocol Buffers.*

## Typedefs

- using [b\\_io\\_context](#) = boost::asio::io\_service
- using [b\\_io\\_context](#) = boost::asio::io\_service
- using [b\\_io\\_context](#) = boost::asio::io\_service

## Functions

- `std::string asString` (long double *x*, `std::string::size_type` *precision*)
- `int64_t asInt` (const `std::string` &*s*)

### 9.7.5.1 Detailed Description

Library for the computation of PVT solutions.

### 9.7.5.2 Typedef Documentation

#### 9.7.5.2.1 `b_io_context` [1/3]

```
using b_io_context = boost::asio::io_service
```

Definition at line 39 of file [monitor\\_ephemeris\\_udp\\_sink.h](#).

#### 9.7.5.2.2 `b_io_context` [2/3]

```
using b_io_context = boost::asio::io_service
```

Definition at line 37 of file [monitor\\_pvt\\_udp\\_sink.h](#).

#### 9.7.5.2.3 `b_io_context` [3/3]

```
using b_io_context = boost::asio::io_service
```

Definition at line 64 of file [rtcm.h](#).

### 9.7.5.3 Function Documentation

#### 9.7.5.3.1 `asInt()`

```
int64_t asInt (
    const std::string & s) [inline]
```

Definition at line 1251 of file [rinex\\_printer.h](#).

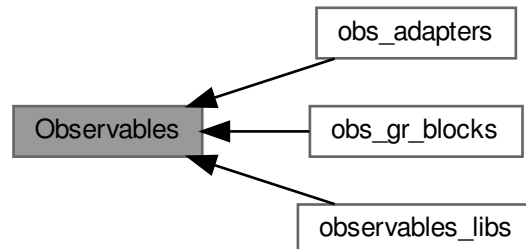
#### 9.7.5.3.2 `asString()`

```
std::string asString (
    long double x,
    std::string::size_type precision) [inline]
```

Definition at line 1227 of file [rinex\\_printer.h](#).

## 9.8 Observables

Collaboration diagram for Observables:



### Topics

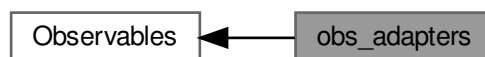
- [obs\\_adapters](#)
- [obs\\_gr\\_blocks](#)
- [observables\\_libs](#)

### 9.8.1 Detailed Description

Classes for the computation of GNSS observables

### 9.8.2 obs\_adapters

Collaboration diagram for obs\_adapters:



### Classes

- class [HybridObservables](#)  
*This class implements an [ObservablesInterface](#) for observables of all kind of GNSS signals.*

### 9.8.2.1 Detailed Description

Wrap GNU Radio observables blocks with an [ObservablesInterface](#)

### 9.8.3 obs\_gr\_blocks

Collaboration diagram for obs\_gr\_blocks:



#### Classes

- class [Gnss\\_circular\\_deque< T >](#)
- class [hybrid\\_observables\\_gs](#)

*This class implements a block that computes observables.*

#### Typedefs

- using [hybrid\\_observables\\_gs\\_sptr](#) = `gnss_shared_ptr<hybrid\_observables\_gs>`

#### Functions

- `hybrid_observables_gs_sptr hybrid_observables_gs_make (const Obs\_Conf &conf_)`

### 9.8.3.1 Detailed Description

GNU Radio blocks for the computation of GNSS observables

### 9.8.3.2 Typedef Documentation

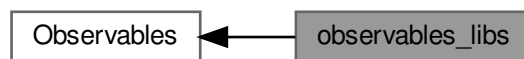
#### 9.8.3.2.1 hybrid\_observables\_gs\_sptr

```
using hybrid_observables_gs_sptr = gnss_shared_ptr<hybrid\_observables\_gs>
```

Definition at line 51 of file [hybrid\\_observables\\_gs.h](#).

### 9.8.4 observables\_libs

Collaboration diagram for observables\_libs:



#### Classes

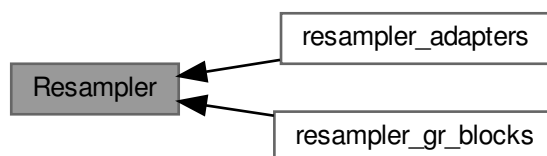
- class [Obs\\_Conf](#)

#### 9.8.4.1 Detailed Description

Utilities for GNSS observables configuration.

## 9.9 Resampler

Collaboration diagram for Resampler:



#### Topics

- [resampler\\_adapters](#)
- [resampler\\_gr\\_blocks](#)

#### 9.9.1 Detailed Description

Classes for input signal resampling

### 9.9.2 resampler\_adapters

Collaboration diagram for resampler\_adapters:



#### Classes

- class [DirectResamplerConditioner](#)  
*Interface of an adapter of a direct resampler conditioner block to a SignalConditionerInterface.*
- class [MmseResamplerConditioner](#)  
*Interface of a MMSE resampler block adapter to a SignalConditionerInterface.*

#### 9.9.2.1 Detailed Description

Classes that wrap GNU Radio resampler blocks with a [GNSSBlockInterface](#)

### 9.9.3 resampler\_gr\_blocks

Collaboration diagram for resampler\_gr\_blocks:



#### Classes

- class [direct\\_resampler\\_conditioner\\_cb](#)  
*This class implements a direct resampler conditioner for `std::complex<signed char>`*
- class [direct\\_resampler\\_conditioner\\_cc](#)  
*This class implements a direct resampler conditioner for complex data.*
- class [direct\\_resampler\\_conditioner\\_cs](#)  
*This class implements a direct resampler conditioner for `std::complex<short>`*

## Typedefs

- using [direct\\_resampler\\_conditioner\\_cb\\_sptr](#) = gnss\_shared\_ptr<[direct\\_resampler\\_conditioner\\_cb](#)>
- using [direct\\_resampler\\_conditioner\\_cc\\_sptr](#) = gnss\_shared\_ptr<[direct\\_resampler\\_conditioner\\_cc](#)>
- using [direct\\_resampler\\_conditioner\\_cs\\_sptr](#) = gnss\_shared\_ptr<[direct\\_resampler\\_conditioner\\_cs](#)>

## Functions

- [direct\\_resampler\\_conditioner\\_cb\\_sptr](#) **direct\_resampler\_make\_conditioner\_cb** (double sample\_freq\_in, double sample\_freq\_out)
- [direct\\_resampler\\_conditioner\\_cc\\_sptr](#) **direct\_resampler\_make\_conditioner\_cc** (double sample\_freq\_in, double sample\_freq\_out)
- [direct\\_resampler\\_conditioner\\_cs\\_sptr](#) **direct\_resampler\_make\_conditioner\_cs** (double sample\_freq\_in, double sample\_freq\_out)

### 9.9.3.1 Detailed Description

GNU Radio blocks for input signal resampling

### 9.9.3.2 Typedef Documentation

#### 9.9.3.2.1 [direct\\_resampler\\_conditioner\\_cb\\_sptr](#)

```
using direct_resampler_conditioner_cb_sptr = gnss_shared_ptr<direct\_resampler\_conditioner\_cb>
```

Definition at line 35 of file [direct\\_resampler\\_conditioner\\_cb.h](#).

#### 9.9.3.2.2 [direct\\_resampler\\_conditioner\\_cc\\_sptr](#)

```
using direct_resampler_conditioner_cc_sptr = gnss_shared_ptr<direct\_resampler\_conditioner\_cc>
```

Definition at line 40 of file [direct\\_resampler\\_conditioner\\_cc.h](#).

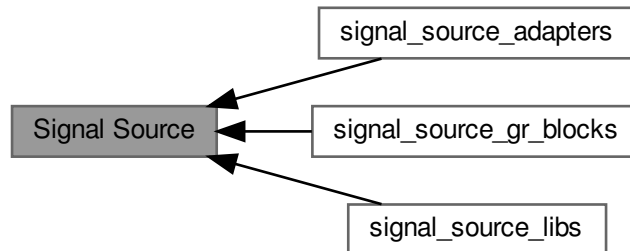
#### 9.9.3.2.3 [direct\\_resampler\\_conditioner\\_cs\\_sptr](#)

```
using direct_resampler_conditioner_cs_sptr = gnss_shared_ptr<direct\_resampler\_conditioner\_cs>
```

Definition at line 34 of file [direct\\_resampler\\_conditioner\\_cs.h](#).

## 9.10 Signal Source

Collaboration diagram for Signal Source:



### Topics

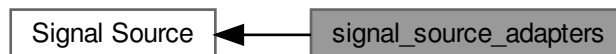
- [signal\\_source\\_adapters](#)
- [signal\\_source\\_gr\\_blocks](#)
- [signal\\_source\\_libs](#)

### 9.10.1 Detailed Description

Classes for Signal Source management.

### 9.10.2 signal\_source\_adapters

Collaboration diagram for signal\_source\_adapters:





## Classes

- class [Ad936xCustomSignalSource](#)  
*This class instantiates the Ad936xCustom gnuradio signal source. It has support also for a customized Ad936x↔ Custom firmware and signal source to support PPS samplestamp reading.*
- class [Adrv9361z7035SignalSourceFPGA](#)
- class [CustomUDPSignalSource](#)  
*This class reads from UDP packets, which streams interleaved I/Q samples over a network.*
- class [DMASignalSourceFPGA](#)
- class [FifoSignalSource](#)  
*Class that reads a sample stream from a Unix FIFO.*
- class [FileSignalSource](#)  
*Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).*
- class [FileSourceBase](#)  
*Base class to file-oriented [SignalSourceBase](#) GNSS blocks.*
- class [FileTimestampSignalSource](#)  
*Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).*
- class [FlexibandSignalSource](#)  
*This class configures and reads samples from Teleorbit Flexiband front-end. This software requires a Flexiband GNU Radio driver installed (not included with GNSS-SDR).*
- class [Fmcomms2SignalSource](#)
- class [Fmcomms5SignalSourceFPGA](#)
- class [FourBitCpxFileSignalSource](#)  
*Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).*
- class [GenSignalSource](#)  
*This class wraps blocks that generates synthesized GNSS signal and filters the signal.*
- class [IONGSMSSignalSource](#)  
*Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).*
- class [LabsatSignalSource](#)  
*This class reads samples stored in LabSat version 2, 3, and 3 Wideband format.*
- class [LimesdrSignalSource](#)  
*This class instantiates the LimeSDR gnuradio signal source. It has support also for a customized LimeSDR firmware and signal source to support PPS samplestamp reading.*
- class [MAX2771EVKITSignalSourceFPGA](#)
- class [MultichannelFileSignalSource](#)  
*Class that reads signals samples from files at different frequency bands and adapts it to a [SignalSourceInterface](#).*
- class [NsrFileSignalSource](#)  
*Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).*
- class [OsmosdrSignalSource](#)  
*This class reads samples OsmoSDR-compatible front-ends, such as HackRF or Realtek's RTL2832U-based USB dongle DVB-T receivers (see <https://osmocom.org/projects/rtl-sdr/wiki>)*
- class [PlutosdrSignalSource](#)
- class [RawArraySignalSource](#)  
*This class reads samples from an antenna array RF front-end signal sampler.*
- class [RtlTcpSignalSource](#)  
*This class reads from rtl\_tcp, which streams interleaved I/Q samples over TCP. (see <https://osmocom.org/projects/rtl-sdr/wiki>)*
- class [SpirFileSignalSource](#)  
*Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).*
- class [SpirGSS6450FileSignalSource](#)  
*Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).*
- class [TwoBitCpxFileSignalSource](#)

*Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).*

- class [TwoBitPackedFileSignalSource](#)

*Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).*

- class [UhdSignalSource](#)

*This class reads samples from a UHD device (see <http://code.ettus.com/redmine/ettus/projects/uhd/wiki>)*

- class [ZmqSignalSource](#)

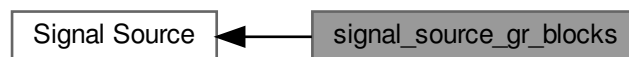
*This class supports the following properties:*

### 9.10.2.1 Detailed Description

Classes that wrap GNU Radio signal sources with a [GNSSBlockInterface](#)

### 9.10.3 signal\_source\_gr\_blocks

Collaboration diagram for signal\_source\_gr\_blocks:



### Classes

- class [ad936x\\_iio\\_source](#)

*This class implements conversion between Labsat 2, 3 and 3 Wideband formats to gr\_complex.*

- class [FifoReader](#)
- class [Gr\\_Complex\\_Ip\\_Packet\\_Source](#)
- class [IONGSMSFileSource](#)
- class [labsat23\\_source](#)

*This class implements conversion between Labsat 2, 3 and 3 Wideband formats to gr\_complex.*

- class [rtl\\_tcp\\_signal\\_source\\_c](#)

*This class reads interleaved I/Q samples from an rtl\_tcp server and outputs complex types.*

- class [unpack\\_2bit\\_samples](#)

*This class takes 2 bit samples that have been packed into bytes or shorts as input and generates a byte for each sample. It generates eight times as much data as is input (every two bits become 16 bits)*

- class [unpack\\_byte\\_2bit\\_cpx\\_samples](#)

*This class implements conversion between byte packet samples to 2bit\_cpx samples 1 byte = 2 x complex 2bit I, + 2bit Q samples.*

- class [unpack\\_byte\\_2bit\\_samples](#)

*This class implements conversion between byte packet samples to 2bit samples 1 byte = 4 2bit samples.*

- class [unpack\\_byte\\_4bit\\_samples](#)

*This class implements conversion between byte packet samples to 4bit\_cpx samples 1 byte = 1 x complex 4bit I, + 4bit Q samples.*

- class [unpack\\_intspir\\_1bit\\_samples](#)

*This class implements conversion between byte packet samples to 2bit samples 1 byte = 4 2bit samples.*

- class [unpack\\_short\\_byte\\_samples](#)

*This class implements conversion between short packet samples to byte samples 1 short = 2 byte samples.*

- class [unpack\\_spir\\_gss6450\\_samples](#)

## Typedefs

- using `ad936x_iio_source_sptr` = `gnss_shared_ptr<ad936x_iio_source>`
- using `labsat23_source_sptr` = `gnss_shared_ptr<labsat23_source>`
- using `rtl_tcp_signal_source_c_sptr` = `gnss_shared_ptr<rtl_tcp_signal_source_c>`
- using `b_io_context` = `boost::asio::io_service`
- using `unpack_2bit_samples_sptr` = `gnss_shared_ptr<unpack_2bit_samples>`
- using `unpack_byte_2bit_cpx_samples_sptr` = `gnss_shared_ptr<unpack_byte_2bit_cpx_samples>`
- using `unpack_byte_2bit_samples_sptr` = `gnss_shared_ptr<unpack_byte_2bit_samples>`
- using `unpack_byte_4bit_samples_sptr` = `gnss_shared_ptr<unpack_byte_4bit_samples>`
- using `unpack_intspir_1bit_samples_sptr` = `gnss_shared_ptr<unpack_intspir_1bit_samples>`
- using `unpack_short_byte_samples_sptr` = `gnss_shared_ptr<unpack_short_byte_samples>`
- using `unpack_spir_gss6450_samples_sptr` = `gnss_shared_ptr<unpack_spir_gss6450_samples>`

## Functions

- `ad936x_iio_source_sptr` **ad936x\_iio\_make\_source\_sptr** (const std::string &pluto\_uri\_, const std::string &board\_type\_, int64\_t bandwidth\_, int64\_t sample\_rate\_, int64\_t freq\_, const std::string &rf\_port\_select\_←, const std::string &rf\_filter, const std::string &gain\_mode\_rx0\_, const std::string &gain\_mode\_rx1\_, double rf\_gain\_rx0\_, double rf\_gain\_rx1\_, bool enable\_ch0, bool enable\_ch1, int64\_t freq\_2ch, bool ppsmode\_←, bool customsamplesize\_, const std::string &fe\_ip\_, int fe\_ctlport\_, int ssize\_, int bshift\_, bool spattern\_, double lo\_attenuation\_db\_, bool high\_side\_lo\_, int tx\_lo\_channel\_)
- `labsat23_source_sptr` **labsat23\_make\_source\_sptr** (const char \*signal\_file\_basename, const std::vector<int> &channel\_selector, [Concurrent\\_Queue](#)<pmt::pmt\_t> \*queue, bool digital\_io\_enabled)
- `rtl_tcp_signal_source_c_sptr` **rtl\_tcp\_make\_signal\_source\_c** (const std::string &address, int16\_t port, bool flip\_iq=false)
- `unpack_2bit_samples_sptr` **make\_unpack\_2bit\_samples** (bool big\_endian\_bytes, size\_t item\_size, bool big\_endian\_items, bool reverse\_interleaving=false)
- `unpack_byte_2bit_cpx_samples_sptr` **make\_unpack\_byte\_2bit\_cpx\_samples** ()
- `unpack_byte_2bit_samples_sptr` **make\_unpack\_byte\_2bit\_samples** ()
- `unpack_byte_4bit_samples_sptr` **make\_unpack\_byte\_4bit\_samples** ()
- `unpack_intspir_1bit_samples_sptr` **make\_unpack\_intspir\_1bit\_samples** ()
- `unpack_short_byte_samples_sptr` **make\_unpack\_short\_byte\_samples** ()
- `unpack_spir_gss6450_samples_sptr` **make\_unpack\_spir\_gss6450\_samples** (int adc\_nbit\_)

### 9.10.3.1 Detailed Description

GNU Radio blocks for signal sources.

### 9.10.3.2 Typedef Documentation

#### 9.10.3.2.1 ad936x\_iio\_source\_sptr

```
using ad936x_iio_source_sptr = gnss_shared_ptr<ad936x_iio_source>
```

Definition at line 45 of file [ad936x\\_iio\\_source.h](#).

#### 9.10.3.2.2 b\_io\_context

```
using b_io_context = boost::asio::io_service
```

Definition at line 53 of file [rtl\\_tcp\\_signal\\_source\\_c.h](#).

#### 9.10.3.2.3 labsat23\_source\_sptr

```
using labsat23_source_sptr = gnss_shared_ptr<labsat23_source>
```

Definition at line 40 of file [labsat23\\_source.h](#).

#### 9.10.3.2.4 rtl\_tcp\_signal\_source\_c\_sptr

```
using rtl_tcp_signal_source_c_sptr = gnss_shared_ptr<rtl_tcp_signal_source_c>
```

Definition at line 48 of file [rtl\\_tcp\\_signal\\_source\\_c.h](#).

#### 9.10.3.2.5 unpack\_2bit\_samples\_sptr

```
using unpack_2bit_samples_sptr = gnss_shared_ptr<unpack_2bit_samples>
```

Definition at line 71 of file [unpack\\_2bit\\_samples.h](#).

#### 9.10.3.2.6 unpack\_byte\_2bit\_cpx\_samples\_sptr

```
using unpack_byte_2bit_cpx_samples_sptr = gnss_shared_ptr<unpack_byte_2bit_cpx_samples>
```

Definition at line 36 of file [unpack\\_byte\\_2bit\\_cpx\\_samples.h](#).

#### 9.10.3.2.7 unpack\_byte\_2bit\_samples\_sptr

```
using unpack_byte_2bit_samples_sptr = gnss_shared_ptr<unpack_byte_2bit_samples>
```

Definition at line 32 of file [unpack\\_byte\\_2bit\\_samples.h](#).

#### 9.10.3.2.8 unpack\_byte\_4bit\_samples\_sptr

```
using unpack_byte_4bit_samples_sptr = gnss_shared_ptr<unpack_byte_4bit_samples>
```

Definition at line 34 of file [unpack\\_byte\\_4bit\\_samples.h](#).

#### 9.10.3.2.9 unpack\_intspir\_1bit\_samples\_sptr

```
using unpack_intspir_1bit_samples_sptr = gnss_shared_ptr<unpack_intspir_1bit_samples>
```

Definition at line 33 of file [unpack\\_intspir\\_1bit\\_samples.h](#).

#### 9.10.3.2.10 unpack\_short\_byte\_samples\_sptr

```
using unpack_short_byte_samples_sptr = gnss_shared_ptr<unpack_short_byte_samples>
```

Definition at line 33 of file [unpack\\_short\\_byte\\_samples.h](#).

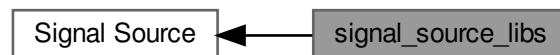
#### 9.10.3.2.11 unpack\_spir\_gss6450\_samples\_sptr

```
using unpack_spir_gss6450_samples_sptr = gnss_shared_ptr<unpack_spir_gss6450_samples>
```

Definition at line 32 of file [unpack\\_spir\\_gss6450\\_samples.h](#).

### 9.10.4 signal\_source\_libs

Collaboration diagram for signal\_source\_libs:



#### Classes

- struct [stream\\_cfg](#)
- class [ad936x\\_iio\\_custom](#)
- class [ad936x\\_iio\\_samples](#)
- class [Fpga\\_buffer\\_monitor](#)
  - Class that checks the receiver buffer overflow flags and monitors the status of the receiver buffers.*
- class [Fpga\\_dynamic\\_bit\\_selection](#)
  - Class that controls the switch in the FPGA, which connects the FPGA acquisition and multicorrelator modules to either the DMA or the Analog Front-End.*
- class [Fpga\\_Switch](#)
  - Class that controls the switch in the FPGA, which connects the FPGA acquisition and multicorrelator modules to either the DMA or the Analog Front-End.*
- class [Gnss\\_Sdr\\_Timestamp](#)
- class [Gnss\\_Sdr\\_Valve](#)
  - Implementation of a GNU Radio block that sends a STOP message to the control queue right after a specific number of samples have passed through it.*
- struct [LONGSMSChunkUnpackingCtx< WT >](#)
- class [Rtl\\_Tcp\\_Dongle\\_Info](#)
  - This class represents the dongle information which is sent by rtl\_tcp.*

#### Macros

- `#define FIR\_BUF\_SIZE 8192`

#### Enumerations

- enum `iodev { RX , TX }`
- enum `RTL\_TCP\_COMMAND {`
  - `RTL\_TCP\_SET\_FREQUENCY = 1 , RTL\_TCP\_SET\_SAMPLE\_RATE = 2 , RTL\_TCP\_SET\_GAIN\_MODE`
  - `= 3 , RTL\_TCP\_SET\_GAIN = 4 ,`
  - `RTL\_TCP\_SET\_IF\_GAIN = 6 , RTL\_TCP\_SET\_AGC\_MODE = 8 }`
  - Command IDs for configuration rtl\_tcp.*

## Functions

- void **errchk** (int v, const char \*what)
- void **wr\_ch\_lll** (struct iio\_channel \*chn, const char \*what, int64\_t val)
- void **wr\_ch\_str** (struct iio\_channel \*chn, const char \*what, const char \*str)
- struct iio\_device \* **get\_ad9361\_phy** (struct iio\_context \*ctx)
- bool **get\_ad9361\_stream\_dev** (struct iio\_context \*ctx, enum iodev d, struct iio\_device \*\*dev)
- bool **get\_ad9361\_stream\_ch** (struct iio\_context \*ctx, enum iodev d, struct iio\_device \*dev, int chid, struct iio\_channel \*\*chn)
- bool **get\_phy\_chan** (struct iio\_context \*ctx, enum iodev d, int chid, struct iio\_channel \*\*chn)
- bool **get\_lo\_chan** (struct iio\_context \*ctx, enum iodev d, struct iio\_channel \*\*chn)
- bool **cfg\_ad9361\_streaming\_ch** (struct iio\_context \*ctx, struct [stream\\_cfg](#) \*cfg, enum iodev type, int chid)
- bool **config\_ad9361\_rx\_local** (uint64\_t bandwidth\_, uint64\_t sample\_rate\_, uint64\_t freq0\_, uint64\_t freq1\_↵\_, const std::string &rf\_port\_select\_, bool rx1\_enable\_, bool rx2\_enable\_, const std::string &gain\_mode\_↵\_rx1\_, const std::string &gain\_mode\_rx2\_, double rf\_gain\_rx1\_, double rf\_gain\_rx2\_, bool quadrature\_, bool rfdc\_, bool bbdc\_, std::string filter\_source\_, std::string filter\_filename\_, float Fpass\_, float Fstop\_)
- bool **config\_ad9361\_rx\_remote** (const std::string &remote\_host, uint64\_t bandwidth\_, uint64\_t sample\_↵\_rate\_, uint64\_t freq\_, const std::string &rf\_port\_select\_, bool rx1\_enable\_, bool rx2\_enable\_, const std\_↵\_::string &gain\_mode\_rx1\_, const std::string &gain\_mode\_rx2\_, double rf\_gain\_rx1\_, double rf\_gain\_rx2\_, bool quadrature\_, bool rfdc\_, bool bbdc\_, std::string filter\_source\_, std::string filter\_filename\_, float Fpass\_, float Fstop\_)
- bool **config\_ad9361\_lo\_local** (uint64\_t bandwidth\_, uint64\_t sample\_rate\_, uint64\_t freq\_rf\_tx\_hz\_, double tx\_attenuation\_db\_, int64\_t freq\_dds\_tx\_hz\_, double scale\_dds\_dbfs\_, double phase\_dds\_deg\_)
- bool **config\_ad9361\_lo\_remote** (const std::string &remote\_host, uint64\_t bandwidth\_, uint64\_t sample\_↵\_rate\_, uint64\_t freq\_rf\_tx\_hz\_, double tx\_attenuation\_db\_, int64\_t freq\_dds\_tx\_hz\_, double scale\_dds\_↵\_dbfs\_, double phase\_dds\_deg\_)
- bool **ad9361\_disable\_lo\_remote** (const std::string &remote\_host)
- bool **ad9361\_disable\_lo\_local** ()
- bool **load\_fir\_filter** (std::string &filter, struct iio\_device \*phy)
- bool **disable\_ad9361\_rx\_local** ()
- bool **disable\_ad9361\_rx\_remote** (const std::string &remote\_host)
- gnss\_shared\_ptr< [Gnss\\_Sdr\\_Timestamp](#) > **gnss\_sdr\_make\_timestamp** (size\_t sizeof\_stream\_item, std::string timestamp\_file, double clock\_offset\_ms, int items\_to\_samples)
- gnss\_shared\_ptr< [Gnss\\_Sdr\\_Valve](#) > **gnss\_sdr\_make\_valve** (size\_t sizeof\_stream\_item, uint64\_t nitems, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)
- gnss\_shared\_ptr< [Gnss\\_Sdr\\_Valve](#) > **gnss\_sdr\_make\_valve** (size\_t sizeof\_stream\_item, uint64\_t nitems, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue, bool stop\_flowgraph)
- boost::system::error\_code **rtl\_tcp\_command** (RTL\_TCP\_COMMAND id, unsigned param, boost::asio::ip\_↵\_::tcp::socket &socket)

*Send a command to rtl\_tcp over the given socket.*

### 9.10.4.1 Detailed Description

Library with utilities for signal sources.

### 9.10.4.2 Macro Definition Documentation

#### 9.10.4.2.1 FIR\_BUF\_SIZE

```
#define FIR_BUF_SIZE 8192
```

Definition at line 33 of file [ad9361\\_manager.h](#).

### 9.10.4.3 Enumeration Type Documentation

#### 9.10.4.3.1 iodev

```
enum iodev
```

Definition at line 43 of file [ad9361\\_manager.h](#).

#### 9.10.4.3.2 RTL\_TCP\_COMMAND

```
enum RTL_TCP_COMMAND
```

Command IDs for configuration rtl\_tcp.

Definition at line 32 of file [rtl\\_tcp\\_commands.h](#).

### 9.10.4.4 Function Documentation

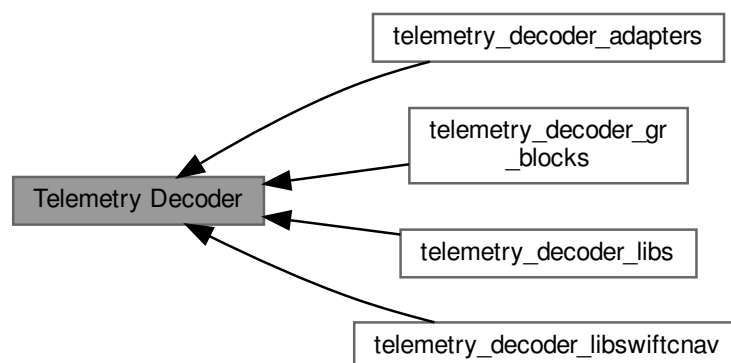
#### 9.10.4.4.1 rtl\_tcp\_command()

```
boost::system::error_code rtl_tcp_command (  
    RTL_TCP_COMMAND id,  
    unsigned param,  
    boost::asio::ip::tcp::socket & socket)
```

Send a command to rtl\_tcp over the given socket.

## 9.11 Telemetry Decoder

Collaboration diagram for Telemetry Decoder:



## Topics

- [telemetry\\_decoder\\_adapters](#)
- [telemetry\\_decoder\\_gr\\_blocks](#)
- [telemetry\\_decoder\\_libswiftcnv](#)
- [telemetry\\_decoder\\_libs](#)

### 9.11.1 Detailed Description

Classes for the decoding of GNSS Navigation messages.

### 9.11.2 `telemetry_decoder_adapters`

Collaboration diagram for `telemetry_decoder_adapters`:



## Classes

- class [BeidouB1iTelemetryDecoder](#)  
*This class implements a NAV data decoder for BEIDOU B1I.*
- class [BeidouB3iTelemetryDecoder](#)  
*This class implements a NAV data decoder for BEIDOU B1I.*
- class [GalileoE1BTelemetryDecoder](#)  
*This class implements a NAV data decoder for Galileo INAV frames in E1B radio link.*
- class [GalileoE5aTelemetryDecoder](#)  
*This class implements a NAV data decoder for Galileo INAV frames in E1B radio link.*
- class [GalileoE5bTelemetryDecoder](#)  
*This class implements a NAV data decoder for Galileo INAV frames in E5b radio link.*
- class [GalileoE6TelemetryDecoder](#)  
*This class implements a NAV data decoder for Galileo CNAV frames in E6 radio link.*
- class [GlonassL1CaTelemetryDecoder](#)  
*This class implements a NAV data decoder for GLONASS L1 C/A.*
- class [GlonassL2CaTelemetryDecoder](#)  
*This class implements a NAV data decoder for GLONASS L2 C/A.*
- class [GpsL1CaTelemetryDecoder](#)  
*This class implements a NAV data decoder for GPS L1 C/A.*
- class [GpsL2CTelemetryDecoder](#)  
*This class implements a NAV data decoder for GPS L2 M.*
- class [GpsL5TelemetryDecoder](#)  
*This class implements a NAV data decoder for GPS L5.*
- class [SbasL1TelemetryDecoder](#)  
*This class implements a NAV data decoder for SBAS frames in L1 radio link.*



### 9.11.2.1 Detailed Description

Wrap GNU Radio blocks for the decoding of GNSS Navigation messages with a [TelemetryDecoderInterface](#)

### 9.11.3 telemetry\_decoder\_gr\_blocks

Collaboration diagram for telemetry\_decoder\_gr\_blocks:



#### Classes

- class [beidou\\_b1i\\_telemetry\\_decoder\\_gs](#)  
*This class implements a block that decodes the BeiDou DNAV data.*
- class [beidou\\_b3i\\_telemetry\\_decoder\\_gs](#)  
*This class implements a block that decodes the BeiDou DNAV data.*
- class [galileo\\_telemetry\\_decoder\\_gs](#)  
*This class implements a block that decodes the INAV and FNAV data defined in Galileo ICD.*
- class [glonass\\_l1\\_ca\\_telemetry\\_decoder\\_gs](#)  
*This class implements a block that decodes the GNAV data defined in GLONASS ICD v5.1.*
- class [glonass\\_l2\\_ca\\_telemetry\\_decoder\\_gs](#)  
*This class implements a block that decodes the GNAV data defined in GLONASS ICD v5.1.*
- class [gps\\_l1\\_ca\\_telemetry\\_decoder\\_gs](#)  
*This class implements a block that decodes the NAV data defined in IS-GPS-200M.*
- class [gps\\_l2c\\_telemetry\\_decoder\\_gs](#)  
*This class implements a block that decodes CNAV data defined in IS-GPS-200M.*
- class [gps\\_l5\\_telemetry\\_decoder\\_gs](#)  
*This class implements a GPS L5 Telemetry decoder.*
- class [sbas\\_l1\\_telemetry\\_decoder\\_gs](#)  
*This class implements a block that decodes the SBAS integrity and corrections data defined in RTCA MOPS DO-229.*

#### Typedefs

- using [beidou\\_b1i\\_telemetry\\_decoder\\_gs\\_sptr](#) = gnss\_shared\_ptr<[beidou\\_b1i\\_telemetry\\_decoder\\_gs](#)>
- using [beidou\\_b3i\\_telemetry\\_decoder\\_gs\\_sptr](#)
- using [galileo\\_telemetry\\_decoder\\_gs\\_sptr](#) = gnss\_shared\_ptr<[galileo\\_telemetry\\_decoder\\_gs](#)>
- using [glonass\\_l1\\_ca\\_telemetry\\_decoder\\_gs\\_sptr](#) = gnss\_shared\_ptr<[glonass\\_l1\\_ca\\_telemetry\\_decoder\\_gs](#)>
- using [glonass\\_l2\\_ca\\_telemetry\\_decoder\\_gs\\_sptr](#) = gnss\_shared\_ptr<[glonass\\_l2\\_ca\\_telemetry\\_decoder\\_gs](#)>
- using [gps\\_l1\\_ca\\_telemetry\\_decoder\\_gs\\_sptr](#) = gnss\_shared\_ptr<[gps\\_l1\\_ca\\_telemetry\\_decoder\\_gs](#)>
- using [gps\\_l2c\\_telemetry\\_decoder\\_gs\\_sptr](#) = gnss\_shared\_ptr<[gps\\_l2c\\_telemetry\\_decoder\\_gs](#)>
- using [gps\\_l5\\_telemetry\\_decoder\\_gs\\_sptr](#) = gnss\_shared\_ptr<[gps\\_l5\\_telemetry\\_decoder\\_gs](#)>
- using [sbas\\_l1\\_telemetry\\_decoder\\_gs\\_sptr](#) = gnss\_shared\_ptr<[sbas\\_l1\\_telemetry\\_decoder\\_gs](#)>

## Functions

- `beidou_b1i_telemetry_decoder_gs_sptr` **`beidou_b1i_make_telemetry_decoder_gs`** (const [Gnss\\_Satellite](#) &satellite, const [Tlm\\_Conf](#) &conf)
- `beidou_b3i_telemetry_decoder_gs_sptr` **`beidou_b3i_make_telemetry_decoder_gs`** (const [Gnss\\_Satellite](#) &satellite, const [Tlm\\_Conf](#) &conf)
- `galileo_telemetry_decoder_gs_sptr` **`galileo_make_telemetry_decoder_gs`** (const [Gnss\\_Satellite](#) &satellite, const [Tlm\\_Conf](#) &conf, int frame\_type)
- `glonass_l1_ca_telemetry_decoder_gs_sptr` **`glonass_l1_ca_make_telemetry_decoder_gs`** (const [Gnss\\_Satellite](#) &satellite, const [Tlm\\_Conf](#) &conf)
- `glonass_l2_ca_telemetry_decoder_gs_sptr` **`glonass_l2_ca_make_telemetry_decoder_gs`** (const [Gnss\\_Satellite](#) &satellite, const [Tlm\\_Conf](#) &conf)
- `gps_l1_ca_telemetry_decoder_gs_sptr` **`gps_l1_ca_make_telemetry_decoder_gs`** (const [Gnss\\_Satellite](#) &satellite, const [Tlm\\_Conf](#) &conf)
- `gps_l2c_telemetry_decoder_gs_sptr` **`gps_l2c_make_telemetry_decoder_gs`** (const [Gnss\\_Satellite](#) &satellite, const [Tlm\\_Conf](#) &conf)
- `gps_l5_telemetry_decoder_gs_sptr` **`gps_l5_make_telemetry_decoder_gs`** (const [Gnss\\_Satellite](#) &satellite, const [Tlm\\_Conf](#) &conf)
- `sbas_l1_telemetry_decoder_gs_sptr` **`sbas_l1_make_telemetry_decoder_gs`** (const [Gnss\\_Satellite](#) &satellite, bool dump)

### 9.11.3.1 Detailed Description

GNU Radio blocks for the demodulation of GNSS navigation messages.

### 9.11.3.2 Typedef Documentation

#### 9.11.3.2.1 `beidou_b1i_telemetry_decoder_gs_sptr`

```
using beidou_b1i_telemetry_decoder_gs_sptr = gnss_shared_ptr<beidou_b1i_telemetry_decoder_gs>
```

Definition at line 46 of file [beidou\\_b1i\\_telemetry\\_decoder\\_gs.h](#).

#### 9.11.3.2.2 `beidou_b3i_telemetry_decoder_gs_sptr`

```
using beidou_b3i_telemetry_decoder_gs_sptr
```

**Initial value:**

```
gnss_shared_ptr<beidou_b3i_telemetry_decoder_gs>
```

Definition at line 44 of file [beidou\\_b3i\\_telemetry\\_decoder\\_gs.h](#).

#### 9.11.3.2.3 `galileo_telemetry_decoder_gs_sptr`

```
using galileo_telemetry_decoder_gs_sptr = gnss_shared_ptr<galileo_telemetry_decoder_gs>
```

Definition at line 51 of file [galileo\\_telemetry\\_decoder\\_gs.h](#).

#### 9.11.3.2.4 glonass\_l1\_ca\_telemetry\_decoder\_gs\_sptr

```
using glonass_l1_ca_telemetry_decoder_gs_sptr = gnss_shared_ptr<glonass_l1_ca_telemetry_decoder_gs>
```

Definition at line 47 of file [glonass\\_l1\\_ca\\_telemetry\\_decoder\\_gs.h](#).

#### 9.11.3.2.5 glonass\_l2\_ca\_telemetry\_decoder\_gs\_sptr

```
using glonass_l2_ca_telemetry_decoder_gs_sptr = gnss_shared_ptr<glonass_l2_ca_telemetry_decoder_gs>
```

Definition at line 46 of file [glonass\\_l2\\_ca\\_telemetry\\_decoder\\_gs.h](#).

#### 9.11.3.2.6 gps\_l1\_ca\_telemetry\_decoder\_gs\_sptr

```
using gps_l1_ca_telemetry_decoder_gs_sptr = gnss_shared_ptr<gps_l1_ca_telemetry_decoder_gs>
```

Definition at line 46 of file [gps\\_l1\\_ca\\_telemetry\\_decoder\\_gs.h](#).

#### 9.11.3.2.7 gps\_l2c\_telemetry\_decoder\_gs\_sptr

```
using gps_l2c_telemetry_decoder_gs_sptr = gnss_shared_ptr<gps_l2c_telemetry_decoder_gs>
```

Definition at line 47 of file [gps\\_l2c\\_telemetry\\_decoder\\_gs.h](#).

#### 9.11.3.2.8 gps\_l5\_telemetry\_decoder\_gs\_sptr

```
using gps_l5_telemetry_decoder_gs_sptr = gnss_shared_ptr<gps_l5_telemetry_decoder_gs>
```

Definition at line 48 of file [gps\\_l5\\_telemetry\\_decoder\\_gs.h](#).

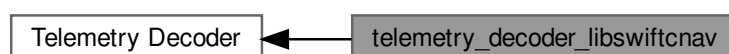
#### 9.11.3.2.9 sbas\_l1\_telemetry\_decoder\_gs\_sptr

```
using sbas_l1_telemetry_decoder_gs_sptr = gnss_shared_ptr<sbas_l1_telemetry_decoder_gs>
```

Definition at line 44 of file [sbas\\_l1\\_telemetry\\_decoder\\_gs.h](#).

### 9.11.4 telemetry\_decoder\_libswiftcnv

Collaboration diagram for telemetry\_decoder\_libswiftcnv:



## Classes

- struct [cnav\\_msg\\_t](#)
- struct [cnav\\_v27\\_part\\_t](#)
- struct [cnav\\_msg\\_decoder\\_t](#)

## Macros

- `#define` [GPS\\_L2\\_V27\\_HISTORY\\_LENGTH\\_BITS](#) 64
- `#define` [GPS\\_L2C\\_V27\\_INIT\\_BITS](#) (32)
- `#define` [GPS\\_L2C\\_V27\\_DECODE\\_BITS](#) (32)
- `#define` [GPS\\_L2C\\_V27\\_DELAY\\_BITS](#) (32)
- `#define` [ABS](#)(x)
- `#define` [MIN](#)(x, y)
- `#define` [MAX](#)(x, y)
- `#define` [CLAMP\\_DIFF](#)(a, b)

## Functions

- `uint8_t` [parity](#) (`uint32_t` x)
- `uint32_t` [getbitu](#) (`const uint8_t *buff`, `uint32_t` pos, `uint8_t` len)
- `int32_t` [getbits](#) (`const uint8_t *buff`, `uint32_t` pos, `uint8_t` len)
- `void` [setbitu](#) (`uint8_t *buff`, `uint32_t` pos, `uint32_t` len, `uint32_t` data)
- `void` [setbits](#) (`uint8_t *buff`, `uint32_t` pos, `uint32_t` len, `int32_t` data)
- `void` [bitcopy](#) (`void *dst`, `uint32_t` dst\_index, `const void *src`, `uint32_t` src\_index, `uint32_t` count)
- `void` [bitshl](#) (`void *buf`, `uint32_t` size, `uint32_t` shift)
- `uint8_t` [count\\_bits\\_u64](#) (`uint64_t` v, `uint8_t` bv)
- `uint8_t` [count\\_bits\\_u32](#) (`uint32_t` v, `uint8_t` bv)
- `uint8_t` [count\\_bits\\_u16](#) (`uint16_t` v, `uint8_t` bv)
- `uint8_t` [count\\_bits\\_u8](#) (`uint8_t` v, `uint8_t` bv)
- `const` [v27\\_poly\\_t](#) \* [cnav\\_msg\\_decoder\\_get\\_poly](#) (`void`)
- `void` [cnav\\_msg\\_decoder\\_init](#) ([cnav\\_msg\\_decoder\\_t](#) \*dec)
- `bool` [cnav\\_msg\\_decoder\\_add\\_symbol](#) ([cnav\\_msg\\_decoder\\_t](#) \*dec, `unsigned char` symbol, [cnav\\_msg\\_t](#) \*msg, `uint32_t` \*delay)
- `uint32_t` [crc24q](#) (`const uint8_t *buf`, `uint32_t` len, `uint32_t` crc)
- `uint32_t` [crc24q\\_bits](#) (`uint32_t` crc, `const uint8_t *buf`, `uint32_t` n\_bits, `bool` invert)

### 9.11.4.1 Detailed Description

Utilities for CNAV message decoding by Swift Navigation Inc.

### 9.11.4.2 Macro Definition Documentation

#### 9.11.4.2.1 ABS

```
#define ABS(  
    x)
```

#### Value:

```
((x) < 0 ? -(x) : (x))
```

Definition at line 34 of file [swift\\_common.h](#).

#### 9.11.4.2.2 CLAMP\_DIFF

```
#define CLAMP_DIFF(  
    a,  
    b)
```

**Value:**

$(\text{MAX}((a), (b)) - (b))$

Definition at line 37 of file [swift\\_common.h](#).

#### 9.11.4.2.3 GPS\_L2\_V27\_HISTORY\_LENGTH\_BITS

```
#define GPS_L2_V27_HISTORY_LENGTH_BITS 64
```

Size of the Viterbi decoder history.

Definition at line 40 of file [cnav\\_msg.h](#).

#### 9.11.4.2.4 GPS\_L2C\_V27\_DECODE\_BITS

```
#define GPS_L2C_V27_DECODE_BITS (32)
```

Bits to decode at a time.

Definition at line 44 of file [cnav\\_msg.h](#).

#### 9.11.4.2.5 GPS\_L2C\_V27\_DELAY\_BITS

```
#define GPS_L2C_V27_DELAY_BITS (32)
```

Bits in decoder tail. We ignore them.

Definition at line 46 of file [cnav\\_msg.h](#).

#### 9.11.4.2.6 GPS\_L2C\_V27\_INIT\_BITS

```
#define GPS_L2C_V27_INIT_BITS (32)
```

Bits to accumulate before decoding starts.

Definition at line 42 of file [cnav\\_msg.h](#).

#### 9.11.4.2.7 MAX

```
#define MAX(  
    x,  
    y)
```

**Value:**

$((x) > (y)) ? (x) : (y)$

Definition at line 36 of file [swift\\_common.h](#).

#### 9.11.4.2.8 MIN

```
#define MIN(  
    x,  
    y)
```

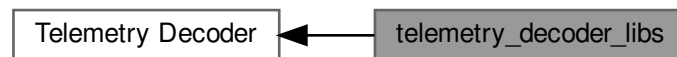
##### Value:

```
((x) < (y)) ? (x) : (y)
```

Definition at line 35 of file [swift\\_common.h](#).

### 9.11.5 telemetry\_decoder\_libs

Collaboration diagram for `telemetry_decoder_libs`:



#### Classes

- class [Tlm\\_Conf](#)
- class [Tlm\\_CRC\\_Stats](#)  
*Class that computes the telemetry CRC statistics.*
- class [Viterbi\\_Decoder](#)  
*Class that implements a Viterbi decoder.*
- class [Viterbi\\_Decoder\\_Sbas](#)  
*Class that implements a Viterbi decoder.*
- class [Serdes\\_Nav\\_Message](#)  
*This class implements serialization and deserialization of [Nav\\_Message\\_Packet](#) objects using Protocol Buffers.*

#### Functions

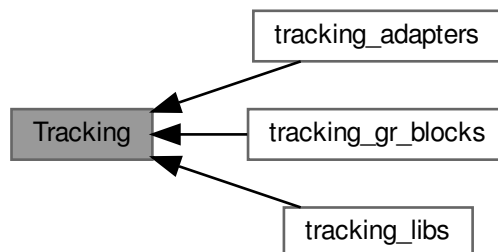
- int **save\_tlm\_matfile** (const std::string &dumpfile)
- bool **tlm\_remove\_file** (const std::string &file\_to\_remove)

#### 9.11.5.1 Detailed Description

Utilities for the decoding of GNSS navigation messages.

## 9.12 Tracking

Collaboration diagram for Tracking:



### Topics

- [tracking\\_adapters](#)
- [tracking\\_gr\\_blocks](#)
- [tracking\\_libs](#)

### 9.12.1 Detailed Description

Classes for GNSS signal tracking.

### 9.12.2 tracking\_adapters

Collaboration diagram for tracking\_adapters:



## Classes

- class [BeidouB1dIIPITracking](#)  
*This class implements a code DLL + carrier PLL tracking loop.*
- class [BeidouB3dIIPITracking](#)  
*This class implements a code DLL + carrier PLL tracking loop.*
- class [GalileoE1dIPIIVemITracking](#)  
*This class Adapts a DLL+PLL VEML (Very Early Minus Late) tracking loop block to a [TrackingInterface](#) for Galileo E1 signals.*
- class [GalileoE1dIPIIVemITrackingFpga](#)  
*This class Adapts a DLL+PLL VEML (Very Early Minus Late) tracking loop block to a [TrackingInterface](#) for Galileo E1 signals.*
- class [GalileoE1TcpConnectorTracking](#)  
*This class implements a code DLL + carrier PLL tracking loop.*
- class [GalileoE5aDIIPITracking](#)  
*This class implements a code DLL + carrier PLL tracking loop.*
- class [GalileoE5aDIIPITrackingFpga](#)  
*This class implements a code DLL + carrier PLL tracking loop.*
- class [GalileoE5bDIIPITracking](#)  
*This class implements a code DLL + carrier PLL tracking loop.*
- class [GalileoE6DIIPITracking](#)  
*This class implements a code DLL + carrier PLL tracking loop.*
- class [GlonassL1CaDIIPICaIdTracking](#)  
*This class implements a code DLL + carrier PLL tracking loop.*
- class [GlonassL1CaDIIPITracking](#)  
*This class implements a code DLL + carrier PLL tracking loop.*
- class [GlonassL2CaDIIPICaIdTracking](#)  
*This class implements a code DLL + carrier PLL tracking loop.*
- class [GlonassL2CaDIIPITracking](#)  
*This class implements a code DLL + carrier PLL tracking loop.*
- class [GpsL1CaDIIPITracking](#)  
*This class implements a code DLL + carrier PLL tracking loop.*
- class [GpsL1CaDIIPITrackingFpga](#)  
*This class implements a code DLL + carrier PLL tracking loop.*
- class [GpsL1CaDIIPITrackingGPU](#)  
*This class implements a code DLL + carrier PLL tracking loop using GPU accelerated functions.*
- class [GpsL1CaGaussianTracking](#)  
*This class implements a code DLL + carrier PLL tracking loop.*
- class [GpsL1CaTcpConnectorTracking](#)  
*This class implements a code DLL + carrier PLL tracking loop.*
- class [GpsL2MDIIPITracking](#)  
*This class implements a code DLL + carrier PLL tracking loop.*
- class [GpsL2MDIIPITrackingFpga](#)  
*This class implements a code DLL + carrier PLL tracking loop.*
- class [GpsL5DIIPITracking](#)  
*This class implements a code DLL + carrier PLL tracking loop.*
- class [GpsL5DIIPITrackingFpga](#)  
*This class implements a code DLL + carrier PLL tracking loop.*

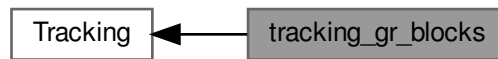
### 9.12.2.1 Detailed Description

Wrap GNU Radio blocks for GNSS signal tracking with a [TrackingInterface](#)



### 9.12.3 tracking\_gr\_blocks

Collaboration diagram for tracking\_gr\_blocks:



#### Classes

- class [dll\\_pll\\_veml\\_tracking](#)  
*This class implements a code DLL + carrier PLL tracking block.*
- class [dll\\_pll\\_veml\\_tracking\\_fpga](#)  
*This class implements a code DLL + carrier PLL tracking block.*
- class [Galileo\\_E1\\_Tcp\\_Connector\\_Tracking\\_cc](#)  
*This class implements a code DLL + carrier PLL VEML (Very Early Minus Late) tracking block for Galileo E1 signals.*
- class [glonass\\_l1\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_cc](#)  
*This class implements a DLL + PLL tracking loop block.*
- class [glonass\\_l1\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_sc](#)  
*This class implements a DLL + PLL tracking loop block.*
- class [Glonass\\_L1\\_Ca\\_Dll\\_Pll\\_Tracking\\_cc](#)  
*This class implements a DLL + PLL tracking loop block.*
- class [glonass\\_l2\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_cc](#)  
*This class implements a DLL + PLL tracking loop block.*
- class [glonass\\_l2\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_sc](#)  
*This class implements a DLL + PLL tracking loop block.*
- class [Glonass\\_L2\\_Ca\\_Dll\\_Pll\\_Tracking\\_cc](#)  
*This class implements a DLL + PLL tracking loop block.*
- class [Gps\\_L1\\_Ca\\_Dll\\_Pll\\_Tracking\\_GPU\\_cc](#)  
*This class implements a DLL + PLL tracking loop block.*
- class [Gps\\_L1\\_Ca\\_Gaussian\\_Tracking\\_cc](#)  
*This class implements a DLL + PLL tracking loop block.*
- class [Gps\\_L1\\_Ca\\_Tcp\\_Connector\\_Tracking\\_cc](#)  
*This class implements a DLL + PLL tracking loop block.*

#### Typedefs

- using [dll\\_pll\\_veml\\_tracking\\_sptr](#) = gnss\_shared\_ptr<[dll\\_pll\\_veml\\_tracking](#)>
- using [dll\\_pll\\_veml\\_tracking\\_fpga\\_sptr](#) = gnss\_shared\_ptr<[dll\\_pll\\_veml\\_tracking\\_fpga](#)>
- using [galileo\\_e1\\_tcp\\_connector\\_tracking\\_cc\\_sptr](#) = gnss\_shared\_ptr<[Galileo\\_E1\\_Tcp\\_Connector\\_Tracking\\_cc](#)>
- using [glonass\\_l1\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_cc\\_sptr](#) = gnss\_shared\_ptr<[glonass\\_l1\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_cc](#)>
- using [glonass\\_l1\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_sc\\_sptr](#) = gnss\_shared\_ptr<[glonass\\_l1\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_sc](#)>
- using [glonass\\_l1\\_ca\\_dll\\_pll\\_tracking\\_cc\\_sptr](#) = gnss\_shared\_ptr<[Glonass\\_L1\\_Ca\\_Dll\\_Pll\\_Tracking\\_cc](#)>
- using [glonass\\_l2\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_cc\\_sptr](#) = gnss\_shared\_ptr<[glonass\\_l2\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_cc](#)>
- using [glonass\\_l2\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_sc\\_sptr](#) = gnss\_shared\_ptr<[glonass\\_l2\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_sc](#)>
- using [glonass\\_l2\\_ca\\_dll\\_pll\\_tracking\\_cc\\_sptr](#) = gnss\_shared\_ptr<[Glonass\\_L2\\_Ca\\_Dll\\_Pll\\_Tracking\\_cc](#)>
- using [gps\\_l1\\_ca\\_dll\\_pll\\_tracking\\_gpu\\_cc\\_sptr](#) = gnss\_shared\_ptr<[Gps\\_L1\\_Ca\\_Dll\\_Pll\\_Tracking\\_GPU\\_cc](#)>
- using [gps\\_l1\\_ca\\_gaussian\\_tracking\\_cc\\_sptr](#) = gnss\_shared\_ptr<[Gps\\_L1\\_Ca\\_Gaussian\\_Tracking\\_cc](#)>
- using [gps\\_l1\\_ca\\_tcp\\_connector\\_tracking\\_cc\\_sptr](#) = gnss\_shared\_ptr<[Gps\\_L1\\_Ca\\_Tcp\\_Connector\\_Tracking\\_cc](#)>

## Functions

- `dll_pll_veml_tracking_sptr` **`dll_pll_veml_make_tracking`** (const [Dll\\_Pll\\_Conf](#) &conf\_)
- `dll_pll_veml_tracking_fpga_sptr` **`dll_pll_veml_make_tracking_fpga`** (const [Dll\\_Pll\\_Conf\\_Fpga](#) &conf\_)
- `galileo_e1_tcp_connector_tracking_cc_sptr` **`galileo_e1_tcp_connector_make_tracking_cc`** (int64\_t fs\_in, uint32\_t vector\_length, bool dump, const std::string &dump\_filename, float pll\_bw\_hz, float dll\_bw\_hz, float early\_late\_space\_chips, float very\_early\_late\_space\_chips, size\_t port\_ch0)
- `glonass_l1_ca_dll_pll_c_aid_tracking_cc_sptr` **`glonass_l1_ca_dll_pll_c_aid_make_tracking_cc`** (int64\_t fs\_in, uint32\_t vector\_length, bool dump, const std::string &dump\_filename, float pll\_bw\_hz, float dll\_bw\_hz, float pll\_bw\_narrow\_hz, float dll\_bw\_narrow\_hz, int32\_t extend\_correlation\_ms, float early\_late\_space\_chips)
- `glonass_l1_ca_dll_pll_c_aid_tracking_sc_sptr` **`glonass_l1_ca_dll_pll_c_aid_make_tracking_sc`** (int64\_t fs\_in, uint32\_t vector\_length, bool dump, const std::string &dump\_filename, float pll\_bw\_hz, float dll\_bw\_hz, float pll\_bw\_narrow\_hz, float dll\_bw\_narrow\_hz, int32\_t extend\_correlation\_ms, float early\_late\_space\_chips)
- `glonass_l1_ca_dll_pll_tracking_cc_sptr` **`glonass_l1_ca_dll_pll_make_tracking_cc`** (int64\_t fs\_in, uint32\_t vector\_length, bool dump, const std::string &dump\_filename, float pll\_bw\_hz, float dll\_bw\_hz, float early\_late\_space\_chips)
- `glonass_l2_ca_dll_pll_c_aid_tracking_cc_sptr` **`glonass_l2_ca_dll_pll_c_aid_make_tracking_cc`** (int64\_t fs\_in, uint32\_t vector\_length, bool dump, const std::string &dump\_filename, float pll\_bw\_hz, float dll\_bw\_hz, float pll\_bw\_narrow\_hz, float dll\_bw\_narrow\_hz, int32\_t extend\_correlation\_ms, float early\_late\_space\_chips)
- `glonass_l2_ca_dll_pll_c_aid_tracking_sc_sptr` **`glonass_l2_ca_dll_pll_c_aid_make_tracking_sc`** (int64\_t fs\_in, uint32\_t vector\_length, bool dump, const std::string &dump\_filename, float pll\_bw\_hz, float dll\_bw\_hz, float pll\_bw\_narrow\_hz, float dll\_bw\_narrow\_hz, int32\_t extend\_correlation\_ms, float early\_late\_space\_chips)
- `glonass_l2_ca_dll_pll_tracking_cc_sptr` **`glonass_l2_ca_dll_pll_make_tracking_cc`** (int64\_t fs\_in, uint32\_t vector\_length, bool dump, const std::string &dump\_filename, float pll\_bw\_hz, float dll\_bw\_hz, float early\_late\_space\_chips)
- `gps_l1_ca_dll_pll_tracking_gpu_cc_sptr` **`gps_l1_ca_dll_pll_make_tracking_gpu_cc`** (int64\_t fs\_in, uint32\_t vector\_length, bool dump, const std::string &dump\_filename, float pll\_bw\_hz, float dll\_bw\_hz, float early\_late\_space\_chips)
- `gps_l1_ca_gaussian_tracking_cc_sptr` **`gps_l1_ca_gaussian_make_tracking_cc`** (uint32\_t order, int64\_t fs\_in, uint32\_t vector\_length, bool dump, const std::string &dump\_filename, float dll\_bw\_hz, float early\_late\_space\_chips, bool bce\_run, uint32\_t bce\_ptrans, uint32\_t bce\_strans, int32\_t bce\_nu, int32\_t bce\_kappa)
- `gps_l1_ca_tcp_connector_tracking_cc_sptr` **`gps_l1_ca_tcp_connector_make_tracking_cc`** (int64\_t fs\_in, uint32\_t vector\_length, bool dump, const std::string &dump\_filename, float early\_late\_space\_chips, size\_t port\_ch0)

### 9.12.3.1 Detailed Description

GNU Radio blocks for GNSS signal tracking.

### 9.12.3.2 Typedef Documentation

#### 9.12.3.2.1 `dll_pll_veml_tracking_fpga_sptr`

```
using dll_pll_veml_tracking_fpga_sptr = gnss_shared_ptr<dll_pll_veml_tracking_fpga>
```

Definition at line 50 of file [dll\\_pll\\_veml\\_tracking\\_fpga.h](#).

#### 9.12.3.2.2 `dll_pll_veml_tracking_sptr`

```
using dll_pll_veml_tracking_sptr = gnss_shared_ptr<dll_pll_veml_tracking>
```

Definition at line 51 of file [dll\\_pll\\_veml\\_tracking.h](#).

#### 9.12.3.2.3 `galileo_e1_tcp_connector_tracking_cc_sptr`

```
using galileo_e1_tcp_connector_tracking_cc_sptr = gnss_shared_ptr<Galileo_E1_Tcp_Connector_Tracking_cc>
```

Definition at line 46 of file [galileo\\_e1\\_tcp\\_connector\\_tracking\\_cc.h](#).

#### 9.12.3.2.4 `glonass_l1_ca_dll_pll_c_aid_tracking_cc_sptr`

```
using glonass_l1_ca_dll_pll_c_aid_tracking_cc_sptr = gnss_shared_ptr<glonass_l1_ca_dll_pll_c_aid_tracking_cc>
```

Definition at line 50 of file [glonass\\_l1\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_cc.h](#).

#### 9.12.3.2.5 `glonass_l1_ca_dll_pll_c_aid_tracking_sc_sptr`

```
using glonass_l1_ca_dll_pll_c_aid_tracking_sc_sptr = gnss_shared_ptr<glonass_l1_ca_dll_pll_c_aid_tracking_sc>
```

Definition at line 51 of file [glonass\\_l1\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_sc.h](#).

#### 9.12.3.2.6 `glonass_l1_ca_dll_pll_tracking_cc_sptr`

```
using glonass_l1_ca_dll_pll_tracking_cc_sptr = gnss_shared_ptr<Glonass_L1_Ca_Dll_Pll_Tracking_cc>
```

Definition at line 48 of file [glonass\\_l1\\_ca\\_dll\\_pll\\_tracking\\_cc.h](#).

#### 9.12.3.2.7 `glonass_l2_ca_dll_pll_c_aid_tracking_cc_sptr`

```
using glonass_l2_ca_dll_pll_c_aid_tracking_cc_sptr = gnss_shared_ptr<glonass_l2_ca_dll_pll_c_aid_tracking_cc>
```

Definition at line 47 of file [glonass\\_l2\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_cc.h](#).

#### 9.12.3.2.8 `glonass_l2_ca_dll_pll_c_aid_tracking_sc_sptr`

```
using glonass_l2_ca_dll_pll_c_aid_tracking_sc_sptr = gnss_shared_ptr<glonass_l2_ca_dll_pll_c_aid_tracking_sc>
```

Definition at line 47 of file [glonass\\_l2\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_sc.h](#).

#### 9.12.3.2.9 `glonass_l2_ca_dll_pll_tracking_cc_sptr`

```
using glonass_l2_ca_dll_pll_tracking_cc_sptr = gnss_shared_ptr<Glonass_L2_Ca_Dll_Pll_Tracking_cc>
```

Definition at line 45 of file [glonass\\_l2\\_ca\\_dll\\_pll\\_tracking\\_cc.h](#).

#### 9.12.3.2.10 `gps_l1_ca_dll_pll_tracking_gpu_cc_sptr`

```
using gps_l1_ca_dll_pll_tracking_gpu_cc_sptr = gnss_shared_ptr<Gps_L1_Ca_Dll_Pll_Tracking_GPU_cc>
```

Definition at line 44 of file [gps\\_l1\\_ca\\_dll\\_pll\\_tracking\\_gpu\\_cc.h](#).

#### 9.12.3.2.11 `gps_l1_ca_gaussian_tracking_cc_sptr`

```
using gps_l1_ca_gaussian_tracking_cc_sptr = gnss_shared_ptr<Gps_L1_Ca_Gaussian_Tracking_cc>
```

Definition at line 54 of file [gps\\_l1\\_ca\\_gaussian\\_tracking\\_cc.h](#).

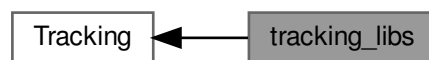
#### 9.12.3.2.12 `gps_l1_ca_tcp_connector_tracking_cc_sptr`

```
using gps_l1_ca_tcp_connector_tracking_cc_sptr = gnss_shared_ptr<Gps_L1_Ca_Tcp_Connector_Tracking_cc>
```

Definition at line 44 of file [gps\\_l1\\_ca\\_tcp\\_connector\\_tracking\\_cc.h](#).

### 9.12.4 `tracking_libs`

Collaboration diagram for `tracking_libs`:



#### Classes

- class [Bayesian\\_estimator](#)  
*Bayesian\_estimator is an estimator of noise characteristics (i.e. mean, covariance)*
- class [Cpu\\_Multicorrelator](#)  
*Class that implements carrier wipe-off and correlators.*
- class [Cpu\\_Multicorrelator\\_16sc](#)  
*Class that implements carrier wipe-off and correlators.*
- class [Cpu\\_Multicorrelator\\_Real\\_Codes](#)  
*Class that implements carrier wipe-off and correlators.*
- struct [GPU\\_Complex](#)
- struct [GPU\\_Complex\\_Short](#)
- class [cuda\\_multicorrelator](#)  
*Class that implements carrier wipe-off and correlators using NVIDIA CUDA GPU accelerators.*
- class [Dll\\_Pll\\_Conf](#)
- class [Dll\\_Pll\\_Conf\\_Fpga](#)
- class [Exponential\\_Smoother](#)

- Class that implements a first-order exponential smoother.*

  - class [Fpga\\_Multicorrelator\\_8sc](#)
- Class that implements carrier wipe-off and correlators.*

  - class [ModelFunction](#)
  - class [CubatureFilter](#)
  - class [UnscentedFilter](#)
  - class [Tcp\\_Communication](#)
- TCP communication class.*

  - class [Tcp\\_Packet\\_Data](#)
- Class that implements a TCP data packet.*

  - class [Tracking\\_2nd\\_DLL\\_filter](#)
- This class implements a 2nd order DLL filter for code tracking loop.*

  - class [Tracking\\_2nd\\_PLL\\_filter](#)
- This class implements a 2nd order PLL filter for carrier tracking loop.*

  - class [Tracking\\_FLL\\_PLL\\_filter](#)
- This class implements a hybrid FLL and PLL filter for tracking carrier loop.*

  - class [Tracking\\_loop\\_filter](#)
- This class implements a generic 1st, 2nd or 3rd order loop filter.*

## Macros

- `#define NUM_TX_VARIABLES_GALILEO_E1 13`
- `#define NUM_TX_VARIABLES_GPS_L1_CA 9`
- `#define NUM_RX_VARIABLES 4`

## Typedefs

- using [b\\_io\\_context](#) = boost::asio::io\_service

## Functions

- float [cn0\\_svn\\_estimator](#) (const gr\_complex \*Prompt\_buffer, int length, float coh\_integration\_time\_s)  
*cn0\_svn\_estimator is a Carrier-to-Noise (CN0) estimator based on the Signal-to-Noise Variance (SNV) estimator*
- float [cn0\\_m2m4\\_estimator](#) (const gr\_complex \*Prompt\_buffer, int length, float coh\_integration\_time\_s)  
*cn0\_m2m4\_estimator is a Carrier-to-Noise (CN0) estimator based on the Second- and Fourth-Order Moments Method (M2M4)*
- float [carrier\\_lock\\_detector](#) (const gr\_complex \*Prompt\_buffer, int length)  
*A carrier lock detector.*
- double [fil\\_four\\_quadrant\\_atan](#) (gr\_complex prompt\_s1, gr\_complex prompt\_s2, double t1, double t2)
- double [fil\\_diff\\_atan](#) (gr\_complex prompt\_s1, gr\_complex prompt\_s2, double t1, double t2)
- double [phase\\_unwrap](#) (double phase\_rad)  
*Phase unwrapping function, input is [rad].*
- double [pll\\_four\\_quadrant\\_atan](#) (gr\_complex prompt\_s1)  
*PLL four quadrant arctan discriminator.*
- double [pll\\_cloop\\_two\\_quadrant\\_atan](#) (gr\_complex prompt\_s1)  
*PLL Costas loop two quadrant arctan discriminator.*
- double [dll\\_nc\\_e\\_minus\\_l\\_normalized](#) (gr\_complex early\_s1, gr\_complex late\_s1, float spc=0.5, float slope=1.0, float y\_intercept=1.0)  
*DLL Noncoherent Early minus Late envelope normalized discriminator.*

- double [dll\\_nc\\_vemlp\\_normalized](#) (gr\_complex very\_early\_s1, gr\_complex early\_s1, gr\_complex late\_s1, gr\_complex very\_late\_s1)  
*DLL Noncoherent Very Early Minus Late Power (VEMLP) normalized discriminator.*
- template<typename Fun>  
double [CalculateSlope](#) (Fun &&f, double x)
- template<typename Fun>  
double [CalculateSlopeAbs](#) (Fun &&f, double x)
- template<typename Fun>  
double [GetYIntercept](#) (Fun &&f, double x)
- template<typename Fun>  
double [GetYInterceptAbs](#) (Fun &&f, double x)
- template<int M = 1, int N = M>  
double [SinBocCorrelationFunction](#) (double offset\_in\_chips)
- template<int M = 1, int N = M>  
double [CosBocCorrelationFunction](#) (double offset\_in\_chips)

#### 9.12.4.1 Detailed Description

Utilities for GNSS signal tracking.

#### 9.12.4.2 Macro Definition Documentation

##### 9.12.4.2.1 NUM\_RX\_VARIABLES

```
#define NUM_RX_VARIABLES 4
```

Definition at line 39 of file [tcp\\_communication.h](#).

##### 9.12.4.2.2 NUM\_TX\_VARIABLES\_GALILEO\_E1

```
#define NUM_TX_VARIABLES_GALILEO_E1 13
```

Definition at line 37 of file [tcp\\_communication.h](#).

##### 9.12.4.2.3 NUM\_TX\_VARIABLES\_GPS\_L1\_CA

```
#define NUM_TX_VARIABLES_GPS_L1_CA 9
```

Definition at line 38 of file [tcp\\_communication.h](#).

#### 9.12.4.3 Typedef Documentation

##### 9.12.4.3.1 b\_io\_context

```
using b_io_context = boost::asio::io_service
```

Definition at line 34 of file [tcp\\_communication.h](#).

### 9.12.4.4 Function Documentation

#### 9.12.4.4.1 CalculateSlope()

```
template<typename Fun>
double CalculateSlope (
    Fun && f,
    double x)
```

Definition at line 112 of file [tracking\\_discriminators.h](#).

#### 9.12.4.4.2 CalculateSlopeAbs()

```
template<typename Fun>
double CalculateSlopeAbs (
    Fun && f,
    double x)
```

Definition at line 120 of file [tracking\\_discriminators.h](#).

#### 9.12.4.4.3 carrier\_lock\_detector()

```
float carrier_lock_detector (
    const gr_complex * Prompt_buffer,
    int length)
```

A carrier lock detector.

The Carrier Phase Lock Detector block uses the estimate of the cosine of twice the carrier phase error is given by

$$C2\phi = \frac{NBD}{NBP}, \quad (9.1)$$

where  $NBD = (\sum_{i=0}^{N-1} |Im(Pc(i))|)^2 + (\sum_{i=0}^{N-1} |Re(Pc(i))|)^2$ ,  $NBP = \sum_{i=0}^{N-1} Im(Pc(i))^2 - \sum_{i=0}^{N-1} Re(Pc(i))^2$ , and  $Pc(i)$  is the prompt correlator output for the sample index  $i$ . Ref: Van Dierendonck, A.J. (1996), Global Positioning System: Theory and Applications, Volume I, Chapter 8: GPS Receivers, AJ Systems, Los Altos, CA 94024. Inc.: 329-407.

#### 9.12.4.4.4 cn0\_m2m4\_estimator()

```
float cn0_m2m4_estimator (
    const gr_complex * Prompt_buffer,
    int length,
    float coh_integration_time_s)
```

cn0\_m2m4\_estimator is a Carrier-to-Noise (CN0) estimator based on the Second- and Fourth-Order Moments Method (M2M4)

Signal-to-Noise (SNR) ( $\rho$ ) estimator using the Moments Method:

$$\hat{\rho} = \frac{\sqrt{2\hat{M}_2^2 - \hat{M}_4}}{\hat{M}_2 - \sqrt{2\hat{M}_2^2 - \hat{M}_4}}, \quad (9.2)$$

where  $\hat{M}_2 = \frac{1}{N} \sum_{k=0}^{K-1} |P[k]|^2$ ,  $\hat{M}_4 = \frac{1}{K} \sum_{k=0}^{K-1} |P[k]|^4$ ,  $|\cdot|$  is the absolute value, and  $P[k]$  is the prompt correlator output for the sample index  $k$ .

The SNR value is converted to CN0 [dB-Hz] taking into account the coherent integration time, using the following formula:

$$CN0_{dB} = 10 * \log(\hat{\rho}) - 10 * \log(T_{int}), \quad (9.3)$$

where  $T_{int}$  is the coherent integration time, in seconds.

Ref: D. R. Pauluzzi, N. C. Beaulieu, "A comparison of SNR estimation techniques for the AWGN channel," IEEE Trans. on Comm., vol. 48, no. 10, pp. 1681–1691, Oct. 2000.

#### 9.12.4.4.5 `cn0_svn_estimator()`

```
float cn0_svn_estimator (
    const gr_complex * Prompt_buffer,
    int length,
    float coh_integration_time_s)
```

`cn0_svn_estimator` is a Carrier-to-Noise (CN0) estimator based on the Signal-to-Noise Variance (SNV) estimator

Signal-to-Noise (SNR) ( $\rho$ ) estimator using the Signal-to-Noise Variance (SNV) estimator:

$$\hat{\rho} = \frac{\hat{P}_s}{\hat{P}_n} = \frac{\hat{P}_s}{\hat{P}_{tot} - \hat{P}_s}, \quad (9.4)$$

where  $\hat{P}_s = \left( \frac{1}{N} \sum_{i=0}^{N-1} |Re(Pc(i))| \right)^2$  is the estimation of the signal power,  $\hat{P}_{tot} = \frac{1}{N} \sum_{i=0}^{N-1} |Pc(i)|^2$  is the estimator of the total power,  $|\cdot|$  is the absolute value,  $Re(\cdot)$  stands for the real part of the value, and  $Pc(i)$  is the prompt correlator output for the sample index  $i$ .

The SNR value is converted to CN0 [dB-Hz], taking into account the coherent integration time, using the following formula:

$$CN0_{dB} = 10 * \log(\hat{\rho}) - 10 * \log(T_{int}), \quad (9.5)$$

where  $T_{int}$  is the coherent integration time, in seconds.

Ref: Marco Pini, Emanuela Falletti and Maurizio Fantino, "Performance Evaluation of C/N0 Estimators using a Real Time GNSS Software Receiver," IEEE 10th International Symposium on Spread Spectrum Techniques and Applications, pp.28-30, August 2008.

#### 9.12.4.4.6 `CosBocCorrelationFunction()`

```
template<int M = 1, int N = M>
double CosBocCorrelationFunction (
    double offset_in_chips)
```

Definition at line 171 of file [tracking\\_discriminators.h](#).

#### 9.12.4.4.7 `dll_nc_e_minus_l_normalized()`

```
double dll_nc_e_minus_l_normalized (
    gr_complex early_s1,
    gr_complex late_s1,
    float spc = 0.5,
    float slope = 1.0,
    float y_intercept = 1.0)
```

DLL Noncoherent Early minus Late envelope normalized discriminator.

DLL Noncoherent Early minus Late envelope normalized discriminator:

$$error = \frac{y_{intercept} - slope * \epsilon \frac{E - L}{E + L}}{slope} \quad (9.6)$$

where  $E = \sqrt{I_{ES}^2 + Q_{ES}^2}$  is the Early correlator output absolute value and  $L = \sqrt{I_{LS}^2 + Q_{LS}^2}$  is the Late correlator output absolute value. The output is in [chips].



#### 9.12.4.4.8 dll\_nc\_vemlp\_normalized()

```
double dll_nc_vemlp_normalized (
    gr_complex very_early_s1,
    gr_complex early_s1,
    gr_complex late_s1,
    gr_complex very_late_s1)
```

DLL Noncoherent Very Early Minus Late Power (VEMLP) normalized discriminator.

DLL Noncoherent Very Early Minus Late Power (VEMLP) normalized discriminator, using the outputs of four correlators, Very Early (VE), Early (E), Late (L) and Very Late (VL):

$$error = \frac{E - L}{E + L}, \quad (9.7)$$

where  $E = \sqrt{I_{VE}^2 + Q_{VE}^2 + I_E^2 + Q_E^2}$  and  $L = \sqrt{I_{VL}^2 + Q_{VL}^2 + I_L^2 + Q_L^2}$ . The output is in [chips].

#### 9.12.4.4.9 fll\_four\_quadrant\_atan()

```
double fll_four_quadrant_atan (
    gr_complex prompt_s1,
    gr_complex prompt_s2,
    double t1,
    double t2)
```

brief FLL four quadrant arctan discriminator

FLL four quadrant arctan discriminator:

$$\frac{\phi_2 - \phi_1}{t_2 - t_1} = \frac{ATAN2(cross, dot)}{t_1 - t_2}, \quad (9.8)$$

where  $cross = I_{PS1}Q_{PS2} - I_{PS2}Q_{PS1}$  and  $dot = I_{PS1}I_{PS2} + Q_{PS1}Q_{PS2}$ ,  $I_{PS1}, Q_{PS1}$  are the inphase and quadrature prompt correlator outputs respectively at sample time  $t_1$ , and  $I_{PS2}, Q_{PS2}$  are the inphase and quadrature prompt correlator outputs respectively at sample time  $t_2$ . The output is in [radians/second].

#### 9.12.4.4.10 GetYIntercept()

```
template<typename Fun>
double GetYIntercept (
    Fun && f,
    double x)
```

Definition at line 128 of file [tracking\\_discriminators.h](#).

#### 9.12.4.4.11 GetYInterceptAbs()

```
template<typename Fun>
double GetYInterceptAbs (
    Fun && f,
    double x)
```

Definition at line 137 of file [tracking\\_discriminators.h](#).

**9.12.4.4.12 phase\_unwrap()**

```
double phase_unwrap (
    double phase_rad)
```

Phase unwrapping function, input is [rad].

**9.12.4.4.13 pll\_cloop\_two\_quadrant\_atan()**

```
double pll_cloop_two_quadrant_atan (
    gr_complex prompt_sl)
```

PLL Costas loop two quadrant arctan discriminator.

PLL Costas loop two quadrant arctan discriminator:

$$\phi = ATAN\left(\frac{Q_{PS}}{I_{PS}}\right), \quad (9.9)$$

where  $I_{PS1}$ ,  $Q_{PS1}$  are the inphase and quadrature prompt correlator outputs respectively. The output is in [radians].

**9.12.4.4.14 pll\_four\_quadrant\_atan()**

```
double pll_four_quadrant_atan (
    gr_complex prompt_sl)
```

PLL four quadrant arctan discriminator.

PLL four quadrant arctan discriminator:

$$\phi = ATAN2(Q_{PS}, I_{PS}), \quad (9.10)$$

where  $I_{PS1}$ ,  $Q_{PS1}$  are the inphase and quadrature prompt correlator outputs respectively. The output is in [radians].

**9.12.4.4.15 SinBocCorrelationFunction()**

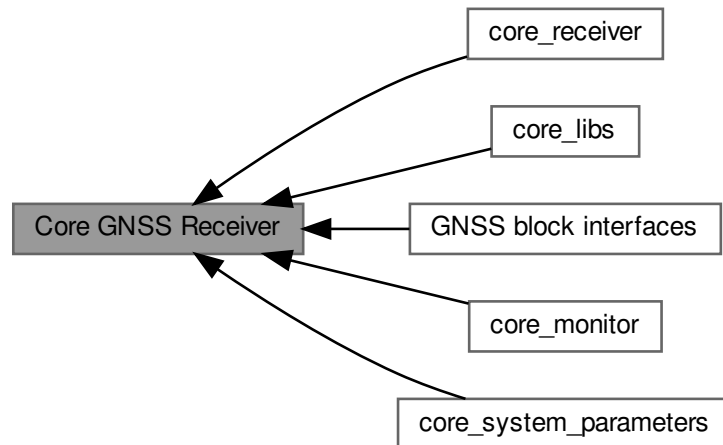
```
template<int M = 1, int N = M>
double SinBocCorrelationFunction (
    double offset_in_chips)
```

Definition at line 150 of file [tracking\\_discriminators.h](#).

## 9.13 Core GNSS Receiver

Header file of the interface to a `signal_source` GNSS block.

Collaboration diagram for Core GNSS Receiver:



### Topics

- [GNSS block interfaces](#)
- [core\\_libs](#)
- [core\\_monitor](#)
- [core\\_receiver](#)
- [core\\_system\\_parameters](#)

### 9.13.1 Detailed Description

Header file of the interface to a `signal_source` GNSS block.

`\signal_source_interface.h`

#### Author

Jim Melton, 2020. [jim.melton\(at\)sncorp.com](mailto:jim.melton@sncorp.com)

This header file contains the interface to an abstract class for signal sources. Since all its methods are virtual, this class cannot be instantiated directly, and a subclass can only be instantiated directly if all inherited pure virtual methods have been implemented by that class or a parent class.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)

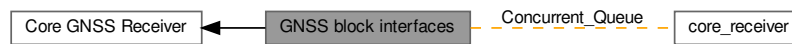
GNSS-SDR is a software defined Global Navigation Satellite Systems receiver

This file is part of GNSS-SDR.

SPDX-License-Identifier: GPL-3.0-or-later  
Core GNSS Receiver.

### 9.13.2 GNSS block interfaces

Collaboration diagram for GNSS block interfaces:



#### Classes

- class [Concurrent\\_Queue< Data >](#)  
*This class implements a thread-safe std::queue.*
- class [AcquisitionInterface](#)  
*This abstract class represents an interface to an acquisition GNSS block.*
- class [ChannelInterface](#)  
*This abstract class represents an interface to a channel GNSS block.*
- class [ConfigurationInterface](#)  
*This abstract class represents an interface to configuration parameters.*
- class [GNSSBlockInterface](#)  
*This abstract class represents an interface to GNSS blocks.*
- class [ObservablesInterface](#)  
*This abstract class represents an interface to an observables block.*
- class [PvtInterface](#)  
*This class represents an interface to a PVT block.*
- class [SignalSourceInterface](#)  
*This abstract class represents an interface to signal\_source GNSS block.*
- class [TelemetryDecoderInterface](#)  
*This abstract class represents an interface to a navigation GNSS block.*
- class [TrackingInterface](#)  
*This abstract class represents an interface to a tracking block.*

#### Typedefs

- `template<typename T>`  
using [gnss\\_shared\\_ptr](#) = boost::shared\_ptr<T>

#### Functions

- `template<typename C, typename... Args>`  
gnss\_shared\_ptr< C > [gnss\\_make\\_shared](#) (Args &&... args)
- `virtual size_t SignalSourceInterface::getRfChannels () const =0`

#### 9.13.2.1 Detailed Description

GNSS block interfaces.

GNSS block interfaces.

#### 9.13.2.2 Typedef Documentation

##### 9.13.2.2.1 gnss\_shared\_ptr

```
template<typename T>
using gnss_shared_ptr = boost::shared_ptr<T>
Definition at line 50 of file gnss\_block\_interface.h.
```

### 9.13.2.3 Function Documentation

#### 9.13.2.3.1 gnss\_make\_shared()

```
template<typename C, typename... Args>
gnss_shared_ptr< C > gnss_make_shared (
    Args &&... args)
```

Definition at line 52 of file [gnss\\_block\\_interface.h](#).

#### 9.13.2.3.2 SignalSourceInterface()

```
SignalSourceInterface::SignalSourceInterface () [inline], [protected]
```

Definition at line 57 of file [signal\\_source\\_interface.h](#).

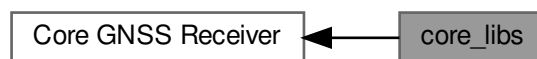
#### 9.13.2.3.3 ~SignalSourceInterface()

```
SignalSourceInterface::~SignalSourceInterface () [inline]
```

Definition at line 63 of file [signal\\_source\\_interface.h](#).

## 9.13.3 core\_libs

Collaboration diagram for core\_libs:



### Classes

- class [Channel\\_Event](#)
- class [channel\\_status\\_msg\\_receiver](#)  
*GNU Radio block that receives asynchronous channel messages from tlm blocks.*
- class [Command\\_Event](#)
- class [galileo\\_e6\\_has\\_msg\\_receiver](#)  
*GNU Radio block that receives asynchronous Galileo HAS message pages from the telemetry blocks, stores them in memory, and decodes HAS messages when enough data have been received. The decoded HAS message is sent to the PVT block.*
- class [galileo\\_tow\\_map](#)
- class [Gnss\\_Crypto](#)  
*Class implementing cryptographic functions for Navigation Message Authentication.*
- class [gnss\\_sdr\\_fpga\\_sample\\_counter](#)
- class [gnss\\_sdr\\_sample\\_counter](#)
- class [Gnss\\_Sdr\\_Supl\\_Client](#)  
*class that implements a C++ interface to external Secure User Location Protocol (SUPL) client library..*
- class [gnss\\_sdr\\_time\\_counter](#)
- class [INIReader](#)  
*Read an INI file into easy-to-access name/value pairs. (Note that I've gone for simplicity here rather than speed, but it should be pretty decent.)*
- class [nav\\_message\\_monitor](#)  
*GNU Radio block that receives asynchronous [Nav\\_Message\\_Packet](#) objects from the telemetry blocks and sends them via UDP.*

- class [Nav\\_Message\\_Packet](#)
- class [Nav\\_Message\\_Udp\\_Sink](#)
- class [Osnma\\_Helper](#)
- class [osnma\\_msg\\_receiver](#)

*GNU Radio block that receives asynchronous OSNMA messages from the telemetry blocks, stores them in memory, and decodes OSNMA info when enough data have been received. The decoded OSNMA data is sent to the PVT block.*

- class [OSNMA\\_NavDataManager](#)

*Class for managing OSNMA navigation data.*

- class [StringConverter](#)

*Class that interprets the contents of a string and converts it into different types.*

## Macros

- `#define INI_ALLOW_MULTILINE 1`

## Typedefs

- using [channel\\_event\\_sptr](#) = std::shared\_ptr<[Channel\\_Event](#)>
- using [channel\\_status\\_msg\\_receiver\\_sptr](#) = gnss\_shared\_ptr<[channel\\_status\\_msg\\_receiver](#)>
- using [command\\_event\\_sptr](#) = std::shared\_ptr<[Command\\_Event](#)>
- using [galileo\\_e6\\_has\\_msg\\_receiver\\_sptr](#) = gnss\_shared\_ptr<[galileo\\_e6\\_has\\_msg\\_receiver](#)>
- using [galileo\\_tow\\_map\\_sptr](#) = gnss\_shared\_ptr<[galileo\\_tow\\_map](#)>
- using [gnss\\_sdr\\_fpga\\_sample\\_counter\\_sptr](#) = gnss\_shared\_ptr<[gnss\\_sdr\\_fpga\\_sample\\_counter](#)>
- using [gnss\\_sdr\\_sample\\_counter\\_sptr](#) = gnss\_shared\_ptr<[gnss\\_sdr\\_sample\\_counter](#)>
- using [gnss\\_sdr\\_time\\_counter\\_sptr](#) = std::shared\_ptr<[gnss\\_sdr\\_time\\_counter](#)>
- using [nav\\_message\\_monitor\\_sptr](#) = gnss\_shared\_ptr<[nav\\_message\\_monitor](#)>
- using [b\\_io\\_context](#) = boost::asio::io\_service
- using [osnma\\_msg\\_receiver\\_sptr](#) = gnss\_shared\_ptr<[osnma\\_msg\\_receiver](#)>

## Functions

- [channel\\_event\\_sptr](#) **channel\_event\_make** (int channel\_id, int event\_type)
- [channel\\_status\\_msg\\_receiver\\_sptr](#) **channel\_status\_msg\_receiver\_make** ()
- [command\\_event\\_sptr](#) **command\_event\_make** (int command\_id, int event\_type)
- [galileo\\_e6\\_has\\_msg\\_receiver\\_sptr](#) **galileo\_e6\_has\_msg\_receiver\_make** ()
- [galileo\\_tow\\_map\\_sptr](#) **galileo\_tow\_map\_make** ()
- [gnss\\_sdr\\_fpga\\_sample\\_counter\\_sptr](#) **gnss\_sdr\_make\_fpga\_sample\_counter** (double \_fs, int32\_t \_↵ interval\_ms)
- [gnss\\_sdr\\_sample\\_counter\\_sptr](#) **gnss\_sdr\_make\_sample\_counter** (double \_fs, int32\_t \_interval\_ms, size\_t ↵ \_t\_size)
- [gnss\\_sdr\\_time\\_counter\\_sptr](#) **gnss\_sdr\_make\_time\_counter** ()
- int [ini\\_parse](#) (const char \*filename, int(\*handler)(void \*user, const char \*section, const char \*name, const char \*value), void \*user)  
*Parse given INI-style file. May have [section]s, name=value pairs (whitespace stripped), and comments starting with ';' (semicolon). Section is "" if name=value pair parsed before any section heading.*
- [nav\\_message\\_monitor\\_sptr](#) **nav\_message\_monitor\_make** (const std::vector< std::string > &addresses, uint16\_t port)
- [osnma\\_msg\\_receiver\\_sptr](#) **osnma\_msg\_receiver\_make** (const std::string &pemFilePath, const std::string &merkleFilePath, bool strict\_mode=false)
- const std::string **uio\_dir** ("/sys/class/uio/")
- const std::string **uio\_filename** ("uio")
- const std::string **uio\_subdir\_name** ("/name")
- int32\_t **find\_uio\_dev\_file\_name** (std::string &device\_file\_name, const std::string &device\_name, uint32\_t ↵ device\_num)

*This function finds the uio device driver device file name out of the device name and the device number.*

### 9.13.3.1 Detailed Description

Utilities for the core GNSS receiver.

### 9.13.3.2 Macro Definition Documentation

#### 9.13.3.2.1 INI\_ALLOW\_MULTILINE

```
#define INI_ALLOW_MULTILINE 1
```

Definition at line 65 of file [ini.h](#).

### 9.13.3.3 Typedef Documentation

#### 9.13.3.3.1 b\_io\_context

```
using b_io_context = boost::asio::io_service
```

Definition at line 36 of file [nav\\_message\\_udp\\_sink.h](#).

#### 9.13.3.3.2 channel\_event\_sptr

```
using channel_event_sptr = std::shared_ptr<Channel_Event>
```

Definition at line 30 of file [channel\\_event.h](#).

#### 9.13.3.3.3 channel\_status\_msg\_receiver\_sptr

```
using channel_status_msg_receiver_sptr = gnss_shared_ptr<channel_status_msg_receiver>
```

Definition at line 37 of file [channel\\_status\\_msg\\_receiver.h](#).

#### 9.13.3.3.4 command\_event\_sptr

```
using command_event_sptr = std::shared_ptr<Command_Event>
```

Definition at line 30 of file [command\\_event.h](#).

#### 9.13.3.3.5 galileo\_e6\_has\_msg\_receiver\_sptr

```
using galileo_e6_has_msg_receiver_sptr = gnss_shared_ptr<galileo_e6_has_msg_receiver>
```

Definition at line 46 of file [galileo\\_e6\\_has\\_msg\\_receiver.h](#).

#### 9.13.3.3.6 galileo\_tow\_map\_sptr

```
using galileo_tow_map_sptr = gnss_shared_ptr<galileo_tow_map>
```

Definition at line 34 of file [galileo\\_tow\\_map.h](#).

#### 9.13.3.3.7 gnss\_sdr\_fpga\_sample\_counter\_sptr

```
using gnss_sdr_fpga_sample_counter_sptr = gnss_shared_ptr<gnss_sdr_fpga_sample_counter>
```

Definition at line 36 of file [gnss\\_sdr\\_fpga\\_sample\\_counter.h](#).

#### 9.13.3.3.8 gnss\_sdr\_sample\_counter\_sptr

```
using gnss_sdr_sample_counter_sptr = gnss_shared_ptr<gnss_sdr_sample_counter>
```

Definition at line 36 of file [gnss\\_sdr\\_sample\\_counter.h](#).

#### 9.13.3.3.9 gnss\_sdr\_time\_counter\_sptr

```
using gnss_sdr_time_counter_sptr = std::shared_ptr<gnss_sdr_time_counter>
```

Definition at line 34 of file [gnss\\_sdr\\_time\\_counter.h](#).

#### 9.13.3.3.10 nav\_message\_monitor\_sptr

```
using nav_message_monitor_sptr = gnss_shared_ptr<nav_message_monitor>
```

Definition at line 37 of file [nav\\_message\\_monitor.h](#).

#### 9.13.3.3.11 osnma\_msg\_receiver\_sptr

using osnma\_msg\_receiver\_sptr = gnss\_shared\_ptr<osnma\_msg\_receiver>  
 Definition at line 51 of file [osnma\\_msg\\_receiver.h](#).

### 9.13.3.4 Function Documentation

#### 9.13.3.4.1 find\_uio\_dev\_file\_name()

```
int32_t find_uio_dev_file_name (
    std::string & device_file_name,
    const std::string & device_name,
    uint32_t device_num)
```

This function finds the uio device driver device file name out of the device name and the device number.

#### 9.13.3.4.2 ini\_parse()

```
int ini_parse (
    const char * filename,
    int(* handler ) (void *user, const char *section, const char *name, const char
    *value),
    void * user)
```

Parse given INI-style file. May have [section]s, name=value pairs (whitespace stripped), and comments starting with ';' (semicolon). Section is "" if name=value pair parsed before any section heading.

For each name=value pair parsed, call handler function with given user pointer as well as section, name, and value (data only valid for duration of handler call). Handler should return nonzero on success, zero on error.

Returns 0 on success, line number of first error on parse error, on -1 on file open error.

### 9.13.4 core\_monitor

Collaboration diagram for core\_monitor:



### Classes

- class [gnss\\_synchro\\_monitor](#)  
*This class implements a monitoring block which allows sending a data stream with the receiver internal parameters ([Gnss\\_Synchro](#) objects) to local or remote clients over UDP.*
- class [Gnss\\_Synchro\\_Udp\\_Sink](#)  
*This class sends serialized [Gnss\\_Synchro](#) objects over UDP to one or multiple endpoints.*

### Typedefs

- using [gnss\\_synchro\\_monitor\\_sptr](#) = gnss\_shared\_ptr<[gnss\\_synchro\\_monitor](#)>
- using [b\\_io\\_context](#) = boost::asio::io\_service

### Functions

- gnss\_synchro\_monitor\_sptr **gnss\_synchro\_make\_monitor** (int n\_channels, int decimation\_factor, const std::vector< std::string > &udp\_ports, const std::vector< std::string > &udp\_addresses, bool enable\_↵ protobuf)



### 9.13.4.1 Detailed Description

Classes for the [Gnss\\_Synchro](#) monitor.

### 9.13.4.2 Typedef Documentation

#### 9.13.4.2.1 b\_io\_context

using `b_io_context = boost::asio::io_service`  
 Definition at line 38 of file [gnss\\_synchro\\_udp\\_sink.h](#).

#### 9.13.4.2.2 gnss\_synchro\_monitor\_sptr

using `gnss_synchro_monitor_sptr = gnss_shared_ptr<gnss_synchro_monitor>`  
 Definition at line 40 of file [gnss\\_synchro\\_monitor.h](#).

## 9.13.5 core\_receiver

Collaboration diagram for `core_receiver`:



### Classes

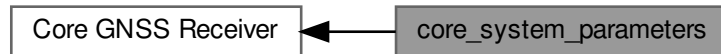
- class [Concurrent\\_Map< Data >](#)  
*This class implements a thread-safe `std::map`.*
- class [Concurrent\\_Queue< Data >](#)  
*This class implements a thread-safe `std::queue`.*
- class [ControlThread](#)  
*This class represents the main thread of the application, so the name is [ControlThread](#). This is the GNSS Receiver Control Plane: it connects the flowgraph, starts running it, and while it does not stop, reads the control messages generated by the blocks, processes them, and applies the corresponding actions.*
- class [FileConfiguration](#)  
*This class is an implementation of the interface [ConfigurationInterface](#).*
- class [GNSSBlockFactory](#)  
*Class that produces all kinds of GNSS blocks.*
- class [GNSSFlowgraph](#)  
*This class represents a GNSS flow graph.*
- class [InMemoryConfiguration](#)  
*This class is an implementation of the interface [ConfigurationInterface](#).*
- class [TcpCmdInterface](#)

### 9.13.5.1 Detailed Description

Classes for the core GNSS receiver.

### 9.13.6 core\_system\_parameters

Collaboration diagram for core\_system\_parameters:



#### Classes

- class [Agnss\\_Ref\\_Location](#)  
*Interface of an Assisted GNSS REFERENCE LOCATION storage.*
- class [Agnss\\_Ref\\_Time](#)  
*Interface of an Assisted GNSS REFERENCE TIME storage.*
- class [Beidou\\_Dnav\\_Almanac](#)  
*This class is a storage for the BeiDou D1 almanac.*
- class [Beidou\\_Dnav\\_Ephemeris](#)  
*This is a storage class for the Beidou SV ephemeris data as described in BeiDou Navigation Satellite System Signal In Space Interface Control Document Open Service Signal B1I (Version 3.0)*
- class [Beidou\\_Dnav\\_Iono](#)  
*This class is a storage for the BEIDOU IONOSPHERIC data as described in ICD v2.1.*
- class [Beidou\\_Dnav\\_Navigation\\_Message](#)  
*This class decodes a BeiDou D1 NAV Data message.*
- class [Beidou\\_Dnav\\_Utc\\_Model](#)  
*This class is a storage for the BeiDou DNAV UTC Model.*
- class [Galileo\\_Almanac](#)  
*This class is a storage for the Galileo SV ALMANAC data.*
- class [Galileo\\_Almanac\\_Helper](#)  
*This class is a storage for the GALILEO ALMANAC data as described in GALILEO ICD.*
- class [Galileo\\_Cnav\\_Message](#)  
*This class handles the Galileo CNAV Data message, as described in the Galileo High Accuracy Service Signal-In-Space Interface Control Document (HAS SIS ICD) Issue 1.0, May 2022.*
- class [Galileo\\_Ephemeris](#)  
*This class is a storage and orbital model functions for the Galileo SV ephemeris data as described in Galileo ICD paragraph 5.1.1.*
- class [Galileo\\_Fnav\\_Message](#)  
*This class handles the Galileo F/NAV Data message, as described in the Galileo Open Service Signal in Space Interface Control Document (OS SIS ICD), Issue 2.0 (Jan. 2021). See [https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo\\_OS\\_SIS\\_ICD\\_v2.0.pdf](https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.0.pdf).*
- struct [mt1\\_header](#)
- class [Galileo\\_HAS\\_data](#)  
*This class is a storage for Galileo HAS message type 1, as defined in Galileo High Accuracy Service Signal-In-Space Interface Control Document (HAS SIS ICD) Issue 1.0, May 2022. See [https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo\\_HAS\\_SIS\\_ICD\\_v1.0.pdf](https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_HAS_SIS_ICD_v1.0.pdf).*
- class [Galileo\\_HAS\\_page](#)  
*This class is a storage for Galileo HAS message page, as defined in Galileo High Accuracy Service Signal-In-Space Interface Control Document (HAS SIS ICD) Issue 1.0, May 2022.*
- class [OSNMA\\_msg](#)  
*This class fills the [OSNMA\\_msg](#) structure with the data received from the telemetry blocks.*

- class [Galileo\\_Inav\\_Message](#)  
*This class handles the Galileo I/NAV Data message, as described in the Galileo Open Service Signal in Space Interface Control Document (OS SIS ICD), Issue 2.0 (Jan. 2021). See [https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo\\_OS\\_SIS\\_ICD\\_v2.0.pdf](https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.0.pdf).*
- class [Galileo\\_Iono](#)  
*This class is a storage for the GALILEO IONOSPHERIC data as described in Galileo ICD paragraph 5.1.6.*
- class [Galileo\\_ISM](#)  
*This class is a storage for the GALILEO Integrity Support Message as described in Galileo ICD paragraph 5.2.*
- class [Mack\\_lookup](#)
- class [Galileo\\_Reduced\\_CED](#)  
*This class manages the Galileo Reduced Clock and Ephemeris Data.*
- class [Galileo\\_Utc\\_Model](#)  
*This class is a storage for the GALILEO UTC MODEL data as described in Galileo ICD [https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo\\_OS\\_SIS\\_ICD\\_v2.0.pdf](https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.0.pdf) paragraph 5.1.7.*
- class [Glonass\\_Gnav\\_Almanac](#)  
*This class is a storage for the GLONASS SV ALMANAC data as described GLONASS ICD (Edition 5.1)*
- class [Glonass\\_Gnav\\_Ephemeris](#)  
*This class is a storage and orbital model functions for the GLONASS SV ephemeris data as described in GLONASS ICD (Edition 5.1)*
- class [Glonass\\_Gnav\\_Navigation\\_Message](#)  
*This class decodes a GLONASS GNAV Data message as described in GLONASS ICD (Edition 5.1)*
- class [Glonass\\_Gnav\\_Utc\\_Model](#)  
*This class is a storage for the GLONASS GNAV UTC MODEL data as described in GLONASS ICD (Edition 5.1)*
- class [Gnss\\_Almanac](#)  
*Base class for GNSS almanac storage.*
- class [Gnss\\_Satellite](#)  
*This class represents a GNSS satellite.*
- class [Gnss\\_Signal](#)  
*This class represents a GNSS signal.*
- class [Gnss\\_Synchro](#)  
*This is the class that contains the information that is shared by the processing blocks.*
- class [Gps\\_Acq\\_Assist](#)  
*This class is a storage for the GPS GSM RRLL acquisition assistance data as described in Digital cellular telecommunications system (Phase 2+); Location Services (LCS); Mobile Station (MS) - Serving Mobile Location Centre (SMLC) Radio Resource LCS Protocol (RRLP) (3GPP TS 44.031 version 5.12.0 Release 5)*
- class [Gps\\_Almanac](#)  
*This class is a storage for the GPS SV ALMANAC data as described in IS-GPS-200M.*
- class [Gps\\_CNAV\\_Ephemeris](#)  
*This is a storage class for the GPS CNAV ephemeris data as described in IS-GPS-200M.*
- class [Gps\\_CNAV\\_Iono](#)  
*This class is a storage for the GPS IONOSPHERIC data as described in IS-GPS-200M.*
- class [Gps\\_CNAV\\_Navigation\\_Message](#)  
*This class decodes a GPS CNAV Data message as described in IS-GPS-200M.*
- class [Gps\\_CNAV\\_Utc\\_Model](#)  
*This class is a storage for the GPS UTC MODEL data as described in in IS-GPS-200M.*
- class [Gps\\_Ephemeris](#)  
*This class is a storage and orbital model functions for the GPS SV ephemeris data as described in IS-GPS-200M.*
- class [Gps\\_Iono](#)  
*This class is a storage for the GPS IONOSPHERIC data as described in IS-GPS-200M.*
- class [Gps\\_Navigation\\_Message](#)  
*This class decodes a GPS NAV Data message as described in IS-GPS-200M.*

- class [Gps\\_Utc\\_Model](#)

*This class is a storage for the GPS UTC MODEL data as described in IS-GPS-200M.*

- class [DSM\\_nma\\_header](#)
- class [DSM\\_dsm\\_header](#)
- class [MACK\\_header](#)
- class [MACK\\_tag\\_info](#)
- class [MACK\\_tag\\_and\\_info](#)
- class [DSM\\_PKR\\_message](#)
- class [DSM\\_KROOT\\_message](#)
- class [MACK\\_message](#)
- class [OSNMA\\_NavData](#)
- class [OSNMA\\_data](#)

*This class handles OSNMA data See [https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo\\_OSNMA\\_User\\_ICD\\_for\\_Test\\_Phase\\_v1.0.pdf](https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OSNMA_User_ICD_for_Test_Phase_v1.0.pdf).*

- class [Tag](#)
- class [OSNMA\\_DSM\\_Reader](#)
- class [ReedSolomon](#)

*Class implementing a Reed-Solomon encoder and decoder RS(255,K,d) where k=255-nroots is the information vector length and d=nroots+1 is the minimum Hamming distance, with symbols of 8 bits. It allows shortened RS codes.*

- class [Sbas\\_Ephemeris](#)

*This class stores SBAS SV ephemeris data.*

## Macros

- `#define DISPLAY\_COLORS 1`
- `#define GLONASS\_GNAV\_PREAMBLE`

## Functions

- `const std::vector< std::pair< int32_t, int32_t > > D1\_PRE ({1, 11})`
- `const std::vector< std::pair< int32_t, int32_t > > D1\_FRAID ({16, 3})`
- `const std::vector< std::pair< int32_t, int32_t > > D1\_SOW ({19, 8}, {31, 12})`
- `const std::vector< std::pair< int32_t, int32_t > > D1\_PNUM ({44, 7})`
- `const std::vector< std::pair< int32_t, int32_t > > D1\_SAT\_H1 ({43, 1})`
- `const std::vector< std::pair< int32_t, int32_t > > D1\_AODC ({44, 5})`
- `const std::vector< std::pair< int32_t, int32_t > > D1\_URAI ({49, 4})`
- `const std::vector< std::pair< int32_t, int32_t > > D1\_WN ({61, 13})`
- `const std::vector< std::pair< int32_t, int32_t > > D1\_TOC ({74, 9}, {91, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > D1\_TGD1 ({99, 10})`
- `const std::vector< std::pair< int32_t, int32_t > > D1\_TGD2 ({109, 4}, {121, 6})`
- `const std::vector< std::pair< int32_t, int32_t > > D1\_ALPHA0 ({127, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > D1\_ALPHA1 ({135, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > D1\_ALPHA2 ({151, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > D1\_ALPHA3 ({159, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > D1\_BETA0 ({167, 6}, {181, 2})`
- `const std::vector< std::pair< int32_t, int32_t > > D1\_BETA1 ({183, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > D1\_BETA2 ({191, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > D1\_BETA3 ({199, 4}, {211, 4})`
- `const std::vector< std::pair< int32_t, int32_t > > D1\_A2 ({215, 11})`
- `const std::vector< std::pair< int32_t, int32_t > > D1\_A0 ({226, 7}, {241, 17})`
- `const std::vector< std::pair< int32_t, int32_t > > D1\_A1 ({258, 5}, {271, 17})`
- `const std::vector< std::pair< int32_t, int32_t > > D1\_AODE ({288, 5})`
- `const std::vector< std::pair< int32_t, int32_t > > D1\_DELTA\_N ({43, 10}, {61, 6})`
- `const std::vector< std::pair< int32_t, int32_t > > D1\_CUC ({67, 16}, {91, 2})`
- `const std::vector< std::pair< int32_t, int32_t > > D1\_M0 ({93, 20}, {121, 12})`

- `const std::vector< std::pair< int32_t, int32_t > > D1_E {{{133, 10}, {151, 22}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_CUS {{{181, 18}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_CRC {{{199, 4}, {211, 14}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_CRS {{{225, 8}, {241, 10}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_SQRT_A {{{251, 12}, {271, 20}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_TOE_SF2 {{{291, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_TOE_SF3 {{{43, 10}, {61, 5}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_I0 {{{66, 17}, {91, 15}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_CIC {{{106, 7}, {121, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_OMEGA_DOT {{{132, 11}, {151, 13}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_CIS {{{164, 9}, {181, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_IDOT {{{190, 13}, {211, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_OMEGA0 {{{212, 21}, {241, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_OMEGA {{{252, 11}, {271, 21}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_SQRT_A_ALMANAC {{{51, 2}, {61, 22}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_A1_ALMANAC {{{91, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_A0_ALMANAC {{{102, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_OMEGA0_ALMANAC {{{121, 22}, {151, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_E_ALMANAC {{{153, 17}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_DELTA_I {{{170, 3}, {181, 13}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_TOA {{{194, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_OMEGA_DOT_ALMANAC {{{202, 1}, {211, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_OMEGA_ALMANAC {{{227, 6}, {241, 18}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_M0_ALMANAC {{{259, 4}, {271, 20}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA1 {{{51, 2}, {61, 7}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA2 {{{68, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA3 {{{77, 6}, {91, 3}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA4 {{{94, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA5 {{{103, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA6 {{{112, 1}, {121, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA7 {{{129, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA8 {{{138, 5}, {151, 4}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA9 {{{155, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA10 {{{164, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA11 {{{181, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA12 {{{190, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA13 {{{199, 4}, {211, 5}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA14 {{{216, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA15 {{{225, 8}, {241, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA16 {{{242, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA17 {{{251, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA18 {{{260, 3}, {271, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA19 {{{277, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA20 {{{51, 2}, {61, 7}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA21 {{{68, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA22 {{{77, 6}, {91, 3}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA23 {{{94, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA24 {{{103, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA25 {{{112, 1}, {121, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA26 {{{129, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA27 {{{138, 5}, {151, 4}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA28 {{{155, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA29 {{{164, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA30 {{{181, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_WNA {{{190, 8}}}`

- `const std::vector< std::pair< int32_t, int32_t > > D1_TOA2 ({198, 5}, {211, 3})`
- `const std::vector< std::pair< int32_t, int32_t > > D1_A0GPS ({97, 14})`
- `const std::vector< std::pair< int32_t, int32_t > > D1_A1GPS ({111, 2}, {121, 14})`
- `const std::vector< std::pair< int32_t, int32_t > > D1_A0GAL ({135, 8}, {151, 6})`
- `const std::vector< std::pair< int32_t, int32_t > > D1_A1GAL ({157, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > D1_A0GLO ({181, 14})`
- `const std::vector< std::pair< int32_t, int32_t > > D1_A1GLO ({195, 8}, {211, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > D1_DELTA_T_LS ({51, 2}, {61, 6})`
- `const std::vector< std::pair< int32_t, int32_t > > D1_DELTA_T_LSF ({67, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > D1_WN_LSF ({75, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > D1_A0UTC ({91, 22}, {121, 10})`
- `const std::vector< std::pair< int32_t, int32_t > > D1_A1UTC ({131, 12}, {151, 12})`
- `const std::vector< std::pair< int32_t, int32_t > > D1_DN ({163, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_PRE ({1, 11})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_FRAID ({16, 3})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_SOW ({19, 8}, {31, 12})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_PNUM ({43, 4})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_SAT_H1 ({47, 1})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_AODC ({48, 5})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_URAI ({61, 4})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_WN ({65, 13})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_TOC ({78, 5}, {91, 12})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_TGD1 ({103, 10})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_TGD2 ({121, 10})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_ALPHA0 ({47, 6}, {61, 2})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_ALPHA1 ({63, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_ALPHA2 ({71, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_ALPHA3 ({79, 4}, {91, 4})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_BETA0 ({95, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_BETA1 ({103, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_BETA2 ({111, 2}, {121, 6})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_BETA3 ({127, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_A0 ({101, 12}, {121, 12})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_A1_MSB ({133, 4})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_A1_LSB ({47, 6}, {61, 12})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_A1 ({279, 22})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_A2 ({73, 10}, {91, 1})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_AODE ({92, 5})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_DELTA_N ({97, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_CUC_MSB ({121, 14})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_CUC_LSB ({47, 4})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_CUC ({283, 18})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_M0 ({51, 2}, {61, 22}, {91, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_CUS ({99, 14}, {121, 4})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_E_MSB ({125, 10})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_E_LSB ({47, 6}, {61, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_SQRT_A ({77, 6}, {91, 22}, {121, 4})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_CIC_MSB ({125, 10})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_CIC_LSB ({47, 6}, {61, 2})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_CIC ({283, 18})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_CIS ({63, 18})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_TOE ({81, 2}, {91, 15})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_I0_MSB ({106, 7}, {121, 14})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_I0_LSB ({47, 6}, {61, 5})`
- `const std::vector< std::pair< int32_t, int32_t > > D2_I0 ({269, 32})`

- `const std::vector< std::pair< int32_t, int32_t > > D2_CRC {{{66, 17}, {91, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D2_CRS {{{92, 18}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D2_OMEGA_DOT_MSB {{{110, 3}, {121, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D2_OMEGA_DOT_LSB {{{47, 5}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D2_OMEGA_DOT {{{277, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D2_OMEGA0 {{{52, 1}, {61, 22}, {91, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D2_OMEGA_MSB {{{100, 13}, {121, 14}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D2_OMEGA_LSB {{{47, 5}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D2_OMEGA {{{269, 32}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D2_IDOT {{{52, 1}, {61, 13}}}`
- `const std::pair< int32_t, int32_t > GALILEO_HAS_STATUS {{1, 2}}`
- `const std::pair< int32_t, int32_t > GALILEO_HAS_RESERVED {{3, 2}}`
- `const std::pair< int32_t, int32_t > GALILEO_HAS_MESSAGE_TYPE {{5, 2}}`
- `const std::pair< int32_t, int32_t > GALILEO_HAS_MESSAGE_ID {{7, 5}}`
- `const std::pair< int32_t, int32_t > GALILEO_HAS_MESSAGE_SIZE {{12, 5}}`
- `const std::pair< int32_t, int32_t > GALILEO_HAS_MESSAGE_PAGE_ID {{17, 8}}`
- `const std::pair< int32_t, int32_t > GALILEO_MT1_HEADER_TOH {{1, 12}}`
- `const std::pair< int32_t, int32_t > GALILEO_MT1_HEADER_MASK_FLAG {{13, 1}}`
- `const std::pair< int32_t, int32_t > GALILEO_MT1_HEADER_ORBIT_CORRECTION_FLAG {{14, 1}}`
- `const std::pair< int32_t, int32_t > GALILEO_MT1_HEADER_CLOCK_FULLSET_FLAG {{15, 1}}`
- `const std::pair< int32_t, int32_t > GALILEO_MT1_HEADER_CLOCK_SUBSET_FLAG {{16, 1}}`
- `const std::pair< int32_t, int32_t > GALILEO_MT1_HEADER_CODE_BIAS_FLAG {{17, 1}}`
- `const std::pair< int32_t, int32_t > GALILEO_MT1_HEADER_PHASE_BIAS_FLAG {{18, 1}}`
- `const std::pair< int32_t, int32_t > GALILEO_MT1_HEADER_RESERVED {{19, 4}}`
- `const std::pair< int32_t, int32_t > GALILEO_MT1_HEADER_MASK_ID {{23, 5}}`
- `const std::pair< int32_t, int32_t > GALILEO_MT1_HEADER_IOD_SET_ID {{28, 5}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_PAGE_TYPE_BIT {{{1, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_SV_ID_PRN_1_BIT {{{7, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_IO_DNAV_1_BIT {{{13, 10}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_T0C_1_BIT {{{23, 14}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_AF0_1_BIT {{{37, 31}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_AF1_1_BIT {{{68, 21}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_AF2_1_BIT {{{89, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_SISA_1_BIT {{{95, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_AI0_1_BIT {{{103, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_AI1_1_BIT {{{114, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_AI2_1_BIT {{{125, 14}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_REGION1_1_BIT {{{139, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_REGION2_1_BIT {{{140, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_REGION3_1_BIT {{{141, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_REGION4_1_BIT {{{142, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_REGION5_1_BIT {{{143, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_BGD_1_BIT {{{144, 10}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_E5AHS_1_BIT {{{154, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_WN_1_BIT {{{156, 12}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_TOW_1_BIT {{{168, 20}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_E5ADVS_1_BIT {{{188, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_IO_DNAV_2_BIT {{{7, 10}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_M0_2_BIT {{{17, 32}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_OMEGADOT_2_BIT {{{49, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_E_2_BIT {{{73, 32}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_A12_2_BIT {{{105, 32}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_OMEGA0_2_BIT {{{137, 32}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_IDOT_2_BIT {{{169, 14}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_WN_2_BIT {{{183, 12}}}`

- `const std::vector< std::pair< int32_t, int32_t > > FNAV_TOW_2_BIT ({195, 20})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_IO_DNAV_3_BIT ({7, 10})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_IO_3_BIT ({17, 32})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_W_3_BIT ({49, 32})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_DELTAN_3_BIT ({81, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_CUC_3_BIT ({97, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_CUS_3_BIT ({113, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_CRC_3_BIT ({129, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_CRS_3_BIT ({145, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_T0E_3_BIT ({161, 14})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_WN_3_BIT ({175, 12})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_TOW_3_BIT ({187, 20})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_IO_DNAV_4_BIT ({7, 10})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_CIC_4_BIT ({17, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_CIS_4_BIT ({33, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_A0_4_BIT ({49, 32})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_A1_4_BIT ({81, 24})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_DELTATLS_4_BIT ({105, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_T0T_4_BIT ({113, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_W_NOT_4_BIT ({121, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_W_NLSF_4_BIT ({129, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_DN_4_BIT ({137, 3})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_DELTATLSF_4_BIT ({140, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_T0G_4_BIT ({148, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_A0G_4_BIT ({156, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_A1G_4_BIT ({172, 12})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_W_N0G_4_BIT ({184, 6})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_TOW_4_BIT ({190, 20})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_IO_DA_5_BIT ({7, 4})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_W_NA_5_BIT ({11, 2})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_T0A_5_BIT ({13, 10})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_SVI_D1_5_BIT ({23, 6})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_DELTA12_1_5_BIT ({29, 13})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_E_1_5_BIT ({42, 11})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_W_1_5_BIT ({53, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_DELTAI_1_5_BIT ({69, 11})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_OMEGA0_1_5_BIT ({80, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_OMEGADOT_1_5_BIT ({96, 11})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_M0_1_5_BIT ({107, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_AF0_1_5_BIT ({123, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_AF1_1_5_BIT ({139, 13})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_E5AHS_1_5_BIT ({152, 2})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_SVI_D2_5_BIT ({154, 6})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_DELTA12_2_5_BIT ({160, 13})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_E_2_5_BIT ({173, 11})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_W_2_5_BIT ({184, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_DELTAI_2_5_BIT ({200, 11})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_IO_DA_6_BIT ({7, 4})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_OMEGADOT_2_6_BIT ({23, 11})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_M0_2_6_BIT ({34, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_AF0_2_6_BIT ({50, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_AF1_2_6_BIT ({66, 13})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_E5AHS_2_6_BIT ({79, 2})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_SVI_D3_6_BIT ({81, 6})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_DELTA12_3_6_BIT ({87, 13})`



- `const std::vector< std::pair< int32_t, int32_t > > FNAV_E_3_6_BIT {{{100, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_W_3_6_BIT {{{111, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_DELTAI_3_6_BIT {{{127, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_OMEGA0_3_6_BIT {{{138, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_OMEGADOT_3_6_BIT {{{154, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_M0_3_6_BIT {{{165, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_AF0_3_6_BIT {{{181, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_AF1_3_6_BIT {{{197, 13}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_E5AHS_3_6_BIT {{{210, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > TYPE {{{1, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > PAGE_TYPE_BIT {{{1, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > IOD_NAV_1_BIT {{{7, 10}}}`
- `const std::vector< std::pair< int32_t, int32_t > > T0_E_1_BIT {{{17, 14}}}`
- `const std::vector< std::pair< int32_t, int32_t > > M0_1_BIT {{{31, 32}}}`
- `const std::vector< std::pair< int32_t, int32_t > > E_1_BIT {{{63, 32}}}`
- `const std::vector< std::pair< int32_t, int32_t > > A_1_BIT {{{95, 32}}}`
- `const std::vector< std::pair< int32_t, int32_t > > IOD_NAV_2_BIT {{{7, 10}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA_0_2_BIT {{{17, 32}}}`
- `const std::vector< std::pair< int32_t, int32_t > > I_0_2_BIT {{{49, 32}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA_2_BIT {{{81, 32}}}`
- `const std::vector< std::pair< int32_t, int32_t > > I_DOT_2_BIT {{{113, 14}}}`
- `const std::vector< std::pair< int32_t, int32_t > > IOD_NAV_3_BIT {{{7, 10}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA_DOT_3_BIT {{{17, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DELTA_N_3_BIT {{{41, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > C_UC_3_BIT {{{57, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > C_US_3_BIT {{{73, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > C_RC_3_BIT {{{89, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > C_RS_3_BIT {{{105, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > SISA_3_BIT {{{121, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > IOD_NAV_4_BIT {{{7, 10}}}`
- `const std::vector< std::pair< int32_t, int32_t > > SV_ID_PRN_4_BIT {{{17, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > C_IC_4_BIT {{{23, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > C_IS_4_BIT {{{39, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > T0C_4_BIT {{{55, 14}}}`
- `const std::vector< std::pair< int32_t, int32_t > > AF0_4_BIT {{{69, 31}}}`
- `const std::vector< std::pair< int32_t, int32_t > > AF1_4_BIT {{{100, 21}}}`
- `const std::vector< std::pair< int32_t, int32_t > > AF2_4_BIT {{{121, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > SPARE_4_BIT {{{127, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > AI0_5_BIT {{{7, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > AI1_5_BIT {{{18, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > AI2_5_BIT {{{29, 14}}}`
- `const std::vector< std::pair< int32_t, int32_t > > REGION1_5_BIT {{{43, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > REGION2_5_BIT {{{44, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > REGION3_5_BIT {{{45, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > REGION4_5_BIT {{{46, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > REGION5_5_BIT {{{47, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > BGD_E1_E5A_5_BIT {{{48, 10}}}`
- `const std::vector< std::pair< int32_t, int32_t > > BGD_E1_E5B_5_BIT {{{58, 10}}}`
- `const std::vector< std::pair< int32_t, int32_t > > E5B_HS_5_BIT {{{68, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > E1_B_HS_5_BIT {{{70, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > E5B_DVS_5_BIT {{{72, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > E1_B_DVS_5_BIT {{{73, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > WN_5_BIT {{{74, 12}}}`
- `const std::vector< std::pair< int32_t, int32_t > > TOW_5_BIT {{{86, 20}}}`
- `const std::vector< std::pair< int32_t, int32_t > > SPARE_5_BIT {{{106, 23}}}`

- `const std::vector< std::pair< int32_t, int32_t > > A0_6_BIT {{{7, 32}}}`
- `const std::vector< std::pair< int32_t, int32_t > > A1_6_BIT {{{39, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DELTA_T_LS_6_BIT {{{63, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > T0T_6_BIT {{{71, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > W_NOT_6_BIT {{{79, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > WN_LSF_6_BIT {{{87, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DN_6_BIT {{{95, 3}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DELTA_T_LSF_6_BIT {{{98, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > TOW_6_BIT {{{106, 20}}}`
- `const std::vector< std::pair< int32_t, int32_t > > IOD_A_7_BIT {{{7, 4}}}`
- `const std::vector< std::pair< int32_t, int32_t > > WN_A_7_BIT {{{11, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > T0A_7_BIT {{{13, 10}}}`
- `const std::vector< std::pair< int32_t, int32_t > > SVI_D1_7_BIT {{{23, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DELTA_A_7_BIT {{{29, 13}}}`
- `const std::vector< std::pair< int32_t, int32_t > > E_7_BIT {{{42, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA_7_BIT {{{53, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DELTA_I_7_BIT {{{69, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA0_7_BIT {{{80, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA_DOT_7_BIT {{{96, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > M0_7_BIT {{{107, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > IOD_A_8_BIT {{{7, 4}}}`
- `const std::vector< std::pair< int32_t, int32_t > > AF0_8_BIT {{{11, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > AF1_8_BIT {{{27, 13}}}`
- `const std::vector< std::pair< int32_t, int32_t > > E5B_HS_8_BIT {{{40, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > E1_B_HS_8_BIT {{{42, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > SVI_D2_8_BIT {{{44, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DELTA_A_8_BIT {{{50, 13}}}`
- `const std::vector< std::pair< int32_t, int32_t > > E_8_BIT {{{63, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA_8_BIT {{{74, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DELTA_I_8_BIT {{{90, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA0_8_BIT {{{101, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA_DOT_8_BIT {{{117, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > IOD_A_9_BIT {{{7, 4}}}`
- `const std::vector< std::pair< int32_t, int32_t > > WN_A_9_BIT {{{11, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > T0A_9_BIT {{{13, 10}}}`
- `const std::vector< std::pair< int32_t, int32_t > > M0_9_BIT {{{23, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > AF0_9_BIT {{{39, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > AF1_9_BIT {{{55, 13}}}`
- `const std::vector< std::pair< int32_t, int32_t > > E5B_HS_9_BIT {{{68, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > E1_B_HS_9_BIT {{{70, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > SVI_D3_9_BIT {{{72, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DELTA_A_9_BIT {{{78, 13}}}`
- `const std::vector< std::pair< int32_t, int32_t > > E_9_BIT {{{91, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA_9_BIT {{{102, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DELTA_I_9_BIT {{{118, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > IOD_A_10_BIT {{{7, 4}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA0_10_BIT {{{11, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA_DOT_10_BIT {{{27, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > M0_10_BIT {{{38, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > AF0_10_BIT {{{54, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > AF1_10_BIT {{{70, 13}}}`
- `const std::vector< std::pair< int32_t, int32_t > > E5B_HS_10_BIT {{{83, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > E1_B_HS_10_BIT {{{85, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > A_0_G_10_BIT {{{87, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > A_1_G_10_BIT {{{103, 12}}}`

```

• const std::vector< std::pair< int32_t, int32_t > > T_0_G_10_BIT ({{115, 8}})
• const std::vector< std::pair< int32_t, int32_t > > WN_0_G_10_BIT ({{123, 6}})
• const std::vector< std::pair< int32_t, int32_t > > CED_DeltaAred_BIT ({{7, 5}})
• const std::vector< std::pair< int32_t, int32_t > > CED_exred_BIT ({{12, 13}})
• const std::vector< std::pair< int32_t, int32_t > > CED_eyred_BIT ({{25, 13}})
• const std::vector< std::pair< int32_t, int32_t > > CED_DeltaI0red_BIT ({{38, 17}})
• const std::vector< std::pair< int32_t, int32_t > > CED_Omega0red_BIT ({{55, 23}})
• const std::vector< std::pair< int32_t, int32_t > > CED_lambda0red_BIT ({{78, 23}})
• const std::vector< std::pair< int32_t, int32_t > > CED_af0red_BIT ({{101, 22}})
• const std::vector< std::pair< int32_t, int32_t > > CED_af1red_BIT ({{123, 6}})
• const std::vector< std::pair< int32_t, int32_t > > RS_IODNAV_LSBS ({{15, 2}})
• const std::vector< std::pair< int32_t, int32_t > > ISM_CONSTELLATION_ID_BIT ({{7, 3}})
• const std::vector< std::pair< int32_t, int32_t > > ISM_SERVICE_LEVEL_ID_BIT ({{10, 3}})
• const std::vector< std::pair< int32_t, int32_t > > ISM_WN_BIT ({{13, 12}})
• const std::vector< std::pair< int32_t, int32_t > > ISM_T0_BIT ({{25, 9}})
• const std::vector< std::pair< int32_t, int32_t > > ISM_MASK_MSB_BIT ({{34, 1}})
• const std::vector< std::pair< int32_t, int32_t > > ISM_MASK_BIT ({{35, 32}})
• const std::vector< std::pair< int32_t, int32_t > > ISM_PCONST_BIT ({{67, 4}})
• const std::vector< std::pair< int32_t, int32_t > > ISM_PSAT_BIT ({{71, 4}})
• const std::vector< std::pair< int32_t, int32_t > > ISM_URA_BIT ({{75, 4}})
• const std::vector< std::pair< int32_t, int32_t > > ISM_URE_BIT ({{79, 4}})
• const std::vector< std::pair< int32_t, int32_t > > ISM_BNOM_BIT ({{83, 4}})
• const std::vector< std::pair< int32_t, int32_t > > ISM_TVALIDITY_BIT ({{87, 4}})
• const std::vector< std::pair< int32_t, int32_t > > ISM_CRC_BIT ({{97, 32}})
• const std::vector< std::pair< int32_t, int32_t > > TIME_0_BIT ({{7, 2}})
• const std::vector< std::pair< int32_t, int32_t > > WN_0_BIT ({{97, 12}})
• const std::vector< std::pair< int32_t, int32_t > > TOW_0_BIT ({{109, 20}})
• const std::string PEMFILE_DEFAULT (".OSNMA_PublicKey.pem")
• const std::string CRFILE_DEFAULT (".OSNMA_PublicKey_20240115100000_newPKID_1.crt")
• const std::string MERKLEFILE_DEFAULT (".OSNMA_MerkleTree_20240115100000_newPKID_1.xml")
• const std::string KROOTFILE_DEFAULT (".OSNMA_DSM_KROOT_NMAHeader.bin")
• const std::vector< std::pair< int32_t, int32_t > > STRING_ID ({{2, 4}})
• const std::vector< std::pair< int32_t, int32_t > > KX ({{78, 8}})
• const std::vector< std::pair< int32_t, int32_t > > P1 ({{8, 2}})
• const std::vector< std::pair< int32_t, int32_t > > T_K_HR ({{10, 5}})
• const std::vector< std::pair< int32_t, int32_t > > T_K_MIN ({{15, 6}})
• const std::vector< std::pair< int32_t, int32_t > > T_K_SEC ({{21, 1}})
• const std::vector< std::pair< int32_t, int32_t > > X_N_DOT ({{22, 24}})
• const std::vector< std::pair< int32_t, int32_t > > X_N_DOT_DOT ({{46, 5}})
• const std::vector< std::pair< int32_t, int32_t > > X_N ({{51, 27}})
• const std::vector< std::pair< int32_t, int32_t > > B_N ({{6, 3}})
• const std::vector< std::pair< int32_t, int32_t > > P2 ({{9, 1}})
• const std::vector< std::pair< int32_t, int32_t > > T_B ({{10, 7}})
• const std::vector< std::pair< int32_t, int32_t > > Y_N_DOT ({{22, 24}})
• const std::vector< std::pair< int32_t, int32_t > > Y_N_DOT_DOT ({{46, 5}})
• const std::vector< std::pair< int32_t, int32_t > > Y_N ({{51, 27}})
• const std::vector< std::pair< int32_t, int32_t > > P3 ({{6, 1}})
• const std::vector< std::pair< int32_t, int32_t > > GAMMA_N ({{7, 11}})
• const std::vector< std::pair< int32_t, int32_t > > P ({{19, 2}})
• const std::vector< std::pair< int32_t, int32_t > > EPH_L_N ({{21, 1}})
• const std::vector< std::pair< int32_t, int32_t > > Z_N_DOT ({{22, 24}})
• const std::vector< std::pair< int32_t, int32_t > > Z_N_DOT_DOT ({{46, 5}})
• const std::vector< std::pair< int32_t, int32_t > > Z_N ({{51, 27}})
• const std::vector< std::pair< int32_t, int32_t > > TAU_N ({{6, 22}})
• const std::vector< std::pair< int32_t, int32_t > > DELTA_TAU_N ({{28, 5}})

```

- `const std::vector< std::pair< int32_t, int32_t > > E_N {{{33, 5}}}`
- `const std::vector< std::pair< int32_t, int32_t > > P4 {{{52, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > F_T {{{53, 4}}}`
- `const std::vector< std::pair< int32_t, int32_t > > N_T {{{60, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > N {{{71, 5}}}`
- `const std::vector< std::pair< int32_t, int32_t > > M {{{76, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DAY_NUMBER_A {{{6, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > TAU_C {{{17, 32}}}`
- `const std::vector< std::pair< int32_t, int32_t > > N_4 {{{50, 5}}}`
- `const std::vector< std::pair< int32_t, int32_t > > TAU_GPS {{{55, 22}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALM_L_N {{{77, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > C_N {{{6, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > M_N_A {{{7, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > N_A {{{9, 5}}}`
- `const std::vector< std::pair< int32_t, int32_t > > TAU_N_A {{{14, 10}}}`
- `const std::vector< std::pair< int32_t, int32_t > > LAMBDA_N_A {{{24, 21}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DELTA_I_N_A {{{45, 18}}}`
- `const std::vector< std::pair< int32_t, int32_t > > EPSILON_N_A {{{63, 15}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA_N_A {{{6, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > T_LAMBDA_N_A {{{22, 21}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DELTA_T_N_A {{{43, 22}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DELTA_T_DOT_N_A {{{65, 7}}}`
- `const std::vector< std::pair< int32_t, int32_t > > H_N_A {{{72, 5}}}`
- `const std::vector< std::pair< int32_t, int32_t > > B1 {{{6, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > B2 {{{17, 10}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_PRN {{{9, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_MSG_TYPE {{{15, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_TOW {{{21, 17}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_ALERT_FLAG {{{38, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_WN {{{39, 13}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_HEALTH {{{52, 3}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_TOP1 {{{55, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_URA {{{66, 5}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_TOE1 {{{71, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_DELTA_A {{{82, 26}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_A_DOT {{{108, 25}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_DELTA_N0 {{{133, 17}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_DELTA_N0_DOT {{{150, 23}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_M0 {{{173, 33}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_E_ECCENTRICITY {{{206, 33}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_OMEGA {{{239, 33}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_INTEGRITY_FLAG {{{272, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_L2_PHASING_FLAG {{{273, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_TOE2 {{{39, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_OMEGA0 {{{50, 33}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_I0 {{{83, 33}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_DELTA_OMEGA_DOT {{{116, 17}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_I0_DOT {{{133, 15}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_CIS {{{148, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_CIC {{{164, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_CRS {{{180, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_CRC {{{204, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_CUS {{{228, 21}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_CUC {{{249, 21}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_TOP2 {{{39, 11}}}`

- `const std::vector< std::pair< int32_t, int32_t > > CNAV_URA_NED0` ({50, 5})
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_URA_NED1` ({55, 3})
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_URA_NED2` ({58, 3})
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_TOC` ({61, 11})
- `const std::vector< std::pair< int, int > > CNAV_AF0` ({72, 26})
- `const std::vector< std::pair< int, int > > CNAV_AF1` ({98, 20})
- `const std::vector< std::pair< int, int > > CNAV_AF2` ({118, 10})
- `const std::vector< std::pair< int, int > > CNAV_TGD` ({128, 13})
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_ISCL1` ({141, 13})
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_ISCL2` ({154, 13})
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_ISCL5I` ({167, 13})
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_ISCL5Q` ({180, 13})
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_ALPHA0` ({193, 8})
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_ALPHA1` ({201, 8})
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_ALPHA2` ({209, 8})
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_ALPHA3` ({217, 8})
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_BETA0` ({225, 8})
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_BETA1` ({233, 8})
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_BETA2` ({241, 8})
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_BETA3` ({249, 8})
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_WNOP` ({257, 8})
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_A0` ({128, 16})
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_A1` ({144, 13})
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_A2` ({157, 7})
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_DELTA_TLS` ({164, 8})
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_TOT` ({172, 16})
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_WN_OT` ({188, 13})
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_WN_LSF` ({201, 13})
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_DN` ({214, 4})
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_DELTA_TLSF` ({218, 8})
- `const std::vector< std::pair< int32_t, int32_t > > TOW` ({31, 17})
- `const std::vector< std::pair< int32_t, int32_t > > INTEGRITY_STATUS_FLAG` ({23, 1})
- `const std::vector< std::pair< int32_t, int32_t > > ALERT_FLAG` ({48, 1})
- `const std::vector< std::pair< int32_t, int32_t > > ANTI_SPOOFING_FLAG` ({49, 1})
- `const std::vector< std::pair< int32_t, int32_t > > SUBFRAME_ID` ({50, 3})
- `const std::vector< std::pair< int32_t, int32_t > > GPS_WEEK` ({61, 10})
- `const std::vector< std::pair< int32_t, int32_t > > CA_OR_P_ON_L2` ({71, 2})
- `const std::vector< std::pair< int32_t, int32_t > > SV_ACCURACY` ({73, 4})
- `const std::vector< std::pair< int32_t, int32_t > > SV_HEALTH` ({77, 6})
- `const std::vector< std::pair< int32_t, int32_t > > L2_P_DATA_FLAG` ({91, 1})
- `const std::vector< std::pair< int32_t, int32_t > > T_GD` ({197, 8})
- `const std::vector< std::pair< int32_t, int32_t > > IODC` ({83, 2}, {211, 8})
- `const std::vector< std::pair< int32_t, int32_t > > T_OC` ({219, 16})
- `const std::vector< std::pair< int32_t, int32_t > > A_F2` ({241, 8})
- `const std::vector< std::pair< int32_t, int32_t > > A_F1` ({249, 16})
- `const std::vector< std::pair< int32_t, int32_t > > A_F0` ({271, 22})
- `const std::vector< std::pair< int32_t, int32_t > > IODE_SF2` ({61, 8})
- `const std::vector< std::pair< int32_t, int32_t > > C_RS` ({69, 16})
- `const std::vector< std::pair< int32_t, int32_t > > DELTA_N` ({91, 16})
- `const std::vector< std::pair< int32_t, int32_t > > M_0` ({107, 8}, {121, 24})
- `const std::vector< std::pair< int32_t, int32_t > > C_UC` ({151, 16})
- `const std::vector< std::pair< int32_t, int32_t > > ECCENTRICITY` ({167, 8}, {181, 24})
- `const std::vector< std::pair< int32_t, int32_t > > C_US` ({211, 16})
- `const std::vector< std::pair< int32_t, int32_t > > SQRT_A` ({227, 8}, {241, 24})
- `const std::vector< std::pair< int32_t, int32_t > > T_OE` ({271, 16})

- `const std::vector< std::pair< int32_t, int32_t > > FIT_INTERVAL_FLAG {{{271, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > AODO {{{272, 5}}}`
- `const std::vector< std::pair< int32_t, int32_t > > C_IC {{{61, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA_0 {{{77, 8}, {91, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > C_IS {{{121, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > I_0 {{{137, 8}, {151, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > C_RC {{{181, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA {{{197, 8}, {211, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA_DOT {{{241, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > IODE_SF3 {{{271, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > I_DOT {{{279, 14}}}`
- `const std::vector< std::pair< int32_t, int32_t > > SV_DATA_ID {{{61, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > SV_PAGE {{{63, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALPHA_0 {{{69, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALPHA_1 {{{77, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALPHA_2 {{{91, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALPHA_3 {{{99, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > BETA_0 {{{107, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > BETA_1 {{{121, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > BETA_2 {{{129, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > BETA_3 {{{137, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > A_1 {{{151, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > A_0 {{{181, 24}, {211, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > T_OT {{{219, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > WN_T {{{227, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DELTAT_LS {{{241, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > WN_LSF {{{249, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DN {{{257, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DELTAT_LSF {{{271, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV25 {{{229, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV26 {{{241, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV27 {{{247, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV28 {{{253, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV29 {{{259, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV30 {{{271, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV31 {{{277, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV32 {{{283, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > T_OA {{{69, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > WN_A {{{77, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV1 {{{91, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV2 {{{97, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV3 {{{103, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV4 {{{109, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV5 {{{121, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV6 {{{127, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV7 {{{133, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV8 {{{139, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV9 {{{151, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV10 {{{157, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV11 {{{163, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV12 {{{169, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV13 {{{181, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV14 {{{187, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV15 {{{193, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV16 {{{199, 6}}}`

- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV17 {{{211, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV18 {{{217, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV19 {{{223, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV20 {{{229, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV21 {{{241, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV22 {{{247, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV23 {{{253, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV24 {{{259, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALM_ECC {{{69, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALM_TOA {{{91, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALM_DELTAT {{{99, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALM_OMEGADOT {{{121, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALM_SVHEALTH {{{137, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALM_SQUAREA {{{151, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALM_OMEGAZERO {{{181, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALM_OMEGA {{{211, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALM_MZERO {{{241, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALM_AFO {{{271, 8}, {290, 3}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALM_AF1 {{{279, 11}}}`

### Variables

- `constexpr double BEIDOU_B1I_FREQ_HZ = FREQ1_BDS`  
*B1I [Hz].*
- `constexpr double BEIDOU_B1I_CODE_RATE_CPS = 2.046e6`  
*Beidou B1I code rate [chips/s].*
- `constexpr double BEIDOU_B1I_CODE_LENGTH_CHIPS = 2046.0`  
*Beidou B1I code length [chips].*
- `constexpr double BEIDOU_B1I_CODE_PERIOD_S = 0.001`  
*Beidou B1I code period [seconds].*
- `constexpr double BEIDOU_B1I_PREAMBLE_DURATION_S = 0.220`
- `constexpr uint32_t BEIDOU_B1I_CODE_PERIOD_MS = 1`  
*Beidou B1I code period [ms].*
- `constexpr uint32_t BEIDOU_B1I_PREAMBLE_LENGTH_BITS = 11`
- `constexpr uint32_t BEIDOU_B1I_PREAMBLE_LENGTH_SYMBOLS = 220`
- `constexpr int32_t BEIDOU_B1I_SECONDARY_CODE_LENGTH = 20`
- `constexpr int32_t BEIDOU_B1I_GEO_PREAMBLE_LENGTH_SYMBOLS = 22`
- `constexpr int32_t BEIDOU_B1I_PREAMBLE_DURATION_MS = 220`
- `constexpr int32_t BEIDOU_B1I_TELEMETRY_RATE_BITS_SECOND = 50`
- `constexpr int32_t BEIDOU_B1I_TELEMETRY_SYMBOLS_PER_BIT = 20`
- `constexpr int32_t BEIDOU_B1I_GEO_TELEMETRY_SYMBOLS_PER_BIT = 2`
- `constexpr int32_t BEIDOU_B1I_TELEMETRY_SYMBOL_PERIOD_MS = static_cast<int32_t>(static_cast<uint32_t>(BEIDOU_B1I_TELEMETRY_SYMBOLS_PER_BIT) * BEIDOU_B1I_CODE_PERIOD_MS)`
- `constexpr int32_t BEIDOU_B1I_TELEMETRY_RATE_SYMBOLS_SECOND = BEIDOU_B1I_TELEMETRY_RATE_BITS_SECOND * BEIDOU_B1I_TELEMETRY_SYMBOLS_PER_BIT`
- `constexpr char BEIDOU_B1I_SECONDARY_CODE_STR [21] = "00000100110101001110"`
- `constexpr char BEIDOU_B1I_GEO_PREAMBLE_SYMBOLS_STR [23] = "1111110000001100001100"`
- `constexpr char BEIDOU_B1I_D2_SECONDARY_CODE_STR [3] = "00"`
- `constexpr double BEIDOU_B3I_FREQ_HZ = FREQ3_BDS`  
*BeiDou B3I [Hz].*
- `constexpr double BEIDOU_B3I_CODE_RATE_CPS = 10.23e6`  
*BeiDou B3I code rate [chips/s].*
- `constexpr double BEIDOU_B3I_CODE_LENGTH_CHIPS = 10230.0`  
*BeiDou B3I code length [chips].*

- constexpr double `BEIDOU_B3I_CODE_PERIOD_S` = 0.001  
*BeiDou B3I code period [seconds].*
- constexpr double `BEIDOU_B3I_PREAMBLE_DURATION_S` = 0.220
- constexpr uint32\_t `BEIDOU_B3I_CODE_PERIOD_MS` = 1  
*BeiDou B3I code period [ms].*
- constexpr uint32\_t `BEIDOU_B3I_PREAMBLE_LENGTH_BITS` = 11
- constexpr uint32\_t `BEIDOU_B3I_PREAMBLE_LENGTH_SYMBOLS` = 220
- constexpr int32\_t `BEIDOU_B3I_SECONDARY_CODE_LENGTH` = 20
- constexpr int32\_t `BEIDOU_B3I_GEO_PREAMBLE_LENGTH_SYMBOLS` = 22
- constexpr int32\_t `BEIDOU_B3I_PREAMBLE_DURATION_MS` = 220
- constexpr int32\_t `BEIDOU_B3I_TELEMETRY_RATE_BITS_SECOND` = 50  
*D1 NAV message bit rate [bits/s].*
- constexpr int32\_t `BEIDOU_B3I_TELEMETRY_SYMBOLS_PER_BIT` = 20
- constexpr int32\_t `BEIDOU_B3I_GEO_TELEMETRY_SYMBOLS_PER_BIT` = 2
- constexpr int32\_t `BEIDOU_B3I_TELEMETRY_SYMBOL_PERIOD_MS` = static\_cast<int32\_t>(static\_cast<uint32\_t>(BEIDOU\_B3I\_TELEMETRY\_SYMBOLS\_PER\_BIT) \* BEIDOU\_B3I\_CODE\_PERIOD\_MS)
- constexpr int32\_t `BEIDOU_B3I_TELEMETRY_RATE_SYMBOLS_SECOND` = BEIDOU\_B3I\_TELEMETRY\_RATE\_BITS\_SECOND \* BEIDOU\_B3I\_TELEMETRY\_SYMBOLS\_PER\_BIT
- constexpr char `BEIDOU_B3I_SECONDARY_CODE_STR` [21] = "00000100110101001110"
- constexpr char `BEIDOU_B3I_GEO_PREAMBLE_SYMBOLS_STR` [23] = "1111110000001100001100"
- constexpr char `BEIDOU_B3I_D2_SECONDARY_CODE_STR` [3] = "00"
- constexpr double `D1_TOC_LSB` = TWO\_P3
- constexpr double `D1_TGD1_LSB` = 0.1e-9
- constexpr double `D1_TGD2_LSB` = 0.1e-9
- constexpr double `D1_ALPHA0_LSB` = TWO\_N30
- constexpr double `D1_ALPHA1_LSB` = TWO\_N27
- constexpr double `D1_ALPHA2_LSB` = TWO\_N24
- constexpr double `D1_ALPHA3_LSB` = TWO\_N24
- constexpr double `D1_BETA0_LSB` = TWO\_P11
- constexpr double `D1_BETA1_LSB` = TWO\_P14
- constexpr double `D1_BETA2_LSB` = TWO\_P16
- constexpr double `D1_BETA3_LSB` = TWO\_P16
- constexpr double `D1_A2_LSB` = TWO\_N66
- constexpr double `D1_A0_LSB` = TWO\_N33
- constexpr double `D1_A1_LSB` = TWO\_N50
- constexpr double `D1_DELTA_N_LSB` = PI\_TWO\_N43
- constexpr double `D1_CUC_LSB` = TWO\_N31
- constexpr double `D1_M0_LSB` = PI\_TWO\_N31
- constexpr double `D1_E_LSB` = TWO\_N33
- constexpr double `D1_CUS_LSB` = TWO\_N31
- constexpr double `D1_CRC_LSB` = TWO\_N6
- constexpr double `D1_CRS_LSB` = TWO\_N6
- constexpr double `D1_SQRT_A_LSB` = TWO\_N19
- constexpr double `D1_TOE_LSB` = TWO\_P3
- constexpr double `D1_I0_LSB` = PI\_TWO\_N31
- constexpr double `D1_CIC_LSB` = TWO\_N31
- constexpr double `D1_OMEGA_DOT_LSB` = PI\_TWO\_N43
- constexpr double `D1_CIS_LSB` = TWO\_N31
- constexpr double `D1_IDOT_LSB` = PI\_TWO\_N43
- constexpr double `D1_OMEGA0_LSB` = PI\_TWO\_N31
- constexpr double `D1_OMEGA_LSB` = PI\_TWO\_N31
- constexpr double `D1_SQRT_A_ALMANAC_LSB` = TWO\_N11
- constexpr double `D1_A1_ALMANAC_LSB` = TWO\_N38
- constexpr double `D1_A0_ALMANAC_LSB` = TWO\_N20



- constexpr double D1\_OMEGA0\_ALMANAC\_LSB = PI\_TWO\_N23
  - constexpr double D1\_E\_ALMANAC\_LSB = TWO\_N21
  - constexpr double D1\_DELTA\_I\_LSB = PI\_TWO\_N19
  - constexpr double D1\_TOA\_LSB = TWO\_P12
  - constexpr double D1\_OMEGA\_DOT\_ALMANAC\_LSB = PI\_TWO\_N38
  - constexpr double D1\_OMEGA\_ALMANAC\_LSB = PI\_TWO\_N23
  - constexpr double D1\_M0\_ALMANAC\_LSB = PI\_TWO\_N23
  - constexpr double D1\_A0GPS\_LSB = 0.1e-9
  - constexpr double D1\_A1GPS\_LSB = 0.1e-9
  - constexpr double D1\_A0GAL\_LSB = 0.1e-9
  - constexpr double D1\_A1GAL\_LSB = 0.1e-9
  - constexpr double D1\_A0GLO\_LSB = 0.1e-9
  - constexpr double D1\_A1GLO\_LSB = 0.1e-9
  - constexpr double D1\_A0UTC\_LSB = TWO\_N30
  - constexpr double D1\_A1UTC\_LSB = TWO\_N50
  - constexpr int32\_t BEIDOU\_DNAV\_PREAMBLE\_LENGTH\_BITS = 11
  - constexpr int32\_t BEIDOU\_DNAV\_PREAMBLE\_LENGTH\_SYMBOLS = 11
  - constexpr int32\_t BEIDOU\_DNAV\_PREAMBLE\_PERIOD\_SYMBOLS = 300
  - constexpr int32\_t BEIDOU\_DNAV\_SUBFRAME\_DATA\_BITS = 300
- Number of bits per subframe in the NAV message [bits].*
- constexpr int32\_t BEIDOU\_DNAV\_BDT2GPST\_LEAP\_SEC\_OFFSET = 14
  - constexpr int32\_t BEIDOU\_DNAV\_BDT2GPST\_WEEK\_NUM\_OFFSET = 1356
  - constexpr uint32\_t BEIDOU\_DNAV\_SUBFRAME\_SYMBOLS = 300
  - constexpr uint32\_t BEIDOU\_DNAV\_WORDS\_SUBFRAME = 10
  - constexpr uint32\_t BEIDOU\_DNAV\_WORD\_LENGTH\_BITS = 30
  - constexpr char BEIDOU\_DNAV\_PREAMBLE [12] = "11100010010"
  - const std::string TEXT\_RESET = "\033[0m"
  - const std::string TEXT\_BLACK = "\033[30m"
  - const std::string TEXT\_RED = "\033[31m"
  - const std::string TEXT\_GREEN = "\033[32m"
  - const std::string TEXT\_YELLOW = "\033[33m"
  - const std::string TEXT\_BLUE = "\033[34m"
  - const std::string TEXT\_MAGENTA = "\033[35m"
  - const std::string TEXT\_CYAN = "\033[36m"
  - const std::string TEXT\_WHITE = "\033[37m"
  - const std::string TEXT\_BOLD\_BLACK = "\033[1m\033[30m"
  - const std::string TEXT\_BOLD\_RED = "\033[1m\033[31m"
  - const std::string TEXT\_BOLD\_GREEN = "\033[1m\033[32m"
  - const std::string TEXT\_BOLD\_YELLOW = "\033[1m\033[33m"
  - const std::string TEXT\_BOLD\_BLUE = "\033[1m\033[34m"
  - const std::string TEXT\_BOLD\_MAGENTA = "\033[1m\033[35m"
  - const std::string TEXT\_BOLD\_CYAN = "\033[1m\033[36m"
  - const std::string TEXT\_BOLD\_WHITE = "\033[1m\033[37m"
  - constexpr size\_t HAS\_MSG\_NSYS\_LENGTH = 4
  - constexpr size\_t HAS\_MSG\_ID\_MASK\_LENGTH = 4
  - constexpr size\_t HAS\_MSG\_SATELLITE\_MASK\_LENGTH = 40
  - constexpr size\_t HAS\_MSG\_SIGNAL\_MASK\_LENGTH = 16
  - constexpr size\_t HAS\_MSG\_NAV\_MESSAGE\_LENGTH = 3
  - constexpr size\_t HAS\_MSG\_MASK\_RESERVED\_LENGTH = 6
  - constexpr size\_t HAS\_MSG\_VALIDITY\_INDEX\_LENGTH = 4
  - constexpr size\_t HAS\_MSG\_IOD\_GPS\_LENGTH = 8
  - constexpr size\_t HAS\_MSG\_IOD\_GAL\_LENGTH = 10
  - constexpr size\_t HAS\_MSG\_DELTA\_RADIAL\_LENGTH = 13
  - constexpr size\_t HAS\_MSG\_DELTA\_IN\_TRACK\_LENGTH = 12

- constexpr size\_t [HAS\\_MSG\\_DELTA\\_CROSS\\_TRACK\\_LENGTH](#) = 12
- constexpr size\_t [HAS\\_MSG\\_DELTA\\_CLOCK\\_MULTIPLIER\\_LENGTH](#) = 2
- constexpr size\_t [HAS\\_MSG\\_DELTA\\_CLOCK\\_CORRECTION\\_LENGTH](#) = 13
- constexpr size\_t [HAS\\_MSG\\_NSYS\\_SUB\\_LENGTH](#) = 4
- constexpr size\_t [HAS\\_MSG\\_ID\\_CLOCK\\_SUBSET\\_LENGTH](#) = 4
- constexpr size\_t [HAS\\_MSG\\_DELTA\\_CLOCK\\_MULTIPLIER\\_SUBSET\\_LENGTH](#) = 2
- constexpr size\_t [HAS\\_MSG\\_DELTA\\_CLOCK\\_CORRECTION\\_SUBSET\\_LENGTH](#) = 13
- constexpr size\_t [HAS\\_MSG\\_CODE\\_BIAS\\_LENGTH](#) = 11
- constexpr size\_t [HAS\\_MSG\\_PHASE\\_BIAS\\_LENGTH](#) = 11
- constexpr size\_t [HAS\\_MSG\\_PHASE\\_DISCONTINUITY\\_INDICATOR\\_LENGTH](#) = 2
- constexpr uint64\_t [MAX\\_SECONDS\\_REMEMBERING\\_MID](#) = 150
- constexpr int32\_t [GALILEO\\_CNAV\\_SYMBOLS\\_PER\\_PAGE](#) = 1000
- constexpr int32\_t [GALILEO\\_CNAV\\_PREAMBLE\\_PERIOD\\_SYMBOLS](#) = 1000
- constexpr int32\_t [GALILEO\\_CNAV\\_PAGE\\_MS](#) = 1
- constexpr int32\_t [GALILEO\\_CNAV\\_INTERLEAVER\\_ROWS](#) = 8
- constexpr int32\_t [GALILEO\\_CNAV\\_INTERLEAVER\\_COLS](#) = 123
- constexpr int32\_t [GALILEO\\_CNAV\\_TELEMETRY\\_RATE\\_BITS\\_SECOND](#) = 1000
- constexpr int32\_t [GALILEO\\_CNAV\\_HAS\\_PAGE\\_DATA\\_BITS](#) = 448
- constexpr int32\_t [GALILEO\\_CNAV\\_PAGE\\_RESERVED\\_BITS](#) = 14
- constexpr int32\_t [GALILEO\\_CNAV\\_BYTES\\_FOR\\_CRC](#) = 58
- constexpr int32\_t [GALILEO\\_CNAV\\_CRC\\_LENGTH](#) = 24
- constexpr int32\_t [GALILEO\\_CNAV\\_MESSAGE\\_BITS\\_PER\\_PAGE](#) = 424
- constexpr int32\_t [GALILEO\\_CNAV\\_PAGE\\_HEADER\\_BITS](#) = 24
- constexpr int32\_t [GALILEO\\_CNAV\\_PREAMBLE\\_LENGTH\\_BITS](#) = 16
- constexpr int32\_t [GALILEO\\_CNAV\\_MAX\\_NUMBER\\_SYMBOLS\\_ENCODED\\_BLOCK](#) = 255
- constexpr int32\_t [GALILEO\\_CNAV\\_MT1\\_HEADER\\_BITS](#) = 32
- constexpr int32\_t [GALILEO\\_CNAV\\_OCTETS\\_IN\\_SUBPAGE](#) = 53
- constexpr int32\_t [GALILEO\\_CNAV\\_INFORMATION\\_VECTOR\\_LENGTH](#) = 32
- constexpr int32\_t [GALILEO\\_CNAV\\_BITS\\_FOR\\_CRC](#) = [GALILEO\\_CNAV\\_HAS\\_PAGE\\_DATA\\_BITS](#) + [GALILEO\\_CNAV\\_PAGE\\_RESERVED\\_BITS](#)
- constexpr int32\_t [HAS\\_MSG\\_NUMBER\\_MASK\\_IDS](#) = 32
- constexpr int32\_t [HAS\\_MSG\\_NUMBER\\_GNSS\\_IDS](#) = 16
- constexpr int32\_t [HAS\\_MSG\\_NUMBER\\_MESSAGE\\_IDS](#) = 32
- constexpr int32\_t [HAS\\_MSG\\_NUMBER\\_SATELLITE\\_IDS](#) = 40
- constexpr int32\_t [HAS\\_MSG\\_NUMBER\\_SIGNAL\\_MASKS](#) = 16
- constexpr float [HAS\\_MSG\\_DELTA\\_RADIAL\\_SCALE\\_FACTOR](#) = 0.0025
- constexpr float [HAS\\_MSG\\_DELTA\\_IN\\_TRACK\\_SCALE\\_FACTOR](#) = 0.008
- constexpr float [HAS\\_MSG\\_DELTA\\_CROSS\\_TRACK\\_SCALE\\_FACTOR](#) = 0.008
- constexpr float [HAS\\_MSG\\_DELTA\\_CLOCK\\_SCALE\\_FACTOR](#) = 0.0025
- constexpr float [HAS\\_MSG\\_CODE\\_BIAS\\_SCALE\\_FACTOR](#) = 0.02
- constexpr float [HAS\\_MSG\\_PHASE\\_BIAS\\_SCALE\\_FACTOR](#) = 0.01
- constexpr uint16\_t [HAS\\_MSG\\_NUMBER\\_MAX\\_TOH](#) = 3599
- constexpr uint8\_t [HAS\\_MSG\\_GPS\\_SYSTEM](#) = 0
- constexpr uint8\_t [HAS\\_MSG\\_GALILEO\\_SYSTEM](#) = 2
- constexpr uint8\_t [HAS\\_MSG\\_WRONG\\_SYSTEM](#) = 255
- constexpr char [GALILEO\\_CNAV\\_PREAMBLE](#) [17] = "1011011101110000"
- const std::unordered\_map< uint8\_t, std::unordered\_map< uint8\_t, std::string > > [HAS\\_SIGNAL\\_INDEX\\_TABLE](#)
- const std::unordered\_map< uint8\_t, uint16\_t > [HAS\\_VALIDITY\\_INTERVALS](#)
- constexpr double [GALILEO\\_E1\\_FREQ\\_HZ](#) = [FREQ1](#)  
*Galileo E1 carrier frequency [Hz].*
- constexpr double [GALILEO\\_E1\\_CODE\\_CHIP\\_RATE\\_CPS](#) = 1.023e6  
*Galileo E1 code rate [chips/s].*
- constexpr double [GALILEO\\_E1\\_CODE\\_PERIOD\\_S](#) = 0.004  
*Galileo E1 code period [s].*

- constexpr double `GALILEO_E1_SUB_CARRIER_A_RATE_HZ` = 1.023e6  
*Galileo E1 sub-carrier 'a' rate [Hz].*
- constexpr double `GALILEO_E1_SUB_CARRIER_B_RATE_HZ` = 6.138e6  
*Galileo E1 sub-carrier 'b' rate [Hz].*
- constexpr double `GALILEO_E1_B_CODE_LENGTH_CHIPS` = 4092.0  
*Galileo E1-B code length [chips].*
- constexpr double `GALILEO_E1_B_SYMBOL_RATE_BPS` = 250.0  
*Galileo E1-B symbol rate [bits/second].*
- constexpr uint32\_t `GALILEO_E1_CODE_PERIOD_MS` = 4  
*Galileo E1 code period [ms].*
- constexpr int32\_t `GALILEO_E1_B_SAMPLES_PER_SYMBOL` = 1  
*(Galileo\_E1\_CODE\_CHIP\_RATE\_HZ / Galileo\_E1\_B\_CODE\_LENGTH\_CHIPS) / Galileo\_E1\_B\_SYMBOL\_RATE\_BPS*
- constexpr int32\_t `GALILEO_E1_C_SECONDARY_CODE_LENGTH` = 25  
*Galileo E1-C secondary code length [chips].*
- constexpr int32\_t `GALILEO_E1_NUMBER_OF_CODES` = 50
- constexpr uint32\_t `GALILEO_E1_OPT_ACQ_FS_SPS` = 2000000  
*Sampling frequency that maximizes the acquisition SNR while using a non-multiple of chip rate.*
- constexpr int32\_t `GALILEO_E1_HISTORY_DEEP` = 100  
*Observable history length for interpolation.*
- constexpr char `GALILEO_E1_C_SECONDARY_CODE` [26] = "0011100000001010110110010"
- constexpr size\_t `GALILEO_E1_B_PRIMARY_CODE_STR_LENGTH` = 1023
- constexpr char `GALILEO_E1_B_PRIMARY_CODE` [GALILEO\_E1\_NUMBER\_OF\_CODES][1024]
- constexpr size\_t `GALILEO_E1_C_PRIMARY_CODE_STR_LENGTH` = 1023
- constexpr char `GALILEO_E1_C_PRIMARY_CODE` [GALILEO\_E1\_NUMBER\_OF\_CODES][1024]
- constexpr double `GALILEO_E5A_FREQ_HZ` = `FREQ5`  
*Galileo E5a carrier frequency [Hz].*
- constexpr double `GALILEO_E5A_CODE_CHIP_RATE_CPS` = 1.023e7  
*Galileo E5a code rate [chips/s].*
- constexpr double `GALILEO_E5A_I_TIERED_CODE_PERIOD_S` = 0.020  
*Galileo E5a-I tiered code period [s].*
- constexpr double `GALILEO_E5A_Q_TIERED_CODE_PERIOD_S` = 0.100  
*Galileo E5a-Q tiered code period [s].*
- constexpr double `GALILEO_E5A_CODE_PERIOD_S` = 0.001  
*Galileo E5a primary code period [s].*
- constexpr int32\_t `GALILEO_E5A_CODE_LENGTH_CHIPS` = 10230  
*Galileo E5a primary code length [chips].*
- constexpr int32\_t `GALILEO_E5A_I_SECONDARY_CODE_LENGTH` = 20  
*Galileo E5a-I secondary code length [chips].*
- constexpr int32\_t `GALILEO_E5A_Q_SECONDARY_CODE_LENGTH` = 100  
*Galileo E5a-Q secondary code length [chips].*
- constexpr int32\_t `GALILEO_E5A_CODE_PERIOD_MS` = 1  
*Galileo E5a primary code period [ms].*
- constexpr int32\_t `GALILEO_E5A_SYMBOL_RATE_BPS` = 50  
*Galileo E5a symbol rate [bits/second].*
- constexpr int32\_t `GALILEO_E5A_NUMBER_OF_CODES` = 50
- constexpr int32\_t `GALILEO_E5A_HISTORY_DEEP` = 20
- constexpr int32\_t `GALILEO_E5A_CRC_ERROR_LIMIT` = 6
- constexpr uint32\_t `GALILEO_E5A_OPT_ACQ_FS_SPS` = 10000000  
*Sampling frequency that maximizes the acquisition SNR while using a non-multiple of chip rate.*
- constexpr int32\_t `GALILEO_FNAV_PREAMBLE_LENGTH_BITS` = 12
- constexpr int32\_t `GALILEO_FNAV_CODES_PER_SYMBOL` = 20

- `constexpr int32_t GALILEO_FNAV_CODES_PER_PREAMBLE = 240`
- `constexpr int32_t GALILEO_FNAV_SYMBOLS_PER_PAGE = 500`
- `constexpr int32_t GALILEO_FNAV_SECONDS_PER_PAGE = 10`
- `constexpr int32_t GALILEO_FNAV_CODES_PER_PAGE = 10000`
- `constexpr int32_t GALILEO_FNAV_INTERLEAVER_ROWS = 8`
- `constexpr int32_t GALILEO_FNAV_INTERLEAVER_COLS = 61`
- `constexpr int32_t GALILEO_FNAV_PAGE_TYPE_BITS = 6`
- `constexpr int32_t GALILEO_FNAV_DATA_FRAME_BITS = 214`
- `constexpr int32_t GALILEO_FNAV_DATA_FRAME_BYTES = 27`
- `constexpr char GALILEO_FNAV_PREAMBLE [13] = "101101110000"`
- `constexpr size_t GALILEO_E5A_I_PRIMARY_CODE_STR_LENGTH = 2558`
- `constexpr char GALILEO_E5A_I_PRIMARY_CODE [GALILEO_E5A_NUMBER_OF_CODES][2559]`
- `constexpr size_t GALILEO_E5A_Q_PRIMARY_CODE_STR_LENGTH = 2558`
- `constexpr char GALILEO_E5A_Q_PRIMARY_CODE [GALILEO_E5A_NUMBER_OF_CODES][2559]`
- `constexpr char GALILEO_E5A_I_SECONDARY_CODE [] = "10000100001011101001"`
- `constexpr size_t GALILEO_E5A_Q_SECONDARY_CODE_STR_LENGTH = 100`
- `constexpr char GALILEO_E5A_Q_SECONDARY_CODE [GALILEO_E5A_NUMBER_OF_CODES][101]`
- `constexpr double GALILEO_E5B_FREQ_HZ = FREQ7`  
*Galileo E5b carrier frequency [Hz].*
- `constexpr double GALILEO_E5B_CODE_CHIP_RATE_CPS = 1.023e7`  
*Galileo E5b code rate [chips/s].*
- `constexpr double GALILEO_E5B_I_TIERED_CODE_PERIOD_S = 0.004`  
*Galileo E5b-I tiered code period [s].*
- `constexpr double GALILEO_E5B_Q_TIERED_CODE_PERIOD_S = 0.100`  
*Galileo E5b-Q tiered code period [s].*
- `constexpr double GALILEO_E5B_CODE_PERIOD_S = 0.001`  
*Galileo E5b primary code period [s].*
- `constexpr int32_t GALILEO_E5B_CODE_PERIOD_MS = 1`  
*Galileo E5b primary code period [ms].*
- `constexpr int32_t GALILEO_E5B_CODE_LENGTH_CHIPS = 10230`  
*Galileo E5b primary code length [chips].*
- `constexpr int32_t GALILEO_E5B_I_SECONDARY_CODE_LENGTH = 4`  
*Galileo E5b-I secondary code length [chips].*
- `constexpr int32_t GALILEO_E5B_Q_SECONDARY_CODE_LENGTH = 100`  
*Galileo E5b-Q secondary code length [chips].*
- `constexpr int32_t GALILEO_E5B_SYMBOL_RATE_BPS = 250`  
*Galileo E5b symbol rate [bits/second].*
- `constexpr int32_t GALILEO_E5B_NUMBER_OF_CODES = 50`
- `constexpr int32_t GALILEO_E5B_HISTORY_DEEP = 100`
- `constexpr uint32_t GALILEO_E5B_OPT_ACQ_FS_SPS = 10000000`  
*Sampling frequency that maximizes the acquisition SNR while using a non-multiple of chip rate.*
- `constexpr char GALILEO_E5B_I_SECONDARY_CODE [5] = "1110"`
- `constexpr size_t GALILEO_E5B_I_PRIMARY_CODE_STR_LENGTH = 2558`
- `constexpr char GALILEO_E5B_I_PRIMARY_CODE [GALILEO_E5B_NUMBER_OF_CODES][2559]`
- `constexpr size_t GALILEO_E5B_Q_PRIMARY_CODE_STR_LENGTH = 2558`
- `constexpr char GALILEO_E5B_Q_PRIMARY_CODE [GALILEO_E5B_NUMBER_OF_CODES][2559]`
- `constexpr size_t GALILEO_E5B_Q_SECONDARY_CODE_STR_LENGTH = 100`
- `constexpr char GALILEO_E5B_Q_SECONDARY_CODE [GALILEO_E5B_NUMBER_OF_CODES][101]`
- `constexpr double GALILEO_E6_FREQ_HZ = FREQ6`  
*Galileo E6 carrier frequency [Hz].*
- `constexpr double GALILEO_E6_B_CODE_CHIP_RATE_CPS = 5.115e6`  
*Galileo E6 B code rate [chips/s].*

- constexpr double `GALILEO_E6_C_CODE_CHIP_RATE_CPS` = 5.115e6  
*Galileo E6 C code rate [chips/s].*
- constexpr double `GALILEO_E6_CODE_PERIOD_S` = 0.001  
*Galileo E6 code period [s].*
- constexpr double `GALILEO_E6_B_CODE_LENGTH_CHIPS` = 5115.0  
*Galileo E6 B code length [chips].*
- constexpr double `GALILEO_E6_C_CODE_LENGTH_CHIPS` = 5115.0  
*Galileo E6 C code length [chips].*
- constexpr double `GALILEO_E6_C_SECONDARY_CODE_LENGTH_CHIPS` = 100.0  
*Galileo E6 C secondary code length [chips].*
- constexpr uint32\_t `GALILEO_E6_CODE_PERIOD_MS` = 1  
*Galileo E& B/C code period [ms].*
- constexpr int32\_t `GALILEO_E6_NUMBER_OF_CODES` = 50
- constexpr uint32\_t `GALILEO_E6_OPT_ACQ_FS_SPS` = 1000000
- constexpr size\_t `GALILEO_E6_B_PRIMARY_CODE_STR_LENGTH` = 1279
- constexpr char `GALILEO_E6_B_PRIMARY_CODE` [`GALILEO_E6_NUMBER_OF_CODES`][1280]
- constexpr size\_t `GALILEO_E6_C_PRIMARY_CODE_STR_LENGTH` = 1279
- constexpr char `GALILEO_E6_C_PRIMARY_CODE` [`GALILEO_E6_NUMBER_OF_CODES`][1280]
- constexpr size\_t `GALILEO_E6_C_SECONDARY_CODE_STR_LENGTH` = 25
- constexpr char `GALILEO_E6_C_SECONDARY_CODE` [`GALILEO_E6_NUMBER_OF_CODES`][26]
- constexpr int32\_t `FNAV_T0C_1_LSB` = 60
- constexpr double `FNAV_AF0_1_LSB` = `TWO_N34`
- constexpr double `FNAV_AF1_1_LSB` = `TWO_N46`
- constexpr double `FNAV_AF2_1_LSB` = `TWO_N59`
- constexpr double `FNAV_AI0_1_LSB` = `TWO_N2`
- constexpr double `FNAV_AI1_1_LSB` = `TWO_N8`
- constexpr double `FNAV_AI2_1_LSB` = `TWO_N15`
- constexpr double `FNAV_BGD_1_LSB` = `TWO_N32`
- constexpr double `FNAV_M0_2_LSB` = `PI_TWO_N31`
- constexpr double `FNAV_OMEGADOT_2_LSB` = `PI_TWO_N43`
- constexpr double `FNAV_E_2_LSB` = `TWO_N33`
- constexpr double `FNAV_A12_2_LSB` = `TWO_N19`
- constexpr double `FNAV_OMEGA0_2_LSB` = `PI_TWO_N31`
- constexpr double `FNAV_IDOT_2_LSB` = `PI_TWO_N43`
- constexpr double `FNAV_I0_3_LSB` = `PI_TWO_N31`
- constexpr double `FNAV_W_3_LSB` = `PI_TWO_N31`
- constexpr double `FNAV_DELTAN_3_LSB` = `PI_TWO_N43`
- constexpr double `FNAV_CUC_3_LSB` = `TWO_N29`
- constexpr double `FNAV_CUS_3_LSB` = `TWO_N29`
- constexpr double `FNAV_CRC_3_LSB` = `TWO_N5`
- constexpr double `FNAV_CRS_3_LSB` = `TWO_N5`
- constexpr int32\_t `FNAV_T0E_3_LSB` = 60
- constexpr double `FNAV_CIC_4_LSB` = `TWO_N29`
- constexpr double `FNAV_CIS_4_LSB` = `TWO_N29`
- constexpr double `FNAV_A0_4_LSB` = `TWO_N30`
- constexpr double `FNAV_A1_4_LSB` = `TWO_N50`
- constexpr int32\_t `FNAV_T0T_4_LSB` = 3600
- constexpr int32\_t `FNAV_T0G_4_LSB` = 3600
- constexpr double `FNAV_A0G_4_LSB` = `TWO_N35`
- constexpr double `FNAV_A1G_4_LSB` = `TWO_N51`
- constexpr int32\_t `FNAV_T0A_5_LSB` = 600
- constexpr double `FNAV_DELTAA12_5_LSB` = `TWO_N9`
- constexpr double `FNAV_E_5_LSB` = `TWO_N16`
- constexpr double `FNAV_W_5_LSB` = `TWO_N15`

- constexpr double `FNAV_DELTAI_5_LSB` = `TWO_N14`
- constexpr double `FNAV_OMEGA0_5_LSB` = `TWO_N15`
- constexpr double `FNAV_OMEGADOT_5_LSB` = `TWO_N33`
- constexpr double `FNAV_M0_5_LSB` = `TWO_N15`
- constexpr double `FNAV_AF0_5_LSB` = `TWO_N19`
- constexpr double `FNAV_AF1_5_LSB` = `TWO_N38`
- constexpr double `GALILEO_INAV_PAGE_PART_WITH_PREABLE_SECONDS` = 2.04

*Page Duration + (Galileo I/NAV Preamble bits)\*(Galileo E5b-I tiered Code Period(seconds))*

- constexpr uint32\_t `GALILEO_INAV_PAGE_SYMBOLS` = 500

*The complete Galileo INAV page length.*

- constexpr int32\_t `GALILEO_INAV_PREAMBLE_LENGTH_BITS` = 10
- constexpr int32\_t `GALILEO_INAV_PREAMBLE_PERIOD_SYMBOLS` = 250
- constexpr int32\_t `GALILEO_INAV_PAGE_PART_SYMBOLS` = 250

*Each Galileo INAV pages are composed of two parts (even and odd) each of 250 symbols, including preamble. See Galileo ICD 4.3.2.*

- constexpr int32\_t `GALILEO_INAV_PAGE_PART_SECONDS` = 1
- constexpr int32\_t `GALILEO_INAV_PAGE_PART_MS` = 1000
- constexpr int32\_t `GALILEO_INAV_PAGE_SECONDS` = 2
- constexpr int32\_t `GALILEO_INAV_INTERLEAVER_ROWS` = 8
- constexpr int32\_t `GALILEO_INAV_INTERLEAVER_COLS` = 30
- constexpr int32\_t `GALILEO_TELEMETRY_RATE_BITS_SECOND` = 250
- constexpr int32\_t `GALILEO_PAGE_TYPE_BITS` = 6
- constexpr int32\_t `GALILEO_DATA_JK_BITS` = 128
- constexpr int32\_t `GALILEO_DATA_FRAME_BITS` = 196
- constexpr int32\_t `GALILEO_DATA_FRAME_BYTES` = 25
- constexpr char `GALILEO_INAV_PREAMBLE` [11] = "0101100000"
- constexpr int32\_t `T0E_1_LSB` = 60
- constexpr double `M0_1_LSB` = `PI_TWO_N31`
- constexpr double `E_1_LSB` = `TWO_N33`
- constexpr double `A_1_LSB_GAL` = `TWO_N19`
- constexpr double `OMEGA_0_2_LSB` = `PI_TWO_N31`
- constexpr double `I_0_2_LSB` = `PI_TWO_N31`
- constexpr double `OMEGA_2_LSB` = `PI_TWO_N31`
- constexpr double `I_DOT_2_LSB` = `PI_TWO_N43`
- constexpr double `OMEGA_DOT_3_LSB` = `PI_TWO_N43`
- constexpr double `DELTA_N_3_LSB` = `PI_TWO_N43`
- constexpr double `C_UC_3_LSB` = `TWO_N29`
- constexpr double `C_US_3_LSB` = `TWO_N29`
- constexpr double `C_RC_3_LSB` = `TWO_N5`
- constexpr double `C_RS_3_LSB` = `TWO_N5`
- constexpr double `C_IC_4_LSB` = `TWO_N29`
- constexpr double `C_IS_4_LSB` = `TWO_N29`
- constexpr int32\_t `T0C_4_LSB` = 60
- constexpr double `AF0_4_LSB` = `TWO_N34`
- constexpr double `AF1_4_LSB` = `TWO_N46`
- constexpr double `AF2_4_LSB` = `TWO_N59`
- constexpr double `A10_5_LSB` = `TWO_N2`
- constexpr double `A11_5_LSB` = `TWO_N8`
- constexpr double `A12_5_LSB` = `TWO_N15`
- constexpr double `BGD_E1_E5A_5_LSB` = `TWO_N32`
- constexpr double `BGD_E1_E5B_5_LSB` = `TWO_N32`
- constexpr double `A0_6_LSB` = `TWO_N30`
- constexpr double `A1_6_LSB` = `TWO_N50`
- constexpr int32\_t `T0T_6_LSB` = 3600



- constexpr int32\_t T0A\_7\_LSB = 600
- constexpr double DELTA\_A\_7\_LSB = TWO\_N9
- constexpr double E\_7\_LSB = TWO\_N16
- constexpr double OMEGA\_7\_LSB = TWO\_N15
- constexpr double DELTA\_I\_7\_LSB = TWO\_N14
- constexpr double OMEGA0\_7\_LSB = TWO\_N15
- constexpr double OMEGA\_DOT\_7\_LSB = TWO\_N33
- constexpr double M0\_7\_LSB = TWO\_N15
- constexpr double AF0\_8\_LSB = TWO\_N19
- constexpr double AF1\_8\_LSB = TWO\_N38
- constexpr double DELTA\_A\_8\_LSB = TWO\_N9
- constexpr double E\_8\_LSB = TWO\_N16
- constexpr double OMEGA\_8\_LSB = TWO\_N15
- constexpr double DELTA\_I\_8\_LSB = TWO\_N14
- constexpr double OMEGA0\_8\_LSB = TWO\_N15
- constexpr double OMEGA\_DOT\_8\_LSB = TWO\_N33
- constexpr int32\_t T0A\_9\_LSB = 600
- constexpr double M0\_9\_LSB = TWO\_N15
- constexpr double AF0\_9\_LSB = TWO\_N19
- constexpr double AF1\_9\_LSB = TWO\_N38
- constexpr double DELTA\_A\_9\_LSB = TWO\_N9
- constexpr double E\_9\_LSB = TWO\_N16
- constexpr double OMEGA\_9\_LSB = TWO\_N15
- constexpr double DELTA\_I\_9\_LSB = TWO\_N14
- constexpr double OMEGA0\_10\_LSB = TWO\_N15
- constexpr double OMEGA\_DOT\_10\_LSB = TWO\_N33
- constexpr double M0\_10\_LSB = TWO\_N15
- constexpr double AF0\_10\_LSB = TWO\_N19
- constexpr double AF1\_10\_LSB = TWO\_N38
- constexpr double A\_0G\_10\_LSB = TWO\_N35
- constexpr double A\_1G\_10\_LSB = TWO\_N51
- constexpr int32\_t T\_0\_G\_10\_LSB = 3600
- constexpr double CED\_DeltaAred\_LSB = TWO\_P8
- constexpr double CED\_exred\_LSB = TWO\_N22
- constexpr double CED\_eyred\_LSB = TWO\_N22
- constexpr double CED\_DeltaI0red\_LSB = TWO\_N22
- constexpr double CED\_Omega0red\_LSB = TWO\_N22
- constexpr double CED\_lambda0red\_LSB = TWO\_N22
- constexpr double CED\_af0red\_LSB = TWO\_N26
- constexpr double CED\_af1red\_LSB = TWO\_N35
- constexpr size\_t INAV\_RS\_SUBVECTOR\_LENGTH = 15
- constexpr size\_t INAV\_RS\_PARITY\_VECTOR\_LENGTH = 60
- constexpr size\_t INAV\_RS\_INFO\_VECTOR\_LENGTH = 58
- constexpr size\_t INAV\_RS\_BUFFER\_LENGTH = 118
- constexpr int32\_t BITS\_IN\_OCTET = 8
- constexpr int32\_t FIRST\_RS\_BIT = 7
- constexpr int32\_t FIRST\_RS\_BIT\_AFTER\_IODNAV = 17
- constexpr int32\_t GALILEO\_ISM\_CRC\_DATA\_BITS = 96
- constexpr int32\_t GALILEO\_ISM\_CRC\_DATA\_BYTES = 12
- constexpr char GALILEO\_INAV\_PLAIN\_SSP1 [9] = "00000100"
- constexpr char GALILEO\_INAV\_PLAIN\_SSP2 [9] = "00101011"
- constexpr char GALILEO\_INAV\_PLAIN\_SSP3 [9] = "00101111"
- constexpr char GALILEO\_INAV\_ENCODED\_SSP1 [17] = "1110100100100101"
- constexpr char GALILEO\_INAV\_ENCODED\_SSP2 [17] = "0110110001001110"
- constexpr char GALILEO\_INAV\_ENCODED\_SSP3 [17] = "1101000000111110"

- constexpr size\_t [SIZE\\_DSM\\_BLOCKS\\_BYTES](#) = 13
- const std::unordered\_map< uint8\_t, std::string > [OSNMA\\_TABLE\\_1](#)
- const std::unordered\_map< uint8\_t, std::string > [OSNMA\\_TABLE\\_2](#)
- const std::unordered\_map< uint8\_t, std::pair< uint16\_t, uint16\_t > > [OSNMA\\_TABLE\\_3](#)
- const std::unordered\_map< uint8\_t, std::string > [OSNMA\\_TABLE\\_5](#)
- const std::unordered\_map< std::string, uint16\_t > [OSNMA\\_TABLE\\_6](#)
- const std::unordered\_map< uint8\_t, std::pair< uint16\_t, uint16\_t > > [OSNMA\\_TABLE\\_7](#)
- const std::unordered\_map< uint8\_t, std::string > [OSNMA\\_TABLE\\_8](#)
- const std::unordered\_map< uint8\_t, uint16\_t > [OSNMA\\_TABLE\\_10](#)
- const std::unordered\_map< uint8\_t, uint8\_t > [OSNMA\\_TABLE\\_11](#)
- const std::unordered\_map< std::string, uint16\_t > [OSNMA\\_TABLE\\_15](#)
- const std::unordered\_map< uint8\_t, [Mack\\_lookup](#) > [OSNMA\\_TABLE\\_16](#)
- constexpr double [GLONASS\\_F\\_M\\_A](#) = 0.35e9  
*Gravitational constant of atmosphere [ $m^3/s^2$ ].*
- constexpr double [GLONASS\\_SEMI\\_MAJOR\\_AXIS](#) = 6378136  
*Semi-major axis of Earth [m].*
- constexpr double [GLONASS\\_FLATTENING](#) = 1.0 / 29825784.0  
*Flattening parameter.*
- constexpr double [GLONASS\\_GRAVITY](#) = 97803284.0  
*Equatorial acceleration of gravity [mGal].*
- constexpr double [GLONASS\\_GRAVITY\\_CORRECTION](#) = 0.87  
*Correction to acceleration of gravity at sea-level due to Atmosphere[uGal].*
- constexpr double [GLONASS\\_J2](#) = 1082625.75e-9  
*Second zonal harmonic of the geopotential.*
- constexpr double [GLONASS\\_J4](#) = -2370.89e-9  
*Fourth zonal harmonic of the geopotential.*
- constexpr double [GLONASS\\_J6](#) = 6.08e-9  
*Sixth zonal harmonic of the geopotential.*
- constexpr double [GLONASS\\_J8](#) = 1.40e-11  
*Eighth zonal harmonic of the geopotential.*
- constexpr double [GLONASS\\_U0](#) = 62636861.4  
*Normal potential at surface of common terrestrial ellipsoid [ $m^2/s^2$ ].*
- constexpr double [GLONASS\\_C20](#) = -1082.63e-6  
*Second zonal coefficient of spherical harmonic expansion.*
- constexpr double [GLONASS\\_EARTH\\_RADIUS](#) = 6378.136  
*Equatorial radius of Earth [km].*
- constexpr double [GLONASS\\_EARTH\\_INCLINATION](#) = 0.000409148809899e3  
*Mean inclination of ecliptic to equator (23 deg 26 min 33 sec) [rad].*
- constexpr double [GLONASS\\_TAU\\_0](#) = -0.005835151531174e3  
*(-334 deg 19 min 46.40 sec) [rad];*
- constexpr double [GLONASS\\_TAU\\_1](#) = 0.071018041257371e3  
*(4069 deg 02 min 02.52 sec) [rad];*
- constexpr double [GLONASS\\_MOON\\_Q0](#) = -0.001115184961435e3  
*(-63 deg 53 min 43.41 sec) [rad]*
- constexpr double [GLONASS\\_MOON\\_Q1](#) = 8.328691103668023e3  
*(477198 deg 50 min 56.79 sec) [rad]*
- constexpr double [GLONASS\\_MOON\\_OMEGA\\_0](#) = 0.004523601514852e3  
*(259 deg 10 min 59.79 sec) [rad]*
- constexpr double [GLONASS\\_MOON\\_OMEGA\\_1](#) = -0.033757146246552e3  
*(-1934 deg 08 min 31.23 sec) [rad]*
- constexpr double [GLONASS\\_MOON\\_GM](#) = 4902.835



- Lunar gravitational constant  $[km^3/s^2]$ .*

  - constexpr double `GLONASS_MOON_SEMI_MAJOR_AXIS` = 3.84385243e5

*Semi-major axis of lunar orbit [km];.*
- constexpr double `GLONASS_MOON_ECCENTRICITY` = 0.054900489

*Eccentricity of lunar orbit.*
- constexpr double `GLONASS_MOON_INCLINATION` = 0.000089803977407e3

*Inclination of lunar orbit to ecliptic plane (5 deg 08 min 43.4 sec) [rad].*
- constexpr double `GLONASS_SUN_OMEGA` = 0.004908229466869e3

*TODO What is this operation in the seconds with  $T$ ? (281 deg 13 min 15.0 + 6189.03 x  $T$  sec) [rad].*
- constexpr double `GLONASS_SUN_Q0` = 0.006256583774423e3

*(358 deg 28 min 33.04 sec) [rad]*
- constexpr double `GLONASS_SUN_Q1` = 0e3

*TODO Why is the value greater than 60? (129596579.10 sec) [rad].*
- constexpr double `GLONASS_SUN_GM` = 0.1325263e12

*Solar gravitational constant  $[km^3/s^2]$ .*
- constexpr double `GLONASS_SUN_SEMI_MAJOR_AXIS` = 1.49598e8

*Semi-major axis of solar orbit [km];.*
- constexpr double `GLONASS_SUN_ECCENTRICITY` = 0.016719

*Eccentricity of solar orbit.*
- constexpr double `GLONASS_L2_CA_FREQ_HZ` = `FREQ2_GLO`

*L2 [Hz].*
- constexpr double `GLONASS_L2_CA_DFREQ_HZ` = `DFRQ2_GLO`

*Freq Bias for GLONASS L1 [Hz].*
- constexpr double `GLONASS_L2_CA_CODE_RATE_CPS` = 0.511e6

*GLONASS L1 C/A code rate [chips/s].*
- constexpr double `GLONASS_L2_CA_CODE_LENGTH_CHIPS` = 511.0

*GLONASS L1 C/A code length [chips].*
- constexpr double `GLONASS_L2_CA_CODE_PERIOD_S` = 0.001

*GLONASS L1 C/A code period [seconds].*
- constexpr double `GLONASS_L2_CA_CHIP_PERIOD_S` = 1.9569e-06

*GLONASS L1 C/A chip period [seconds].*
- constexpr double `GLONASS_L2_CA_SYMBOL_RATE_BPS` = 1000.0
- constexpr double `GLONASS_L1_CA_FREQ_HZ` = `FREQ1_GLO`

*L1 [Hz].*
- constexpr double `GLONASS_L1_CA_DFREQ_HZ` = `DFRQ1_GLO`

*Freq Bias for GLONASS L1 [Hz].*
- constexpr double `GLONASS_L1_CA_CODE_RATE_CPS` = 0.511e6

*GLONASS L1 C/A code rate [chips/s].*
- constexpr double `GLONASS_L1_CA_CODE_LENGTH_CHIPS` = 511.0

*GLONASS L1 C/A code length [chips].*
- constexpr double `GLONASS_L1_CA_CODE_PERIOD_S` = 0.001

*GLONASS L1 C/A code period [seconds].*
- constexpr double `GLONASS_L1_CA_CHIP_PERIOD_S` = 1.9569e-06

*GLONASS L1 C/A chip period [seconds].*
- constexpr double `GLONASS_L1_CA_SYMBOL_RATE_BPS` = 1000.0
- constexpr int32\_t `GLONASS_CA_NBR_SATS` = 24
- constexpr int32\_t `GLONASS_L1_CA_HISTORY_DEEP` = 100
- constexpr double `GLONASS_GNAV_PREAMBLE_DURATION_S` = 0.300
- constexpr int32\_t `GLONASS_GNAV_PREAMBLE_LENGTH_BITS` = 30
- constexpr int32\_t `GLONASS_GNAV_PREAMBLE_LENGTH_SYMBOLS` = 300
- constexpr int32\_t `GLONASS_GNAV_PREAMBLE_PERIOD_SYMBOLS` = 2000

- constexpr int32\_t [GLONASS\\_GNAV\\_TELEMETRY\\_RATE\\_BITS\\_SECOND](#) = 50  
*NAV message bit rate [bits/s].*
- constexpr int32\_t [GLONASS\\_GNAV\\_TELEMETRY\\_SYMBOLS\\_PER\\_BIT](#) = 10
- constexpr int32\_t [GLONASS\\_GNAV\\_TELEMETRY\\_SYMBOLS\\_PER\\_PREAMBLE\\_BIT](#) = 10
- constexpr int32\_t [GLONASS\\_GNAV\\_TELEMETRY\\_RATE\\_SYMBOLS\\_SECOND](#) = [GLONASS\\_GNAV\\_TELEMETRY\\_RATE\\_BITS\\_SECOND](#) \* [GLONASS\\_GNAV\\_TELEMETRY\\_SYMBOLS\\_PER\\_BIT](#)  
*NAV message bit rate [symbols/s].*
- constexpr int32\_t [GLONASS\\_GNAV\\_STRING\\_SYMBOLS](#) = 2000  
*Number of bits per string in the GNAV message (85 data bits + 30 time mark bits) [bits].*
- constexpr int32\_t [GLONASS\\_GNAV\\_STRING\\_BITS](#) = 85  
*Number of bits per string in the GNAV message (85 data bits + 30 time mark bits) [bits].*
- constexpr int32\_t [GLONASS\\_GNAV\\_HAMMING\\_CODE\\_BITS](#) = 8  
*Number of bits in hamming code sequence of GNAV message.*
- constexpr int32\_t [GLONASS\\_GNAV\\_DATA\\_SYMBOLS](#) = 1700
- constexpr double [GLONASS\\_LEAP\\_SECONDS](#) [19][7]  
*Record of leap seconds definition for GLOT to GPST conversion and vice versa.*
- const std::map< uint32\_t, int32\_t > [GLONASS\\_PRN](#)
- const std::vector< int32\_t > [GLONASS\\_GNAV\\_CRC\\_I\\_INDEX](#) {9, 10, 12, 13, 15, 17, 19, 20, 22, 24, 26, 28, 30, 32, 34, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84}
- const std::vector< int32\_t > [GLONASS\\_GNAV\\_CRC\\_J\\_INDEX](#) {9, 11, 12, 14, 15, 18, 19, 21, 22, 25, 26, 29, 30, 33, 34, 36, 37, 40, 41, 44, 45, 48, 49, 52, 53, 56, 57, 60, 61, 64, 65, 67, 68, 71, 72, 75, 76, 79, 80, 83, 84}
- const std::vector< int32\_t > [GLONASS\\_GNAV\\_CRC\\_K\\_INDEX](#) {10, 11, 12, 16, 17, 18, 19, 23, 24, 25, 26, 31, 32, 33, 34, 38, 39, 40, 41, 46, 47, 48, 49, 54, 55, 56, 57, 62, 63, 64, 65, 69, 70, 71, 72, 77, 78, 79, 80, 85}
- const std::vector< int32\_t > [GLONASS\\_GNAV\\_CRC\\_L\\_INDEX](#) {13, 14, 15, 16, 17, 18, 19, 27, 28, 29, 30, 31, 32, 33, 34, 42, 43, 44, 45, 46, 47, 48, 49, 58, 59, 60, 61, 62, 63, 64, 65, 73, 74, 75, 76, 77, 78, 79, 80}
- const std::vector< int32\_t > [GLONASS\\_GNAV\\_CRC\\_M\\_INDEX](#) {20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 81, 82, 83, 84, 85}
- const std::vector< int32\_t > [GLONASS\\_GNAV\\_CRC\\_N\\_INDEX](#) {35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65}
- const std::vector< int32\_t > [GLONASS\\_GNAV\\_CRC\\_P\\_INDEX](#) {66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85}
- const std::vector< int32\_t > [GLONASS\\_GNAV\\_CRC\\_Q\\_INDEX](#) {9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85}
- const std::vector< int32\_t > [GLONASS\\_GNAV\\_ECC\\_LOCATOR](#) {0, 0, 1, 8, 2, 9, 10, 11, 3, 12, 13, 14, 15, 16, 17, 18, 4, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 5, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 6, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84}
- constexpr double [FREQ1](#) = 1.57542e9  
*L1/E1 frequency (Hz)*
- constexpr double [FREQ2](#) = 1.22760e9  
*L2 frequency (Hz)*
- constexpr double [FREQ5](#) = 1.17645e9  
*L5/E5a frequency (Hz)*
- constexpr double [FREQ6](#) = 1.27875e9  
*E6/LEX frequency (Hz)*
- constexpr double [FREQ7](#) = 1.20714e9  
*E5b frequency (Hz)*
- constexpr double [FREQ8](#) = 1.191795e9  
*E5a+b frequency (Hz)*
- constexpr double [FREQ9](#) = 2.492028e9  
*S frequency (Hz)*

- constexpr double `FREQ1_GLO` = 1.60200e9  
*GLONASS G1 base frequency (Hz)*
- constexpr double `DFRQ1_GLO` = 0.56250e6  
*GLONASS G1 bias frequency (Hz/n)*
- constexpr double `FREQ2_GLO` = 1.24600e9  
*GLONASS G2 base frequency (Hz)*
- constexpr double `DFRQ2_GLO` = 0.43750e6  
*GLONASS G2 bias frequency (Hz/n)*
- constexpr double `FREQ3_GLO` = 1.202025e9  
*GLONASS G3 frequency (Hz)*
- constexpr double `FREQ1_BDS` = 1.561098e9  
*BeiDou B1 frequency (Hz)*
- constexpr double `FREQ2_BDS` = 1.20714e9  
*BeiDou B2 frequency (Hz)*
- constexpr double `FREQ3_BDS` = 1.26852e9  
*BeiDou B3 frequency (Hz)*
- const std::unordered\_map< std::string, double > `SIGNAL_FREQ_MAP`
- constexpr uint32\_t `CODE_NONE` = 0  
*obs code: none or unknown*
- constexpr uint32\_t `CODE_L1C` = 1  
*obs code: L1C/A, G1C/A, E1C (GPS, GLO, GAL, QZS, SBS)*
- constexpr uint32\_t `CODE_L1P` = 2  
*obs code: L1P, G1P (GPS, GLO)*
- constexpr uint32\_t `CODE_L1W` = 3  
*obs code: L1 Z-track (GPS)*
- constexpr uint32\_t `CODE_L1Y` = 4  
*obs code: L1Y (GPS)*
- constexpr uint32\_t `CODE_L1M` = 5  
*obs code: L1M (GPS)*
- constexpr uint32\_t `CODE_L1N` = 6  
*obs code: L1codeless (GPS)*
- constexpr uint32\_t `CODE_L1S` = 7  
*obs code: L1C(D) (GPS, QZS)*
- constexpr uint32\_t `CODE_L1L` = 8  
*obs code: L1C(P) (GPS, QZS)*
- constexpr uint32\_t `CODE_L1E` = 9  
*(not used)*
- constexpr uint32\_t `CODE_L1A` = 10  
*obs code: E1A (GAL)*
- constexpr uint32\_t `CODE_L1B` = 11  
*obs code: E1B (GAL)*
- constexpr uint32\_t `CODE_L1X` = 12  
*obs code: E1B+C, L1C(D+P) (GAL, QZS)*
- constexpr uint32\_t `CODE_L1Z` = 13  
*obs code: E1A+B+C, L1SAIF (GAL, QZS)*
- constexpr uint32\_t `CODE_L2C` = 14  
*obs code: L2C/A, G1C/A (GPS, GLO)*
- constexpr uint32\_t `CODE_L2D` = 15  
*obs code: L2 L1C/A-(P2-P1) (GPS)*
- constexpr uint32\_t `CODE_L2S` = 16  
*obs code: L2C(M) (GPS, QZS)*

- constexpr uint32\_t [CODE\\_L2L](#) = 17  
*obs code: L2C(L) (GPS,QZS)*
- constexpr uint32\_t [CODE\\_L2X](#) = 18  
*obs code: L2C(M+L),B1I+Q (GPS,QZS,BDS)*
- constexpr uint32\_t [CODE\\_L2P](#) = 19  
*obs code: L2P,G2P (GPS,GLO)*
- constexpr uint32\_t [CODE\\_L2W](#) = 20  
*obs code: L2 Z-track (GPS)*
- constexpr uint32\_t [CODE\\_L2Y](#) = 21  
*obs code: L2Y (GPS)*
- constexpr uint32\_t [CODE\\_L2M](#) = 22  
*obs code: L2M (GPS)*
- constexpr uint32\_t [CODE\\_L2N](#) = 23  
*obs code: L2codeless (GPS)*
- constexpr uint32\_t [CODE\\_L5I](#) = 24  
*obs code: L5/E5aI (GPS,GAL,QZS,SBS)*
- constexpr uint32\_t [CODE\\_L5Q](#) = 25  
*obs code: L5/E5aQ (GPS,GAL,QZS,SBS)*
- constexpr uint32\_t [CODE\\_L5X](#) = 26  
*obs code: L5/E5aI+Q/L5B+C (GPS,GAL,QZS,IRN,SBS)*
- constexpr uint32\_t [CODE\\_L7I](#) = 27  
*obs code: E5bI,B2I (GAL,BDS)*
- constexpr uint32\_t [CODE\\_L7Q](#) = 28  
*obs code: E5bQ,B2Q (GAL,BDS)*
- constexpr uint32\_t [CODE\\_L7X](#) = 29  
*obs code: E5bI+Q,B2I+Q (GAL,BDS)*
- constexpr uint32\_t [CODE\\_L6A](#) = 30  
*obs code: E6A (GAL)*
- constexpr uint32\_t [CODE\\_L6B](#) = 31  
*obs code: E6B (GAL)*
- constexpr uint32\_t [CODE\\_L6C](#) = 32  
*obs code: E6C (GAL)*
- constexpr uint32\_t [CODE\\_L6X](#) = 33  
*obs code: E6B+C,LEXS+L,B3I+Q (GAL,QZS,BDS)*
- constexpr uint32\_t [CODE\\_L6Z](#) = 34  
*obs code: E6A+B+C (GAL)*
- constexpr uint32\_t [CODE\\_L6S](#) = 35  
*obs code: LEXS (QZS)*
- constexpr uint32\_t [CODE\\_L6L](#) = 36  
*obs code: LEXL (QZS)*
- constexpr uint32\_t [CODE\\_L8I](#) = 37  
*obs code: E5(a+b)I (GAL)*
- constexpr uint32\_t [CODE\\_L8Q](#) = 38  
*obs code: E5(a+b)Q (GAL)*
- constexpr uint32\_t [CODE\\_L8X](#) = 39  
*obs code: E5(a+b)I+Q (GAL)*
- constexpr uint32\_t [CODE\\_L2I](#) = 40  
*obs code: B1I (BDS)*
- constexpr uint32\_t [CODE\\_L2Q](#) = 41  
*obs code: B1Q (BDS)*
- constexpr uint32\_t [CODE\\_L6I](#) = 42

- obs code: B3I (BDS)*
- constexpr uint32\_t `CODE_L6Q` = 43
- obs code: B3Q (BDS)*
- constexpr uint32\_t `CODE_L3I` = 44
- obs code: G3I (GLO)*
- constexpr uint32\_t `CODE_L3Q` = 45
- obs code: G3Q (GLO)*
- constexpr uint32\_t `CODE_L3X` = 46
- obs code: G3I+Q (GLO)*
- constexpr uint32\_t `CODE_L1I` = 47
- obs code: B1I (BDS)*
- constexpr uint32\_t `CODE_L1Q` = 48
- obs code: B1Q (BDS)*
- constexpr uint32\_t `CODE_L5A` = 49
- obs code: L5A SPS (IRN)*
- constexpr uint32\_t `CODE_L5B` = 50
- obs code: L5B RS(D) (IRN)*
- constexpr uint32\_t `CODE_L5C` = 51
- obs code: L5C RS(P) (IRN)*
- constexpr uint32\_t `CODE_L9A` = 52
- obs code: SA SPS (IRN)*
- constexpr uint32\_t `CODE_L9B` = 53
- obs code: SB RS(D) (IRN)*
- constexpr uint32\_t `CODE_L9C` = 54
- obs code: SC RS(P) (IRN)*
- constexpr uint32\_t `CODE_L9X` = 55
- obs code: SB+C (IRN)*
- constexpr int32\_t `MAXCODE` = 55
- max number of obs code*
- constexpr int32\_t `GPS_CNAV_DATA_PAGE_BITS` = 300
- constexpr int32\_t `CNAV_TOW_LSB` = 6
- constexpr int32\_t `CNAV_TOP1_LSB` = 300
- constexpr int32\_t `CNAV_TOE1_LSB` = 300
- constexpr double `CNAV_DELTA_A_LSB` = `TWO_N9`
- constexpr double `CNAV_A_DOT_LSB` = `TWO_N21`
- constexpr double `CNAV_DELTA_N0_LSB` = `TWO_N44 * GNSS_PI`
- constexpr double `CNAV_DELTA_N0_DOT_LSB` = `TWO_N57 * GNSS_PI`
- constexpr double `CNAV_M0_LSB` = `TWO_N32 * GNSS_PI`
- constexpr double `CNAV_E_ECCENTRICITY_LSB` = `TWO_N34`
- constexpr double `CNAV_OMEGA_LSB` = `TWO_N32 * GNSS_PI`
- constexpr int32\_t `CNAV_TOE2_LSB` = 300
- constexpr double `CNAV_OMEGA0_LSB` = `TWO_N32 * GNSS_PI`
- constexpr double `CNAV_I0_LSB` = `TWO_N32 * GNSS_PI`
- constexpr double `CNAV_DELTA_OMEGA_DOT_LSB` = `TWO_N44 * GNSS_PI`
- constexpr double `CNAV_I0_DOT_LSB` = `TWO_N44 * GNSS_PI`
- constexpr double `CNAV_CIS_LSB` = `TWO_N30`
- constexpr double `CNAV_CIC_LSB` = `TWO_N30`
- constexpr double `CNAV_CRS_LSB` = `TWO_N8`
- constexpr double `CNAV_CRC_LSB` = `TWO_N8`
- constexpr double `CNAV_CUS_LSB` = `TWO_N30`
- constexpr double `CNAV_CUC_LSB` = `TWO_N30`
- constexpr int32\_t `CNAV_TOP2_LSB` = 300

- constexpr int32\_t CNAV\_TOC\_LSB = 300
  - constexpr double CNAV\_AF0\_LSB = TWO\_N35
  - constexpr double CNAV\_AF1\_LSB = TWO\_N48
  - constexpr double CNAV\_AF2\_LSB = TWO\_N60
  - constexpr double CNAV\_TGD\_LSB = TWO\_N35
  - constexpr double CNAV\_ISCL1\_LSB = TWO\_N35
  - constexpr double CNAV\_ISCL2\_LSB = TWO\_N35
  - constexpr double CNAV\_ISCL5I\_LSB = TWO\_N35
  - constexpr double CNAV\_ISCL5Q\_LSB = TWO\_N35
  - constexpr double CNAV\_ALPHA0\_LSB = TWO\_N30
  - constexpr double CNAV\_ALPHA1\_LSB = TWO\_N27
  - constexpr double CNAV\_ALPHA2\_LSB = TWO\_N24
  - constexpr double CNAV\_ALPHA3\_LSB = TWO\_N24
  - constexpr double CNAV\_BETA0\_LSB = TWO\_P11
  - constexpr double CNAV\_BETA1\_LSB = TWO\_P14
  - constexpr double CNAV\_BETA2\_LSB = TWO\_P16
  - constexpr double CNAV\_BETA3\_LSB = TWO\_P16
  - constexpr double CNAV\_A0\_LSB = TWO\_N35
  - constexpr double CNAV\_A1\_LSB = TWO\_N51
  - constexpr double CNAV\_A2\_LSB = TWO\_N68
  - constexpr int32\_t CNAV\_DELTA\_TLS\_LSB = 1
  - constexpr int32\_t CNAV\_TOT\_LSB = TWO\_P4
  - constexpr int32\_t CNAV\_WN\_OT\_LSB = 1
  - constexpr int32\_t CNAV\_WN\_LSF\_LSB = 1
  - constexpr int32\_t CNAV\_DN\_LSB = 1
  - constexpr int32\_t CNAV\_DELTA\_TLSF\_LSB = 1
  - constexpr double CNAV\_A\_REF = 26559710.0
  - constexpr double CNAV\_OMEGA\_DOT\_REF = -2.6e-9
  - constexpr double GPS\_L1\_FREQ\_HZ = FREQ1
- L1 [Hz].*
- constexpr double GPS\_L1\_CA\_CODE\_RATE\_CPS = 1.023e6
- GPS L1 C/A code rate [chips/s].*
- constexpr double GPS\_L1\_CA\_CODE\_LENGTH\_CHIPS = 1023.0
- GPS L1 C/A code length [chips].*
- constexpr double GPS\_L1\_CA\_CODE\_PERIOD\_S = 0.001
- GPS L1 C/A code period [seconds].*
- constexpr double GPS\_L1\_CA\_CHIP\_PERIOD\_S = 9.7752e-07
- GPS L1 C/A chip period [seconds].*
- constexpr uint32\_t GPS\_L1\_CA\_CODE\_PERIOD\_MS = 1U
- GPS L1 C/A code period [ms].*
- constexpr uint32\_t GPS\_L1\_CA\_BIT\_PERIOD\_MS = 20U
- GPS L1 C/A bit period [ms].*
- constexpr double MAX\_TOA\_DELAY\_MS = 20.0
- Maximum Time-Of-Arrival (TOA) difference between satellites for a receiver operated on Earth surface is 20 ms.*
- constexpr uint32\_t GPS\_L1\_CA\_OPT\_ACQ\_FS\_SPS = 2000000
- Sampling frequency that maximizes the acquisition SNR while using a non-multiple of chip rate.*
- constexpr int32\_t GPS\_L1\_CA\_HISTORY\_DEEP = 100
  - constexpr double GPS\_CA\_PREAMBLE\_DURATION\_S = 0.160
  - constexpr int32\_t GPS\_CA\_PREAMBLE\_LENGTH\_BITS = 8
  - constexpr int32\_t GPS\_CA\_PREAMBLE\_LENGTH\_SYMBOLS = 160
  - constexpr int32\_t GPS\_CA\_PREAMBLE\_DURATION\_MS = 160
  - constexpr int32\_t GPS\_CA\_TELEMETRY\_RATE\_BITS\_SECOND = 50
- NAV message bit rate [bits/s].*





- constexpr double `WN_LSF_LSB` = 1
- constexpr double `DN_LSB` = 1
- constexpr double `DELTAT_LSF_LSB` = 1
- constexpr int32\_t `T_OA_LSB` = `TWO_P12`
- constexpr double `ALM_ECC_LSB` = `TWO_N21`
- constexpr int32\_t `ALM_TOA_LSB` = `static_cast<int32_t>(TWO_P12)`
- constexpr double `ALM_DELTAI_LSB` = `TWO_N19`
- constexpr double `ALM_OMEGADOT_LSB` = `TWO_N38`
- constexpr double `ALM_SQUAREA_LSB` = `TWO_N11`
- constexpr double `ALM_OMEGAZERO_LSB` = `TWO_N23`
- constexpr double `ALM_OMEGA_LSB` = `TWO_N23`
- constexpr double `ALM_MZERO_LSB` = `TWO_N23`
- constexpr double `ALM_AF0_LSB` = `TWO_N20`
- constexpr double `ALM_AF1_LSB` = `TWO_N38`
- constexpr double `GPS_L2_FREQ_HZ` = `FREQ2`  
*L2 [Hz].*
- constexpr double `GPS_L2_L_PERIOD_S` = 1.5  
*GPS L2 L code period [seconds].*
- constexpr double `GPS_L2_M_CODE_RATE_CPS` = 0.5115e6  
*GPS L2 M code rate [chips/s].*
- constexpr double `GPS_L2_M_PERIOD_S` = 0.02  
*GPS L2 M code period [seconds].*
- constexpr double `GPS_L2_L_CODE_RATE_CPS` = 0.5115e6  
*GPS L2 L code rate [chips/s].*
- constexpr int32\_t `GPS_L2_M_CODE_LENGTH_CHIPS` = 10230  
*GPS L2 M code length [chips].*
- constexpr int32\_t `GPS_L2_L_CODE_LENGTH_CHIPS` = 767250  
*GPS L2 L code length [chips].*
- constexpr int32\_t `GPS_L2_CNAV_DATA_PAGE_BITS` = 300  
*GPS L2 CNAV page length, including preamble and CRC [bits].*
- constexpr int32\_t `GPS_L2_SYMBOLS_PER_BIT` = 2
- constexpr int32\_t `GPS_L2_SAMPLES_PER_SYMBOL` = 1
- constexpr int32\_t `GPS_L2_CNAV_DATA_PAGE_SYMBOLS` = 600
- constexpr int32\_t `GPS_L2_CNAV_DATA_PAGE_DURATION_S` = 12
- constexpr int32\_t `GPS_L2C_HISTORY_DEEP` = 5
- constexpr uint32\_t `GPS_L2C_OPT_ACQ_FS_SPS` = 2000000  
*Sampling frequency that maximizes the acquisition SNR while using a non-multiple of chip rate.*
- constexpr int32\_t `GPS_L2C_M_INIT_REG` [115]
- constexpr double `GPS_L5_FREQ_HZ` = `FREQ5`  
*L5 [Hz].*
- constexpr double `GPS_L5I_CODE_RATE_CPS` = 10.23e6  
*GPS L5I code rate [chips/s].*
- constexpr double `GPS_L5I_PERIOD_S` = 0.001  
*GPS L5I code period [seconds].*
- constexpr double `GPS_L5I_SYMBOL_PERIOD_S` = 0.01  
*GPS L5I symbol period [seconds].*
- constexpr double `GPS_L5Q_CODE_RATE_CPS` = 10.23e6  
*GPS L5Q code rate [chips/s].*
- constexpr double `GPS_L5Q_PERIOD_S` = 0.001  
*GPS L5Q code period [seconds].*
- constexpr int32\_t `GPS_L5Q_CODE_LENGTH_CHIPS` = 10230  
*GPS L5Q code length [chips].*



- constexpr int32\_t `GPS_L5I_CODE_LENGTH_CHIPS` = 10230  
*GPS L5I code length [chips].*
- constexpr int32\_t `GPS_L5I_PERIOD_MS` = 1  
*GPS L5I code period [ms].*
- constexpr int32\_t `GPS_L5I_SYMBOL_PERIOD_MS` = 10  
*GPS L5I symbol period [ms].*
- constexpr int32\_t `GPS_L5_HISTORY_DEEP` = 5
- constexpr uint32\_t `GPS_L5_OPT_ACQ_FS_SPS` = 10000000  
*Sampling frequency that maximizes the acquisition SNR while using a non-multiple of chip rate.*
- constexpr int32\_t `GPS_L5I_INIT_REG` [210]
- constexpr int32\_t `GPS_L5Q_INIT_REG` [210]
- constexpr int32\_t `GPS_L5_CNAV_DATA_PAGE_BITS` = 300  
*GPS L5 CNAV page length, including preamble and CRC [bits].*
- constexpr int32\_t `GPS_L5_SYMBOLS_PER_BIT` = 2
- constexpr int32\_t `GPS_L5_SAMPLES_PER_SYMBOL` = 10
- constexpr int32\_t `GPS_L5_CNAV_DATA_PAGE_SYMBOLS` = 600
- constexpr int32\_t `GPS_L5_CNAV_DATA_PAGE_DURATION_S` = 6
- constexpr int32\_t `GPS_L5I_NH_CODE_LENGTH` = 10
- constexpr int32\_t `GPS_L5I_NH_CODE` [10] = {0, 0, 0, 0, 1, 1, 0, 1, 0, 1}
- constexpr int32\_t `GPS_L5Q_NH_CODE_LENGTH` = 20
- constexpr int32\_t `GPS_L5Q_NH_CODE` [20] = {0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0}
- constexpr char `GPS_L5I_NH_CODE_STR` [11] = "0000110101"
- constexpr char `GPS_L5Q_NH_CODE_STR` [21] = "00000100110101001110"
- constexpr double `GNSS_OMEGA_EARTH_DOT` = 7.2921151467e-5  
*Default Earth rotation rate, [rad/s].*
- constexpr double `SPEED_OF_LIGHT_M_S` = 299792458.0  
*Speed of light in vacuum [m/s].*
- constexpr double `SPEED_OF_LIGHT_M_MS` = 299792.4580  
*Speed of light in vacuum [m/ms].*
- constexpr double `GPS_GM` = 3.986005e14  
*Universal gravitational constant times the mass of the Earth, [ $m^3/s^2$ ] IS-GPS-200M, 20.3.3.3.3.1.*
- constexpr double `GPS_F` = -4.442807633e-10  
*Constant, [ $s/(m)^{1/2}$ ], IS-GPS-200M, 20.3.3.3.3.1.*
- constexpr double `GALILEO_GM` = 3.986004418e14  
*Geocentric gravitational constant [ $m^3/s^2$ ], OS SIS ICD v2.0, pag. 44.*
- constexpr double `GALILEO_F` = -4.442807309e-10  
*Constant, [ $s/(m)^{1/2}$ ], OS SIS ICD v2.0, pag. 47.*
- constexpr double `GLONASS_OMEGA_EARTH_DOT` = 7.292115e-5  
*Earth rotation rate, [rad/s] ICD L1, L2 GLONASS Edition 5.1 2008 pag. 55.*
- constexpr double `GLONASS_GM` = 398600.44e9  
*Universal gravitational constant times the mass of the Earth, [ $m^3/s^2$ ].*
- constexpr double `BEIDOU_OMEGA_EARTH_DOT` = 7.2921150e-5  
*Earth rotation rate, [rad/s] as defined in BDS-SIS-ICD-B1I-3.0 2019-02, pag. 3.*
- constexpr double `BEIDOU_GM` = 3.986004418e14  
*Universal gravitational constant times the mass of the Earth, [ $m^3/s^2$ ] as defined in CGCS2000.*
- constexpr double `BEIDOU_F` = -4.442807309e-10  
*Constant, [ $s/(m)^{1/2}$ ]  $F = -2(GM)^{.5}/C^2$ .*
- constexpr double `GNSS_PI` = 3.1415926535898  
*pi constant as defined for GNSS*
- constexpr double `HALF_PI` = `GNSS_PI` / 2.0  
*pi/2*
- constexpr double `TWO_PI` = 2.0 \* `GNSS_PI`

- $2 * \pi$
- constexpr double [TWO\\_P3](#) = 8.0
- $2^3$
- constexpr double [TWO\\_P4](#) = 16.0
- $2^4$
- constexpr double [TWO\\_P8](#) = 256.0
- $2^8$
- constexpr double [TWO\\_P11](#) = 2048.0
- $2^{11}$
- constexpr double [TWO\\_P12](#) = 4096.0
- $2^{12}$
- constexpr double [TWO\\_P14](#) = 16384.0
- $2^{14}$
- constexpr double [TWO\\_P16](#) = 65536.0
- $2^{16}$
- constexpr double [TWO\\_P19](#) = 524288.0
- $2^{19}$
- constexpr double [TWO\\_P31](#) = 2147483648.0
- $2^{31}$
- constexpr double [TWO\\_P32](#) = 4294967296.0
- $2^{32}$
- constexpr double [TWO\\_P56](#) = 7.205759403792794e+016
- $2^{56}$
- constexpr double [TWO\\_P57](#) = 1.441151880758559e+017
- $2^{57}$
- constexpr double [TWO\\_N2](#) = 0.25
- $2^{-2}$
- constexpr double [TWO\\_N5](#) = 0.03125
- $2^{-5}$
- constexpr double [TWO\\_N6](#) = 0.015625
- $2^{-6}$
- constexpr double [TWO\\_N8](#) = 0.00390625
- $2^{-8}$
- constexpr double [TWO\\_N9](#) = 0.001953125
- $2^{-9}$
- constexpr double [TWO\\_N10](#) = 0.0009765625
- $2^{-10}$
- constexpr double [TWO\\_N11](#) = 4.882812500000000e-004
- $2^{-11}$
- constexpr double [TWO\\_N14](#) = 0.00006103515625
- $2^{-14}$
- constexpr double [TWO\\_N15](#) = 3.051757812500000e-005
- $2^{-15}$
- constexpr double [TWO\\_N16](#) = 1.525878906250000e-005
- $2^{-16}$
- constexpr double [TWO\\_N17](#) = 7.629394531250000e-006
- $2^{-17}$
- constexpr double [TWO\\_N18](#) = 3.814697265625000e-006
- $2^{-18}$
- constexpr double [TWO\\_N19](#) = 1.907348632812500e-006
- $2^{-19}$

- constexpr double TWO\_N20 = 9.536743164062500e-007  
 $2^{-20}$
- constexpr double TWO\_N21 = 4.768371582031250e-007  
 $2^{-21}$
- constexpr double TWO\_N22 = 2.384185791015625e-007  
 $2^{-22}$
- constexpr double TWO\_N23 = 1.192092895507810e-007  
 $2^{-23}$
- constexpr double TWO\_N24 = 5.960464477539063e-008  
 $2^{-24}$
- constexpr double TWO\_N25 = 2.980232238769531e-008  
 $2^{-25}$
- constexpr double TWO\_N26 = 1.490116119384765e-009  
 $2^{-26}$
- constexpr double TWO\_N27 = 7.450580596923828e-009  
 $2^{-27}$
- constexpr double TWO\_N29 = 1.862645149230957e-009  
 $2^{-29}$
- constexpr double TWO\_N30 = 9.313225746154785e-010  
 $2^{-30}$
- constexpr double TWO\_N31 = 4.656612873077393e-010  
 $2^{-31}$
- constexpr double TWO\_N32 = 2.328306436538696e-010  
 $2^{-32}$
- constexpr double TWO\_N33 = 1.164153218269348e-010  
 $2^{-33}$
- constexpr double TWO\_N34 = 5.82076609134674e-011  
 $2^{-34}$
- constexpr double TWO\_N35 = 2.91038304567337e-011  
 $2^{-35}$
- constexpr double TWO\_N38 = 3.637978807091713e-012  
 $2^{-38}$
- constexpr double TWO\_N39 = 1.818989403545856e-012  
 $2^{-39}$
- constexpr double TWO\_N40 = 9.094947017729280e-013  
 $2^{-40}$
- constexpr double TWO\_N43 = 1.136868377216160e-013  
 $2^{-43}$
- constexpr double TWO\_N44 = 5.684341886080802e-14  
 $2^{-44}$
- constexpr double TWO\_N46 = 1.4210854715202e-014  
 $2^{-46}$
- constexpr double TWO\_N48 = 3.552713678800501e-15  
 $2^{-46}$
- constexpr double TWO\_N50 = 8.881784197001252e-016  
 $2^{-50}$
- constexpr double TWO\_N51 = 4.44089209850063e-016  
 $2^{-51}$
- constexpr double TWO\_N55 = 2.775557561562891e-017  
 $2^{-55}$
- constexpr double TWO\_N57 = 6.938893903907228e-18

- $2^{-57}$
- constexpr double [TWO\\_N59](#) = 1.73472347597681e-018
- $2^{-59}$
- constexpr double [TWO\\_N60](#) = 8.673617379884036e-19
- $2^{-60}$
- constexpr double [TWO\\_N66](#) = 1.3552527156068805425093160010874271392822265625e-20
- $2^{-66}$
- constexpr double [TWO\\_N68](#) = 3.388131789017201e-21
- $2^{-68}$
- constexpr double [PI\\_TWO\\_N19](#) = 5.992112452678286e-006
- $PI * 2^{-19}$ .
- constexpr double [PI\\_TWO\\_N43](#) = 3.571577341960839e-013
- $PI * 2^{-43}$ .
- constexpr double [PI\\_TWO\\_N31](#) = 1.462918079267160e-009
- $PI * 2^{-31}$ .
- constexpr double [PI\\_TWO\\_N38](#) = 1.142904749427469e-011
- $PI * 2^{-38}$ .
- constexpr double [PI\\_TWO\\_N23](#) = 3.745070282923929e-007
- $PI * 2^{-23}$ .
- constexpr double [D2R](#) = [GNSS\\_PI](#) / 180.0
- deg to rad*
- constexpr double [R2D](#) = 180.0 / [GNSS\\_PI](#)
- rad to deg*
- constexpr double [SC2RAD](#) = [GNSS\\_PI](#)
- semi-circle to radian (IS-GPS)*
- constexpr double [AS2R](#) = [D2R](#) / 3600.0
- arc sec to radian*
- constexpr double [AU](#) = 149597870691.0
- 1 Astronomical Unit AU (m) distance from Earth to the Sun.*

### 9.13.6.1 Detailed Description

Classes containing info about system parameters for the different GNSS.  
GNSS parameters

### 9.13.6.2 Macro Definition Documentation

#### 9.13.6.2.1 DISPLAY\_COLORS

```
#define DISPLAY_COLORS 1
```

Definition at line 29 of file [display.h](#).

#### 9.13.6.2.2 GLONASS\_GNAV\_PREAMBLE

```
#define GLONASS_GNAV_PREAMBLE
```

**Value:**

```
{
    1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0}
```

Definition at line 90 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

### 9.13.6.3 Function Documentation

#### 9.13.6.3.1 ALPHA\_0()

```
const std::vector< std::pair< int32_t, int32_t > > ALPHA_0 (
    {{69, 8}} )
```

**Todo** read all pages of subframe 4

#### 9.13.6.3.2 T\_OA()

```
const std::vector< std::pair< int32_t, int32_t > > T_OA (
    {{69, 8}} )
```

**Todo** read all pages of subframe 5

#### 9.13.6.4 Variable Documentation

##### 9.13.6.4.1 A0\_6\_LSB

```
double A0_6_LSB = TWO_N30 [constexpr]
Definition at line 145 of file Galileo_INAV.h.
```

##### 9.13.6.4.2 A1\_6\_LSB

```
double A1_6_LSB = TWO_N50 [constexpr]
Definition at line 147 of file Galileo_INAV.h.
```

##### 9.13.6.4.3 A\_0\_LSB

```
double A_0_LSB = TWO_N30 [constexpr]
Definition at line 171 of file GPS_L1_CA.h.
```

##### 9.13.6.4.4 A\_0G\_10\_LSB

```
double A_0G_10_LSB = TWO_N35 [constexpr]
Definition at line 242 of file Galileo_INAV.h.
```

##### 9.13.6.4.5 A\_1\_LSB

```
double A_1_LSB = TWO_N50 [constexpr]
Definition at line 169 of file GPS_L1_CA.h.
```

##### 9.13.6.4.6 A\_1\_LSB\_GAL

```
double A_1_LSB_GAL = TWO_N19 [constexpr]
Definition at line 63 of file Galileo_INAV.h.
```

##### 9.13.6.4.7 A\_1G\_10\_LSB

```
double A_1G_10_LSB = TWO_N51 [constexpr]
Definition at line 244 of file Galileo_INAV.h.
```

##### 9.13.6.4.8 A\_F0\_LSB

```
double A_F0_LSB = TWO_N31 [constexpr]
Definition at line 102 of file GPS_L1_CA.h.
```

##### 9.13.6.4.9 A\_F1\_LSB

```
double A_F1_LSB = TWO_N43 [constexpr]
Definition at line 100 of file GPS_L1_CA.h.
```

##### 9.13.6.4.10 A\_F2\_LSB

```
double A_F2_LSB = TWO_N55 [constexpr]
Definition at line 98 of file GPS_L1_CA.h.
```

#### 9.13.6.4.11 AF0\_10\_LSB

double AF0\_10\_LSB = TWO\_N19 [constexpr]  
Definition at line 236 of file [Galileo\\_INAV.h](#).

#### 9.13.6.4.12 AF0\_4\_LSB

double AF0\_4\_LSB = TWO\_N34 [constexpr]  
Definition at line 106 of file [Galileo\\_INAV.h](#).

#### 9.13.6.4.13 AF0\_8\_LSB

double AF0\_8\_LSB = TWO\_N19 [constexpr]  
Definition at line 183 of file [Galileo\\_INAV.h](#).

#### 9.13.6.4.14 AF0\_9\_LSB

double AF0\_9\_LSB = TWO\_N19 [constexpr]  
Definition at line 211 of file [Galileo\\_INAV.h](#).

#### 9.13.6.4.15 AF1\_10\_LSB

double AF1\_10\_LSB = TWO\_N38 [constexpr]  
Definition at line 238 of file [Galileo\\_INAV.h](#).

#### 9.13.6.4.16 AF1\_4\_LSB

double AF1\_4\_LSB = TWO\_N46 [constexpr]  
Definition at line 108 of file [Galileo\\_INAV.h](#).

#### 9.13.6.4.17 AF1\_8\_LSB

double AF1\_8\_LSB = TWO\_N38 [constexpr]  
Definition at line 185 of file [Galileo\\_INAV.h](#).

#### 9.13.6.4.18 AF1\_9\_LSB

double AF1\_9\_LSB = TWO\_N38 [constexpr]  
Definition at line 213 of file [Galileo\\_INAV.h](#).

#### 9.13.6.4.19 AF2\_4\_LSB

double AF2\_4\_LSB = TWO\_N59 [constexpr]  
Definition at line 110 of file [Galileo\\_INAV.h](#).

#### 9.13.6.4.20 AI0\_5\_LSB

double AI0\_5\_LSB = TWO\_N2 [constexpr]  
Definition at line 118 of file [Galileo\\_INAV.h](#).

#### 9.13.6.4.21 AI1\_5\_LSB

double AI1\_5\_LSB = TWO\_N8 [constexpr]  
Definition at line 120 of file [Galileo\\_INAV.h](#).

#### 9.13.6.4.22 AI2\_5\_LSB

double AI2\_5\_LSB = TWO\_N15 [constexpr]  
Definition at line 122 of file [Galileo\\_INAV.h](#).

**9.13.6.4.23 ALM\_AF0\_LSB**

double ALM\_AF0\_LSB = TWO\_N20 [constexpr]  
Definition at line 249 of file GPS\_L1\_CA.h.

**9.13.6.4.24 ALM\_AF1\_LSB**

double ALM\_AF1\_LSB = TWO\_N38 [constexpr]  
Definition at line 251 of file GPS\_L1\_CA.h.

**9.13.6.4.25 ALM\_DELTAI\_LSB**

double ALM\_DELTAI\_LSB = TWO\_N19 [constexpr]  
Definition at line 235 of file GPS\_L1\_CA.h.

**9.13.6.4.26 ALM\_ECC\_LSB**

double ALM\_ECC\_LSB = TWO\_N21 [constexpr]  
Definition at line 231 of file GPS\_L1\_CA.h.

**9.13.6.4.27 ALM\_MZERO\_LSB**

double ALM\_MZERO\_LSB = TWO\_N23 [constexpr]  
Definition at line 246 of file GPS\_L1\_CA.h.

**9.13.6.4.28 ALM\_OMEGA\_LSB**

double ALM\_OMEGA\_LSB = TWO\_N23 [constexpr]  
Definition at line 244 of file GPS\_L1\_CA.h.

**9.13.6.4.29 ALM\_OMEGADOT\_LSB**

double ALM\_OMEGADOT\_LSB = TWO\_N38 [constexpr]  
Definition at line 237 of file GPS\_L1\_CA.h.

**9.13.6.4.30 ALM\_OMEGAZERO\_LSB**

double ALM\_OMEGAZERO\_LSB = TWO\_N23 [constexpr]  
Definition at line 242 of file GPS\_L1\_CA.h.

**9.13.6.4.31 ALM\_SQUAREA\_LSB**

double ALM\_SQUAREA\_LSB = TWO\_N11 [constexpr]  
Definition at line 240 of file GPS\_L1\_CA.h.

**9.13.6.4.32 ALM\_TOA\_LSB**

int32\_t ALM\_TOA\_LSB = static\_cast<int32\_t>(TWO\_P12) [constexpr]  
Definition at line 233 of file GPS\_L1\_CA.h.

**9.13.6.4.33 ALPHA\_0\_LSB**

double ALPHA\_0\_LSB = TWO\_N30 [constexpr]  
Definition at line 153 of file GPS\_L1\_CA.h.

**9.13.6.4.34 ALPHA\_1\_LSB**

double ALPHA\_1\_LSB = TWO\_N27 [constexpr]  
Definition at line 155 of file GPS\_L1\_CA.h.

#### 9.13.6.4.35 ALPHA\_2\_LSB

double ALPHA\_2\_LSB = TWO\_N24 [constexpr]  
Definition at line 157 of file GPS\_L1\_CA.h.

#### 9.13.6.4.36 ALPHA\_3\_LSB

double ALPHA\_3\_LSB = TWO\_N24 [constexpr]  
Definition at line 159 of file GPS\_L1\_CA.h.

#### 9.13.6.4.37 AODO\_LSB

int32\_t AODO\_LSB = 900 [constexpr]  
Definition at line 124 of file GPS\_L1\_CA.h.

#### 9.13.6.4.38 AS2R

double AS2R = D2R / 3600.0 [constexpr]  
arc sec to radian  
Definition at line 126 of file MATH\_CONSTANTS.h.

#### 9.13.6.4.39 AU

double AU = 149597870691.0 [constexpr]  
1 Astronomical Unit AU (m) distance from Earth to the Sun.  
Definition at line 128 of file MATH\_CONSTANTS.h.

#### 9.13.6.4.40 BEIDOU\_B1I\_CODE\_LENGTH\_CHIPS

double BEIDOU\_B1I\_CODE\_LENGTH\_CHIPS = 2046.0 [constexpr]  
Beidou B1I code length [chips].  
Definition at line 34 of file Beidou\_B1I.h.

#### 9.13.6.4.41 BEIDOU\_B1I\_CODE\_PERIOD\_MS

uint32\_t BEIDOU\_B1I\_CODE\_PERIOD\_MS = 1 [constexpr]  
Beidou B1I code period [ms].  
Definition at line 37 of file Beidou\_B1I.h.

#### 9.13.6.4.42 BEIDOU\_B1I\_CODE\_PERIOD\_S

double BEIDOU\_B1I\_CODE\_PERIOD\_S = 0.001 [constexpr]  
Beidou B1I code period [seconds].  
Definition at line 35 of file Beidou\_B1I.h.

#### 9.13.6.4.43 BEIDOU\_B1I\_CODE\_RATE\_CPS

double BEIDOU\_B1I\_CODE\_RATE\_CPS = 2.046e6 [constexpr]  
Beidou B1I code rate [chips/s].  
Definition at line 33 of file Beidou\_B1I.h.

#### 9.13.6.4.44 BEIDOU\_B1I\_D2\_SECONDARY\_CODE\_STR

char BEIDOU\_B1I\_D2\_SECONDARY\_CODE\_STR[3] = "00" [constexpr]  
Definition at line 50 of file Beidou\_B1I.h.

#### 9.13.6.4.45 BEIDOU\_B1I\_FREQ\_HZ

double BEIDOU\_B1I\_FREQ\_HZ = FREQ1\_BDS [constexpr]  
B1I [Hz].  
Definition at line 32 of file Beidou\_B1I.h.



**9.13.6.4.46 BEIDOU\_B1I\_GEO\_PREAMBLE\_LENGTH\_SYMBOLS**

```
int32_t BEIDOU_B1I_GEO_PREAMBLE_LENGTH_SYMBOLS = 22 [constexpr]
```

Definition at line 41 of file [Beidou\\_B1I.h](#).

**9.13.6.4.47 BEIDOU\_B1I\_GEO\_PREAMBLE\_SYMBOLS\_STR**

```
char BEIDOU_B1I_GEO_PREAMBLE_SYMBOLS_STR[23] = "1111110000001100001100" [constexpr]
```

Definition at line 49 of file [Beidou\\_B1I.h](#).

**9.13.6.4.48 BEIDOU\_B1I\_GEO\_TELEMETRY\_SYMBOLS\_PER\_BIT**

```
int32_t BEIDOU_B1I_GEO_TELEMETRY_SYMBOLS_PER_BIT = 2 [constexpr]
```

Definition at line 45 of file [Beidou\\_B1I.h](#).

**9.13.6.4.49 BEIDOU\_B1I\_PREAMBLE\_DURATION\_MS**

```
int32_t BEIDOU_B1I_PREAMBLE_DURATION_MS = 220 [constexpr]
```

Definition at line 42 of file [Beidou\\_B1I.h](#).

**9.13.6.4.50 BEIDOU\_B1I\_PREAMBLE\_DURATION\_S**

```
double BEIDOU_B1I_PREAMBLE_DURATION_S = 0.220 [constexpr]
```

Definition at line 36 of file [Beidou\\_B1I.h](#).

**9.13.6.4.51 BEIDOU\_B1I\_PREAMBLE\_LENGTH\_BITS**

```
uint32_t BEIDOU_B1I_PREAMBLE_LENGTH_BITS = 11 [constexpr]
```

Definition at line 38 of file [Beidou\\_B1I.h](#).

**9.13.6.4.52 BEIDOU\_B1I\_PREAMBLE\_LENGTH\_SYMBOLS**

```
uint32_t BEIDOU_B1I_PREAMBLE_LENGTH_SYMBOLS = 220 [constexpr]
```

Definition at line 39 of file [Beidou\\_B1I.h](#).

**9.13.6.4.53 BEIDOU\_B1I\_SECONDARY\_CODE\_LENGTH**

```
int32_t BEIDOU_B1I_SECONDARY_CODE_LENGTH = 20 [constexpr]
```

Definition at line 40 of file [Beidou\\_B1I.h](#).

**9.13.6.4.54 BEIDOU\_B1I\_SECONDARY\_CODE\_STR**

```
char BEIDOU_B1I_SECONDARY_CODE_STR[21] = "00000100110101001110" [constexpr]
```

Definition at line 48 of file [Beidou\\_B1I.h](#).

**9.13.6.4.55 BEIDOU\_B1I\_TELEMETRY\_RATE\_BITS\_SECOND**

```
int32_t BEIDOU_B1I_TELEMETRY_RATE_BITS_SECOND = 50 [constexpr]
```

Definition at line 43 of file [Beidou\\_B1I.h](#).

**9.13.6.4.56 BEIDOU\_B1I\_TELEMETRY\_RATE\_SYMBOLS\_SECOND**

```
int32_t BEIDOU_B1I_TELEMETRY_RATE_SYMBOLS_SECOND = BEIDOU_B1I_TELEMETRY_RATE_BITS_SECOND *  
BEIDOU_B1I_TELEMETRY_SYMBOLS_PER_BIT [constexpr]
```

Definition at line 47 of file [Beidou\\_B1I.h](#).

**9.13.6.4.57 BEIDOU\_B1I\_TELEMETRY\_SYMBOL\_PERIOD\_MS**

```
int32_t BEIDOU_B1I_TELEMETRY_SYMBOL_PERIOD_MS = static_cast<int32_t>(static_cast<uint32_t>  
<>(BEIDOU_B1I_TELEMETRY_SYMBOLS_PER_BIT) * BEIDOU_B1I_CODE_PERIOD_MS) [constexpr]
```

Definition at line 46 of file [Beidou\\_B1I.h](#).

#### 9.13.6.4.58 BEIDOU\_B1I\_TELEMETRY\_SYMBOLS\_PER\_BIT

`int32_t BEIDOU_B1I_TELEMETRY_SYMBOLS_PER_BIT = 20 [constexpr]`  
Definition at line 44 of file [Beidou\\_B1I.h](#).

#### 9.13.6.4.59 BEIDOU\_B3I\_CODE\_LENGTH\_CHIPS

`double BEIDOU_B3I_CODE_LENGTH_CHIPS = 10230.0 [constexpr]`  
BeiDou B3I code length [chips].  
Definition at line 33 of file [Beidou\\_B3I.h](#).

#### 9.13.6.4.60 BEIDOU\_B3I\_CODE\_PERIOD\_MS

`uint32_t BEIDOU_B3I_CODE_PERIOD_MS = 1 [constexpr]`  
BeiDou B3I code period [ms].  
Definition at line 36 of file [Beidou\\_B3I.h](#).

#### 9.13.6.4.61 BEIDOU\_B3I\_CODE\_PERIOD\_S

`double BEIDOU_B3I_CODE_PERIOD_S = 0.001 [constexpr]`  
BeiDou B3I code period [seconds].  
Definition at line 34 of file [Beidou\\_B3I.h](#).

#### 9.13.6.4.62 BEIDOU\_B3I\_CODE\_RATE\_CPS

`double BEIDOU_B3I_CODE_RATE_CPS = 10.23e6 [constexpr]`  
BeiDou B3I code rate [chips/s].  
Definition at line 32 of file [Beidou\\_B3I.h](#).

#### 9.13.6.4.63 BEIDOU\_B3I\_D2\_SECONDARY\_CODE\_STR

`char BEIDOU_B3I_D2_SECONDARY_CODE_STR[3] = "00" [constexpr]`  
Definition at line 49 of file [Beidou\\_B3I.h](#).

#### 9.13.6.4.64 BEIDOU\_B3I\_FREQ\_HZ

`double BEIDOU_B3I_FREQ_HZ = FREQ3\_BDS [constexpr]`  
BeiDou B3I [Hz].  
Definition at line 31 of file [Beidou\\_B3I.h](#).

#### 9.13.6.4.65 BEIDOU\_B3I\_GEO\_PREAMBLE\_LENGTH\_SYMBOLS

`int32_t BEIDOU_B3I_GEO_PREAMBLE_LENGTH_SYMBOLS = 22 [constexpr]`  
Definition at line 40 of file [Beidou\\_B3I.h](#).

#### 9.13.6.4.66 BEIDOU\_B3I\_GEO\_PREAMBLE\_SYMBOLS\_STR

`char BEIDOU_B3I_GEO_PREAMBLE_SYMBOLS_STR[23] = "1111110000001100001100" [constexpr]`  
Definition at line 48 of file [Beidou\\_B3I.h](#).

#### 9.13.6.4.67 BEIDOU\_B3I\_GEO\_TELEMETRY\_SYMBOLS\_PER\_BIT

`int32_t BEIDOU_B3I_GEO_TELEMETRY_SYMBOLS_PER_BIT = 2 [constexpr]`  
Definition at line 44 of file [Beidou\\_B3I.h](#).

#### 9.13.6.4.68 BEIDOU\_B3I\_PREAMBLE\_DURATION\_MS

`int32_t BEIDOU_B3I_PREAMBLE_DURATION_MS = 220 [constexpr]`  
Definition at line 41 of file [Beidou\\_B3I.h](#).

**9.13.6.4.69 BEIDOU\_B3I\_PREAMBLE\_DURATION\_S**

```
double BEIDOU_B3I_PREAMBLE_DURATION_S = 0.220 [constexpr]
```

Definition at line 35 of file [Beidou\\_B3I.h](#).

**9.13.6.4.70 BEIDOU\_B3I\_PREAMBLE\_LENGTH\_BITS**

```
uint32_t BEIDOU_B3I_PREAMBLE_LENGTH_BITS = 11 [constexpr]
```

Definition at line 37 of file [Beidou\\_B3I.h](#).

**9.13.6.4.71 BEIDOU\_B3I\_PREAMBLE\_LENGTH\_SYMBOLS**

```
uint32_t BEIDOU_B3I_PREAMBLE_LENGTH_SYMBOLS = 220 [constexpr]
```

Definition at line 38 of file [Beidou\\_B3I.h](#).

**9.13.6.4.72 BEIDOU\_B3I\_SECONDARY\_CODE\_LENGTH**

```
int32_t BEIDOU_B3I_SECONDARY_CODE_LENGTH = 20 [constexpr]
```

Definition at line 39 of file [Beidou\\_B3I.h](#).

**9.13.6.4.73 BEIDOU\_B3I\_SECONDARY\_CODE\_STR**

```
char BEIDOU_B3I_SECONDARY_CODE_STR[21] = "00000100110101001110" [constexpr]
```

Definition at line 47 of file [Beidou\\_B3I.h](#).

**9.13.6.4.74 BEIDOU\_B3I\_TELEMETRY\_RATE\_BITS\_SECOND**

```
int32_t BEIDOU_B3I_TELEMETRY_RATE_BITS_SECOND = 50 [constexpr]
```

D1 NAV message bit rate [bits/s].

Definition at line 42 of file [Beidou\\_B3I.h](#).

**9.13.6.4.75 BEIDOU\_B3I\_TELEMETRY\_RATE\_SYMBOLS\_SECOND**

```
int32_t BEIDOU_B3I_TELEMETRY_RATE_SYMBOLS_SECOND = BEIDOU_B3I_TELEMETRY_RATE_BITS_SECOND *  
BEIDOU_B3I_TELEMETRY_SYMBOLS_PER_BIT [constexpr]
```

Definition at line 46 of file [Beidou\\_B3I.h](#).

**9.13.6.4.76 BEIDOU\_B3I\_TELEMETRY\_SYMBOL\_PERIOD\_MS**

```
int32_t BEIDOU_B3I_TELEMETRY_SYMBOL_PERIOD_MS = static_cast<int32_t>(static_cast<uint32_t>(  
t>(BEIDOU_B3I_TELEMETRY_SYMBOLS_PER_BIT) * BEIDOU_B3I_CODE_PERIOD_MS) [constexpr]
```

Definition at line 45 of file [Beidou\\_B3I.h](#).

**9.13.6.4.77 BEIDOU\_B3I\_TELEMETRY\_SYMBOLS\_PER\_BIT**

```
int32_t BEIDOU_B3I_TELEMETRY_SYMBOLS_PER_BIT = 20 [constexpr]
```

Definition at line 43 of file [Beidou\\_B3I.h](#).

**9.13.6.4.78 BEIDOU\_DNAV\_BDT2GPST\_LEAP\_SEC\_OFFSET**

```
int32_t BEIDOU_DNAV_BDT2GPST_LEAP_SEC_OFFSET = 14 [constexpr]
```

Definition at line 89 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.79 BEIDOU\_DNAV\_BDT2GPST\_WEEK\_NUM\_OFFSET**

```
int32_t BEIDOU_DNAV_BDT2GPST_WEEK_NUM_OFFSET = 1356 [constexpr]
```

Definition at line 91 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.80 BEIDOU\_DNAV\_PREAMBLE**

`char BEIDOU_DNAV_PREAMBLE[12] = "11100010010" [constexpr]`  
 Definition at line 96 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.81 BEIDOU\_DNAV\_PREAMBLE\_LENGTH\_BITS**

`int32_t BEIDOU_DNAV_PREAMBLE_LENGTH_BITS = 11 [constexpr]`  
 Definition at line 84 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.82 BEIDOU\_DNAV\_PREAMBLE\_LENGTH\_SYMBOLS**

`int32_t BEIDOU_DNAV_PREAMBLE_LENGTH_SYMBOLS = 11 [constexpr]`  
 Definition at line 85 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.83 BEIDOU\_DNAV\_PREAMBLE\_PERIOD\_SYMBOLS**

`int32_t BEIDOU_DNAV_PREAMBLE_PERIOD_SYMBOLS = 300 [constexpr]`  
 Definition at line 86 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.84 BEIDOU\_DNAV\_SUBFRAME\_DATA\_BITS**

`int32_t BEIDOU_DNAV_SUBFRAME_DATA_BITS = 300 [constexpr]`  
 Number of bits per subframe in the NAV message [bits].  
 Definition at line 87 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.85 BEIDOU\_DNAV\_SUBFRAME\_SYMBOLS**

`uint32_t BEIDOU_DNAV_SUBFRAME_SYMBOLS = 300 [constexpr]`  
 Definition at line 93 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.86 BEIDOU\_DNAV\_WORD\_LENGTH\_BITS**

`uint32_t BEIDOU_DNAV_WORD_LENGTH_BITS = 30 [constexpr]`  
 Definition at line 95 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.87 BEIDOU\_DNAV\_WORDS\_SUBFRAME**

`uint32_t BEIDOU_DNAV_WORDS_SUBFRAME = 10 [constexpr]`  
 Definition at line 94 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.88 BEIDOU\_F**

`double BEIDOU_F = -4.442807309e-10 [constexpr]`  
 Constant,  $[s/(m)^{(1/2)}] F = -2(GM)^{.5}/C^2$ .  
 Definition at line 45 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.89 BEIDOU\_GM**

`double BEIDOU_GM = 3.986004418e14 [constexpr]`  
 Universal gravitational constant times the mass of the Earth,  $[m^3/s^2]$  as defined in CGCS2000.  
 Definition at line 44 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.90 BEIDOU\_OMEGA\_EARTH\_DOT**

`double BEIDOU_OMEGA_EARTH_DOT = 7.2921150e-5 [constexpr]`  
 Earth rotation rate,  $[rad/s]$  as defined in BDS-SIS-ICD-B1I-3.0 2019-02, pag. 3.  
 Definition at line 43 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.91 BETA\_0\_LSB**

double BETA\_0\_LSB = TWO\_P11 [constexpr]  
Definition at line 161 of file GPS\_L1\_CA.h.

**9.13.6.4.92 BETA\_1\_LSB**

double BETA\_1\_LSB = TWO\_P14 [constexpr]  
Definition at line 163 of file GPS\_L1\_CA.h.

**9.13.6.4.93 BETA\_2\_LSB**

double BETA\_2\_LSB = TWO\_P16 [constexpr]  
Definition at line 165 of file GPS\_L1\_CA.h.

**9.13.6.4.94 BETA\_3\_LSB**

double BETA\_3\_LSB = TWO\_P16 [constexpr]  
Definition at line 167 of file GPS\_L1\_CA.h.

**9.13.6.4.95 BGD\_E1\_E5A\_5\_LSB**

double BGD\_E1\_E5A\_5\_LSB = TWO\_N32 [constexpr]  
Definition at line 130 of file Galileo\_INAV.h.

**9.13.6.4.96 BGD\_E1\_E5B\_5\_LSB**

double BGD\_E1\_E5B\_5\_LSB = TWO\_N32 [constexpr]  
Definition at line 132 of file Galileo\_INAV.h.

**9.13.6.4.97 BITS\_IN\_OCTET**

int32\_t BITS\_IN\_OCTET = 8 [constexpr]  
Definition at line 273 of file Galileo\_INAV.h.

**9.13.6.4.98 C\_IC\_4\_LSB**

double C\_IC\_4\_LSB = TWO\_N29 [constexpr]  
Definition at line 100 of file Galileo\_INAV.h.

**9.13.6.4.99 C\_IC\_LSB**

double C\_IC\_LSB = TWO\_N29 [constexpr]  
Definition at line 128 of file GPS\_L1\_CA.h.

**9.13.6.4.100 C\_IS\_4\_LSB**

double C\_IS\_4\_LSB = TWO\_N29 [constexpr]  
Definition at line 102 of file Galileo\_INAV.h.

**9.13.6.4.101 C\_IS\_LSB**

double C\_IS\_LSB = TWO\_N29 [constexpr]  
Definition at line 132 of file GPS\_L1\_CA.h.

**9.13.6.4.102 C\_RC\_3\_LSB**

double C\_RC\_3\_LSB = TWO\_N5 [constexpr]  
Definition at line 90 of file Galileo\_INAV.h.

#### 9.13.6.4.103 C\_RC\_LSB

double C\_RC\_LSB = TWO\_N5 [constexpr]  
Definition at line 136 of file [GPS\\_L1\\_CA.h](#).

#### 9.13.6.4.104 C\_RS\_3\_LSB

double C\_RS\_3\_LSB = TWO\_N5 [constexpr]  
Definition at line 92 of file [Galileo\\_INAV.h](#).

#### 9.13.6.4.105 C\_RS\_LSB

double C\_RS\_LSB = TWO\_N5 [constexpr]  
Definition at line 107 of file [GPS\\_L1\\_CA.h](#).

#### 9.13.6.4.106 C\_UC\_3\_LSB

double C\_UC\_3\_LSB = TWO\_N29 [constexpr]  
Definition at line 86 of file [Galileo\\_INAV.h](#).

#### 9.13.6.4.107 C\_UC\_LSB

double C\_UC\_LSB = TWO\_N29 [constexpr]  
Definition at line 113 of file [GPS\\_L1\\_CA.h](#).

#### 9.13.6.4.108 C\_US\_3\_LSB

double C\_US\_3\_LSB = TWO\_N29 [constexpr]  
Definition at line 88 of file [Galileo\\_INAV.h](#).

#### 9.13.6.4.109 C\_US\_LSB

double C\_US\_LSB = TWO\_N29 [constexpr]  
Definition at line 117 of file [GPS\\_L1\\_CA.h](#).

#### 9.13.6.4.110 CED\_af0red\_LSB

double CED\_af0red\_LSB = TWO\_N26 [constexpr]  
Definition at line 262 of file [Galileo\\_INAV.h](#).

#### 9.13.6.4.111 CED\_af1red\_LSB

double CED\_af1red\_LSB = TWO\_N35 [constexpr]  
Definition at line 264 of file [Galileo\\_INAV.h](#).

#### 9.13.6.4.112 CED\_DeltaAred\_LSB

double CED\_DeltaAred\_LSB = TWO\_P8 [constexpr]  
Definition at line 250 of file [Galileo\\_INAV.h](#).

#### 9.13.6.4.113 CED\_DeltaI0red\_LSB

double CED\_DeltaI0red\_LSB = TWO\_N22 [constexpr]  
Definition at line 256 of file [Galileo\\_INAV.h](#).

#### 9.13.6.4.114 CED\_exred\_LSB

double CED\_exred\_LSB = TWO\_N22 [constexpr]  
Definition at line 252 of file [Galileo\\_INAV.h](#).

**9.13.6.4.115 CED\_eyred\_LSB**

```
double CED_eyred_LSB = TWO_N22 [constexpr]
```

Definition at line 254 of file [Galileo\\_INAV.h](#).

**9.13.6.4.116 CED\_lambda0red\_LSB**

```
double CED_lambda0red_LSB = TWO_N22 [constexpr]
```

Definition at line 260 of file [Galileo\\_INAV.h](#).

**9.13.6.4.117 CED\_Omega0red\_LSB**

```
double CED_Omega0red_LSB = TWO_N22 [constexpr]
```

Definition at line 258 of file [Galileo\\_INAV.h](#).

**9.13.6.4.118 CNAV\_A0\_LSB**

```
double CNAV_A0_LSB = TWO_N35 [constexpr]
```

Definition at line 144 of file [GPS\\_CNAV.h](#).

**9.13.6.4.119 CNAV\_A1\_LSB**

```
double CNAV_A1_LSB = TWO_N51 [constexpr]
```

Definition at line 146 of file [GPS\\_CNAV.h](#).

**9.13.6.4.120 CNAV\_A2\_LSB**

```
double CNAV_A2_LSB = TWO_N68 [constexpr]
```

Definition at line 148 of file [GPS\\_CNAV.h](#).

**9.13.6.4.121 CNAV\_A\_DOT\_LSB**

```
double CNAV_A_DOT_LSB = TWO_N21 [constexpr]
```

Definition at line 58 of file [GPS\\_CNAV.h](#).

**9.13.6.4.122 CNAV\_A\_REF**

```
double CNAV_A_REF = 26559710.0 [constexpr]
```

Definition at line 162 of file [GPS\\_CNAV.h](#).

**9.13.6.4.123 CNAV\_AF0\_LSB**

```
double CNAV_AF0_LSB = TWO_N35 [constexpr]
```

Definition at line 107 of file [GPS\\_CNAV.h](#).

**9.13.6.4.124 CNAV\_AF1\_LSB**

```
double CNAV_AF1_LSB = TWO_N48 [constexpr]
```

Definition at line 109 of file [GPS\\_CNAV.h](#).

**9.13.6.4.125 CNAV\_AF2\_LSB**

```
double CNAV_AF2_LSB = TWO_N60 [constexpr]
```

Definition at line 111 of file [GPS\\_CNAV.h](#).

**9.13.6.4.126 CNAV\_ALPHA0\_LSB**

```
double CNAV_ALPHA0_LSB = TWO_N30 [constexpr]
```

Definition at line 124 of file [GPS\\_CNAV.h](#).

**9.13.6.4.127 CNAV\_ALPHA1\_LSB**

double CNAV\_ALPHA1\_LSB = TWO\_N27 [constexpr]  
Definition at line 126 of file [GPS\\_CNAV.h](#).

**9.13.6.4.128 CNAV\_ALPHA2\_LSB**

double CNAV\_ALPHA2\_LSB = TWO\_N24 [constexpr]  
Definition at line 128 of file [GPS\\_CNAV.h](#).

**9.13.6.4.129 CNAV\_ALPHA3\_LSB**

double CNAV\_ALPHA3\_LSB = TWO\_N24 [constexpr]  
Definition at line 130 of file [GPS\\_CNAV.h](#).

**9.13.6.4.130 CNAV\_BETA0\_LSB**

double CNAV\_BETA0\_LSB = TWO\_P11 [constexpr]  
Definition at line 132 of file [GPS\\_CNAV.h](#).

**9.13.6.4.131 CNAV\_BETA1\_LSB**

double CNAV\_BETA1\_LSB = TWO\_P14 [constexpr]  
Definition at line 134 of file [GPS\\_CNAV.h](#).

**9.13.6.4.132 CNAV\_BETA2\_LSB**

double CNAV\_BETA2\_LSB = TWO\_P16 [constexpr]  
Definition at line 136 of file [GPS\\_CNAV.h](#).

**9.13.6.4.133 CNAV\_BETA3\_LSB**

double CNAV\_BETA3\_LSB = TWO\_P16 [constexpr]  
Definition at line 138 of file [GPS\\_CNAV.h](#).

**9.13.6.4.134 CNAV\_CIC\_LSB**

double CNAV\_CIC\_LSB = TWO\_N30 [constexpr]  
Definition at line 87 of file [GPS\\_CNAV.h](#).

**9.13.6.4.135 CNAV\_CIS\_LSB**

double CNAV\_CIS\_LSB = TWO\_N30 [constexpr]  
Definition at line 85 of file [GPS\\_CNAV.h](#).

**9.13.6.4.136 CNAV\_CRC\_LSB**

double CNAV\_CRC\_LSB = TWO\_N8 [constexpr]  
Definition at line 91 of file [GPS\\_CNAV.h](#).

**9.13.6.4.137 CNAV\_CRS\_LSB**

double CNAV\_CRS\_LSB = TWO\_N8 [constexpr]  
Definition at line 89 of file [GPS\\_CNAV.h](#).

**9.13.6.4.138 CNAV\_CUC\_LSB**

double CNAV\_CUC\_LSB = TWO\_N30 [constexpr]  
Definition at line 95 of file [GPS\\_CNAV.h](#).



**9.13.6.4.139 CNAV\_CUS\_LSB**

double CNAV\_CUS\_LSB = TWO\_N30 [constexpr]  
Definition at line 93 of file [GPS\\_CNAV.h](#).

**9.13.6.4.140 CNAV\_DELTA\_A\_LSB**

double CNAV\_DELTA\_A\_LSB = TWO\_N9 [constexpr]  
Definition at line 55 of file [GPS\\_CNAV.h](#).

**9.13.6.4.141 CNAV\_DELTA\_N0\_DOT\_LSB**

double CNAV\_DELTA\_N0\_DOT\_LSB = TWO\_N57 \* GNSS\_PI [constexpr]  
Definition at line 63 of file [GPS\\_CNAV.h](#).

**9.13.6.4.142 CNAV\_DELTA\_N0\_LSB**

double CNAV\_DELTA\_N0\_LSB = TWO\_N44 \* GNSS\_PI [constexpr]  
Definition at line 61 of file [GPS\\_CNAV.h](#).

**9.13.6.4.143 CNAV\_DELTA\_OMEGA\_DOT\_LSB**

double CNAV\_DELTA\_OMEGA\_DOT\_LSB = TWO\_N44 \* GNSS\_PI [constexpr]  
Definition at line 81 of file [GPS\\_CNAV.h](#).

**9.13.6.4.144 CNAV\_DELTA\_TLS\_LSB**

int32\_t CNAV\_DELTA\_TLS\_LSB = 1 [constexpr]  
Definition at line 150 of file [GPS\\_CNAV.h](#).

**9.13.6.4.145 CNAV\_DELTA\_TLSF\_LSB**

int32\_t CNAV\_DELTA\_TLSF\_LSB = 1 [constexpr]  
Definition at line 160 of file [GPS\\_CNAV.h](#).

**9.13.6.4.146 CNAV\_DN\_LSB**

int32\_t CNAV\_DN\_LSB = 1 [constexpr]  
Definition at line 158 of file [GPS\\_CNAV.h](#).

**9.13.6.4.147 CNAV\_E\_ECCENTRICITY\_LSB**

double CNAV\_E\_ECCENTRICITY\_LSB = TWO\_N34 [constexpr]  
Definition at line 67 of file [GPS\\_CNAV.h](#).

**9.13.6.4.148 CNAV\_I0\_DOT\_LSB**

double CNAV\_I0\_DOT\_LSB = TWO\_N44 \* GNSS\_PI [constexpr]  
Definition at line 83 of file [GPS\\_CNAV.h](#).

**9.13.6.4.149 CNAV\_I0\_LSB**

double CNAV\_I0\_LSB = TWO\_N32 \* GNSS\_PI [constexpr]  
Definition at line 79 of file [GPS\\_CNAV.h](#).

**9.13.6.4.150 CNAV\_ISCL1\_LSB**

double CNAV\_ISCL1\_LSB = TWO\_N35 [constexpr]  
Definition at line 115 of file [GPS\\_CNAV.h](#).

**9.13.6.4.151 CNAV\_ISCL2\_LSB**

```
double CNAV_ISCL2_LSB = TWO_N35 [constexpr]
```

Definition at line 117 of file [GPS\\_CNAV.h](#).

**9.13.6.4.152 CNAV\_ISCL5I\_LSB**

```
double CNAV_ISCL5I_LSB = TWO_N35 [constexpr]
```

Definition at line 119 of file [GPS\\_CNAV.h](#).

**9.13.6.4.153 CNAV\_ISCL5Q\_LSB**

```
double CNAV_ISCL5Q_LSB = TWO_N35 [constexpr]
```

Definition at line 121 of file [GPS\\_CNAV.h](#).

**9.13.6.4.154 CNAV\_M0\_LSB**

```
double CNAV_M0_LSB = TWO_N32 * GNSS_PI [constexpr]
```

Definition at line 65 of file [GPS\\_CNAV.h](#).

**9.13.6.4.155 CNAV\_OMEGA0\_LSB**

```
double CNAV_OMEGA0_LSB = TWO_N32 * GNSS_PI [constexpr]
```

Definition at line 77 of file [GPS\\_CNAV.h](#).

**9.13.6.4.156 CNAV\_OMEGA\_DOT\_REF**

```
double CNAV_OMEGA_DOT_REF = -2.6e-9 [constexpr]
```

Definition at line 163 of file [GPS\\_CNAV.h](#).

**9.13.6.4.157 CNAV\_OMEGA\_LSB**

```
double CNAV_OMEGA_LSB = TWO_N32 * GNSS_PI [constexpr]
```

Definition at line 69 of file [GPS\\_CNAV.h](#).

**9.13.6.4.158 CNAV\_TGD\_LSB**

```
double CNAV_TGD_LSB = TWO_N35 [constexpr]
```

Definition at line 113 of file [GPS\\_CNAV.h](#).

**9.13.6.4.159 CNAV\_TOC\_LSB**

```
int32_t CNAV_TOC_LSB = 300 [constexpr]
```

Definition at line 105 of file [GPS\\_CNAV.h](#).

**9.13.6.4.160 CNAV\_TOE1\_LSB**

```
int32_t CNAV_TOE1_LSB = 300 [constexpr]
```

Definition at line 52 of file [GPS\\_CNAV.h](#).

**9.13.6.4.161 CNAV\_TOE2\_LSB**

```
int32_t CNAV_TOE2_LSB = 300 [constexpr]
```

Definition at line 75 of file [GPS\\_CNAV.h](#).

**9.13.6.4.162 CNAV\_TOP1\_LSB**

```
int32_t CNAV_TOP1_LSB = 300 [constexpr]
```

Definition at line 48 of file [GPS\\_CNAV.h](#).

**9.13.6.4.163 CNAV\_TOP2\_LSB**

`int32_t CNAV_TOP2_LSB = 300 [constexpr]`  
Definition at line 100 of file [GPS\\_CNAV.h](#).

**9.13.6.4.164 CNAV\_TOT\_LSB**

`int32_t CNAV_TOT_LSB = TWO_P4 [constexpr]`  
Definition at line 152 of file [GPS\\_CNAV.h](#).

**9.13.6.4.165 CNAV\_TOW\_LSB**

`int32_t CNAV_TOW_LSB = 6 [constexpr]`  
Definition at line 41 of file [GPS\\_CNAV.h](#).

**9.13.6.4.166 CNAV\_WN\_LSF\_LSB**

`int32_t CNAV_WN_LSF_LSB = 1 [constexpr]`  
Definition at line 156 of file [GPS\\_CNAV.h](#).

**9.13.6.4.167 CNAV\_WN\_OT\_LSB**

`int32_t CNAV_WN_OT_LSB = 1 [constexpr]`  
Definition at line 154 of file [GPS\\_CNAV.h](#).

**9.13.6.4.168 CODE\_L1A**

`uint32_t CODE_L1A = 10 [constexpr]`  
obs code: E1A (GAL)  
Definition at line 40 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.169 CODE\_L1B**

`uint32_t CODE_L1B = 11 [constexpr]`  
obs code: E1B (GAL)  
Definition at line 41 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.170 CODE\_L1C**

`uint32_t CODE_L1C = 1 [constexpr]`  
obs code: L1C/A,G1C/A,E1C (GPS,GLO,GAL,QZS,SBS)  
Definition at line 31 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.171 CODE\_L1E**

`uint32_t CODE_L1E = 9 [constexpr]`  
(not used)  
Definition at line 39 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.172 CODE\_L1I**

`uint32_t CODE_L1I = 47 [constexpr]`  
obs code: B1I (BDS)  
Definition at line 77 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.173 CODE\_L1L**

`uint32_t CODE_L1L = 8 [constexpr]`  
obs code: L1C(P) (GPS,QZS)  
Definition at line 38 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.174 CODE\_L1M**

```
uint32_t CODE_L1M = 5 [constexpr]
```

obs code: L1M (GPS)  
Definition at line 35 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.175 CODE\_L1N**

```
uint32_t CODE_L1N = 6 [constexpr]
```

obs code: L1codeless (GPS)  
Definition at line 36 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.176 CODE\_L1P**

```
uint32_t CODE_L1P = 2 [constexpr]
```

obs code: L1P,G1P (GPS,GLO)  
Definition at line 32 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.177 CODE\_L1Q**

```
uint32_t CODE_L1Q = 48 [constexpr]
```

obs code: B1Q (BDS)  
Definition at line 78 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.178 CODE\_L1S**

```
uint32_t CODE_L1S = 7 [constexpr]
```

obs code: L1C(D) (GPS,QZS)  
Definition at line 37 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.179 CODE\_L1W**

```
uint32_t CODE_L1W = 3 [constexpr]
```

obs code: L1 Z-track (GPS)  
Definition at line 33 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.180 CODE\_L1X**

```
uint32_t CODE_L1X = 12 [constexpr]
```

obs code: E1B+C,L1C(D+P) (GAL,QZS)  
Definition at line 42 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.181 CODE\_L1Y**

```
uint32_t CODE_L1Y = 4 [constexpr]
```

obs code: L1Y (GPS)  
Definition at line 34 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.182 CODE\_L1Z**

```
uint32_t CODE_L1Z = 13 [constexpr]
```

obs code: E1A+B+C,L1SAIF (GAL,QZS)  
Definition at line 43 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.183 CODE\_L2C**

```
uint32_t CODE_L2C = 14 [constexpr]
```

obs code: L2C/A,G1C/A (GPS,GLO)  
Definition at line 44 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.184 CODE\_L2D**

uint32\_t CODE\_L2D = 15 [constexpr]  
obs code: L2 L1C/A-(P2-P1) (GPS)  
Definition at line 45 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.185 CODE\_L2I**

uint32\_t CODE\_L2I = 40 [constexpr]  
obs code: B1I (BDS)  
Definition at line 70 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.186 CODE\_L2L**

uint32\_t CODE\_L2L = 17 [constexpr]  
obs code: L2C(L) (GPS,QZS)  
Definition at line 47 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.187 CODE\_L2M**

uint32\_t CODE\_L2M = 22 [constexpr]  
obs code: L2M (GPS)  
Definition at line 52 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.188 CODE\_L2N**

uint32\_t CODE\_L2N = 23 [constexpr]  
obs code: L2codeless (GPS)  
Definition at line 53 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.189 CODE\_L2P**

uint32\_t CODE\_L2P = 19 [constexpr]  
obs code: L2P,G2P (GPS,GLO)  
Definition at line 49 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.190 CODE\_L2Q**

uint32\_t CODE\_L2Q = 41 [constexpr]  
obs code: B1Q (BDS)  
Definition at line 71 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.191 CODE\_L2S**

uint32\_t CODE\_L2S = 16 [constexpr]  
obs code: L2C(M) (GPS,QZS)  
Definition at line 46 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.192 CODE\_L2W**

uint32\_t CODE\_L2W = 20 [constexpr]  
obs code: L2 Z-track (GPS)  
Definition at line 50 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.193 CODE\_L2X**

uint32\_t CODE\_L2X = 18 [constexpr]  
obs code: L2C(M+L),B1I+Q (GPS,QZS,BDS)  
Definition at line 48 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.194 CODE\_L2Y**

```
uint32_t CODE_L2Y = 21 [constexpr]
```

obs code: L2Y (GPS)  
Definition at line 51 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.195 CODE\_L3I**

```
uint32_t CODE_L3I = 44 [constexpr]
```

obs code: G3I (GLO)  
Definition at line 74 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.196 CODE\_L3Q**

```
uint32_t CODE_L3Q = 45 [constexpr]
```

obs code: G3Q (GLO)  
Definition at line 75 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.197 CODE\_L3X**

```
uint32_t CODE_L3X = 46 [constexpr]
```

obs code: G3I+Q (GLO)  
Definition at line 76 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.198 CODE\_L5A**

```
uint32_t CODE_L5A = 49 [constexpr]
```

obs code: L5A SPS (IRN)  
Definition at line 79 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.199 CODE\_L5B**

```
uint32_t CODE_L5B = 50 [constexpr]
```

obs code: L5B RS(D) (IRN)  
Definition at line 80 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.200 CODE\_L5C**

```
uint32_t CODE_L5C = 51 [constexpr]
```

obs code: L5C RS(P) (IRN)  
Definition at line 81 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.201 CODE\_L5I**

```
uint32_t CODE_L5I = 24 [constexpr]
```

obs code: L5/E5aI (GPS,GAL,QZS,SBS)  
Definition at line 54 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.202 CODE\_L5Q**

```
uint32_t CODE_L5Q = 25 [constexpr]
```

obs code: L5/E5aQ (GPS,GAL,QZS,SBS)  
Definition at line 55 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.203 CODE\_L5X**

```
uint32_t CODE_L5X = 26 [constexpr]
```

obs code: L5/E5aI+Q/L5B+C (GPS,GAL,QZS,IRN,SBS)  
Definition at line 56 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.204 CODE\_L6A**

uint32\_t CODE\_L6A = 30 [constexpr]  
obs code: E6A (GAL)  
Definition at line 60 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.205 CODE\_L6B**

uint32\_t CODE\_L6B = 31 [constexpr]  
obs code: E6B (GAL)  
Definition at line 61 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.206 CODE\_L6C**

uint32\_t CODE\_L6C = 32 [constexpr]  
obs code: E6C (GAL)  
Definition at line 62 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.207 CODE\_L6I**

uint32\_t CODE\_L6I = 42 [constexpr]  
obs code: B3I (BDS)  
Definition at line 72 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.208 CODE\_L6L**

uint32\_t CODE\_L6L = 36 [constexpr]  
obs code: LEXL (QZS)  
Definition at line 66 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.209 CODE\_L6Q**

uint32\_t CODE\_L6Q = 43 [constexpr]  
obs code: B3Q (BDS)  
Definition at line 73 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.210 CODE\_L6S**

uint32\_t CODE\_L6S = 35 [constexpr]  
obs code: LEXS (QZS)  
Definition at line 65 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.211 CODE\_L6X**

uint32\_t CODE\_L6X = 33 [constexpr]  
obs code: E6B+C,LEXS+L,B3I+Q (GAL,QZS,BDS)  
Definition at line 63 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.212 CODE\_L6Z**

uint32\_t CODE\_L6Z = 34 [constexpr]  
obs code: E6A+B+C (GAL)  
Definition at line 64 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.213 CODE\_L7I**

uint32\_t CODE\_L7I = 27 [constexpr]  
obs code: E5bI,B2I (GAL,BDS)  
Definition at line 57 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.214 CODE\_L7Q**

`uint32_t CODE_L7Q = 28 [constexpr]`  
obs code: E5bQ,B2Q (GAL,BDS)  
Definition at line 58 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.215 CODE\_L7X**

`uint32_t CODE_L7X = 29 [constexpr]`  
obs code: E5bl+Q,B2l+Q (GAL,BDS)  
Definition at line 59 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.216 CODE\_L8I**

`uint32_t CODE_L8I = 37 [constexpr]`  
obs code: E5(a+b)I (GAL)  
Definition at line 67 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.217 CODE\_L8Q**

`uint32_t CODE_L8Q = 38 [constexpr]`  
obs code: E5(a+b)Q (GAL)  
Definition at line 68 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.218 CODE\_L8X**

`uint32_t CODE_L8X = 39 [constexpr]`  
obs code: E5(a+b)l+Q (GAL)  
Definition at line 69 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.219 CODE\_L9A**

`uint32_t CODE_L9A = 52 [constexpr]`  
obs code: SA SPS (IRN)  
Definition at line 82 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.220 CODE\_L9B**

`uint32_t CODE_L9B = 53 [constexpr]`  
obs code: SB RS(D) (IRN)  
Definition at line 83 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.221 CODE\_L9C**

`uint32_t CODE_L9C = 54 [constexpr]`  
obs code: SC RS(P) (IRN)  
Definition at line 84 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.222 CODE\_L9X**

`uint32_t CODE_L9X = 55 [constexpr]`  
obs code: SB+C (IRN)  
Definition at line 85 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.223 CODE\_NONE**

`uint32_t CODE_NONE = 0 [constexpr]`  
obs code: none or unknown  
Definition at line 30 of file [gnss\\_obs\\_codes.h](#).



**9.13.6.4.224 D1\_A0\_ALMANAC\_LSB**

double D1\_A0\_ALMANAC\_LSB = TWO\_N20 [constexpr]  
Definition at line 67 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.225 D1\_A0\_LSB**

double D1\_A0\_LSB = TWO\_N33 [constexpr]  
Definition at line 46 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.226 D1\_A0GAL\_LSB**

double D1\_A0GAL\_LSB = 0.1e-9 [constexpr]  
Definition at line 77 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.227 D1\_A0GLO\_LSB**

double D1\_A0GLO\_LSB = 0.1e-9 [constexpr]  
Definition at line 79 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.228 D1\_A0GPS\_LSB**

double D1\_A0GPS\_LSB = 0.1e-9 [constexpr]  
Definition at line 75 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.229 D1\_A0UTC\_LSB**

double D1\_A0UTC\_LSB = TWO\_N30 [constexpr]  
Definition at line 81 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.230 D1\_A1\_ALMANAC\_LSB**

double D1\_A1\_ALMANAC\_LSB = TWO\_N38 [constexpr]  
Definition at line 66 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.231 D1\_A1\_LSB**

double D1\_A1\_LSB = TWO\_N50 [constexpr]  
Definition at line 47 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.232 D1\_A1GAL\_LSB**

double D1\_A1GAL\_LSB = 0.1e-9 [constexpr]  
Definition at line 78 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.233 D1\_A1GLO\_LSB**

double D1\_A1GLO\_LSB = 0.1e-9 [constexpr]  
Definition at line 80 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.234 D1\_A1GPS\_LSB**

double D1\_A1GPS\_LSB = 0.1e-9 [constexpr]  
Definition at line 76 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.235 D1\_A1UTC\_LSB**

double D1\_A1UTC\_LSB = TWO\_N50 [constexpr]  
Definition at line 82 of file [Beidou\\_DNAV.h](#).

#### 9.13.6.4.236 D1\_A2\_LSB

double D1\_A2\_LSB = TWO\_N66 [constexpr]  
Definition at line 45 of file [Beidou\\_DNAV.h](#).

#### 9.13.6.4.237 D1\_ALPHA0\_LSB

double D1\_ALPHA0\_LSB = TWO\_N30 [constexpr]  
Definition at line 37 of file [Beidou\\_DNAV.h](#).

#### 9.13.6.4.238 D1\_ALPHA1\_LSB

double D1\_ALPHA1\_LSB = TWO\_N27 [constexpr]  
Definition at line 38 of file [Beidou\\_DNAV.h](#).

#### 9.13.6.4.239 D1\_ALPHA2\_LSB

double D1\_ALPHA2\_LSB = TWO\_N24 [constexpr]  
Definition at line 39 of file [Beidou\\_DNAV.h](#).

#### 9.13.6.4.240 D1\_ALPHA3\_LSB

double D1\_ALPHA3\_LSB = TWO\_N24 [constexpr]  
Definition at line 40 of file [Beidou\\_DNAV.h](#).

#### 9.13.6.4.241 D1\_BETA0\_LSB

double D1\_BETA0\_LSB = TWO\_P11 [constexpr]  
Definition at line 41 of file [Beidou\\_DNAV.h](#).

#### 9.13.6.4.242 D1\_BETA1\_LSB

double D1\_BETA1\_LSB = TWO\_P14 [constexpr]  
Definition at line 42 of file [Beidou\\_DNAV.h](#).

#### 9.13.6.4.243 D1\_BETA2\_LSB

double D1\_BETA2\_LSB = TWO\_P16 [constexpr]  
Definition at line 43 of file [Beidou\\_DNAV.h](#).

#### 9.13.6.4.244 D1\_BETA3\_LSB

double D1\_BETA3\_LSB = TWO\_P16 [constexpr]  
Definition at line 44 of file [Beidou\\_DNAV.h](#).

#### 9.13.6.4.245 D1\_CIC\_LSB

double D1\_CIC\_LSB = TWO\_N31 [constexpr]  
Definition at line 58 of file [Beidou\\_DNAV.h](#).

#### 9.13.6.4.246 D1\_CIS\_LSB

double D1\_CIS\_LSB = TWO\_N31 [constexpr]  
Definition at line 60 of file [Beidou\\_DNAV.h](#).

#### 9.13.6.4.247 D1\_CRC\_LSB

double D1\_CRC\_LSB = TWO\_N6 [constexpr]  
Definition at line 53 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.248 D1\_CRS\_LSB**

double D1\_CRS\_LSB = TWO\_N6 [constexpr]  
Definition at line 54 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.249 D1\_CUC\_LSB**

double D1\_CUC\_LSB = TWO\_N31 [constexpr]  
Definition at line 49 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.250 D1\_CUS\_LSB**

double D1\_CUS\_LSB = TWO\_N31 [constexpr]  
Definition at line 52 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.251 D1\_DELTA\_I\_LSB**

double D1\_DELTA\_I\_LSB = PI\_TWO\_N19 [constexpr]  
Definition at line 70 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.252 D1\_DELTA\_N\_LSB**

double D1\_DELTA\_N\_LSB = PI\_TWO\_N43 [constexpr]  
Definition at line 48 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.253 D1\_E\_ALMANAC\_LSB**

double D1\_E\_ALMANAC\_LSB = TWO\_N21 [constexpr]  
Definition at line 69 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.254 D1\_E\_LSB**

double D1\_E\_LSB = TWO\_N33 [constexpr]  
Definition at line 51 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.255 D1\_I0\_LSB**

double D1\_I0\_LSB = PI\_TWO\_N31 [constexpr]  
Definition at line 57 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.256 D1\_IDOT\_LSB**

double D1\_IDOT\_LSB = PI\_TWO\_N43 [constexpr]  
Definition at line 61 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.257 D1\_M0\_ALMANAC\_LSB**

double D1\_M0\_ALMANAC\_LSB = PI\_TWO\_N23 [constexpr]  
Definition at line 74 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.258 D1\_M0\_LSB**

double D1\_M0\_LSB = PI\_TWO\_N31 [constexpr]  
Definition at line 50 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.259 D1\_OMEGA0\_ALMANAC\_LSB**

double D1\_OMEGA0\_ALMANAC\_LSB = PI\_TWO\_N23 [constexpr]  
Definition at line 68 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.260 D1\_OMEGA0\_LSB**

```
double D1_OMEGA0_LSB = PI_TWO_N31 [constexpr]
```

Definition at line 62 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.261 D1\_OMEGA\_ALMANAC\_LSB**

```
double D1_OMEGA_ALMANAC_LSB = PI_TWO_N23 [constexpr]
```

Definition at line 73 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.262 D1\_OMEGA\_DOT\_ALMANAC\_LSB**

```
double D1_OMEGA_DOT_ALMANAC_LSB = PI_TWO_N38 [constexpr]
```

Definition at line 72 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.263 D1\_OMEGA\_DOT\_LSB**

```
double D1_OMEGA_DOT_LSB = PI_TWO_N43 [constexpr]
```

Definition at line 59 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.264 D1\_OMEGA\_LSB**

```
double D1_OMEGA_LSB = PI_TWO_N31 [constexpr]
```

Definition at line 63 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.265 D1\_SQRT\_A\_ALMANAC\_LSB**

```
double D1_SQRT_A_ALMANAC_LSB = TWO_N11 [constexpr]
```

Definition at line 65 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.266 D1\_SQRT\_A\_LSB**

```
double D1_SQRT_A_LSB = TWO_N19 [constexpr]
```

Definition at line 55 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.267 D1\_TGD1\_LSB**

```
double D1_TGD1_LSB = 0.1e-9 [constexpr]
```

Definition at line 35 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.268 D1\_TGD2\_LSB**

```
double D1_TGD2_LSB = 0.1e-9 [constexpr]
```

Definition at line 36 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.269 D1\_TOA\_LSB**

```
double D1_TOA_LSB = TWO_P12 [constexpr]
```

Definition at line 71 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.270 D1\_TOC\_LSB**

```
double D1_TOC_LSB = TWO_P3 [constexpr]
```

Definition at line 34 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.271 D1\_TOE\_LSB**

```
double D1_TOE_LSB = TWO_P3 [constexpr]
```

Definition at line 56 of file [Beidou\\_DNAV.h](#).

**9.13.6.4.272 D2R**

```
double D2R = GNSS_PI / 180.0 [constexpr]
deg to rad
Definition at line 123 of file MATH_CONSTANTS.h.
```

**9.13.6.4.273 DELTA\_A\_7\_LSB**

```
double DELTA_A_7_LSB = TWO_N9 [constexpr]
Definition at line 165 of file Galileo_INAV.h.
```

**9.13.6.4.274 DELTA\_A\_8\_LSB**

```
double DELTA_A_8_LSB = TWO_N9 [constexpr]
Definition at line 190 of file Galileo_INAV.h.
```

**9.13.6.4.275 DELTA\_A\_9\_LSB**

```
double DELTA_A_9_LSB = TWO_N9 [constexpr]
Definition at line 218 of file Galileo_INAV.h.
```

**9.13.6.4.276 DELTA\_I\_7\_LSB**

```
double DELTA_I_7_LSB = TWO_N14 [constexpr]
Definition at line 171 of file Galileo_INAV.h.
```

**9.13.6.4.277 DELTA\_I\_8\_LSB**

```
double DELTA_I_8_LSB = TWO_N14 [constexpr]
Definition at line 196 of file Galileo_INAV.h.
```

**9.13.6.4.278 DELTA\_I\_9\_LSB**

```
double DELTA_I_9_LSB = TWO_N14 [constexpr]
Definition at line 224 of file Galileo_INAV.h.
```

**9.13.6.4.279 DELTA\_N\_3\_LSB**

```
double DELTA_N_3_LSB = PI_TWO_N43 [constexpr]
Definition at line 84 of file Galileo_INAV.h.
```

**9.13.6.4.280 DELTA\_N\_LSB**

```
double DELTA_N_LSB = PI_TWO_N43 [constexpr]
Definition at line 109 of file GPS_L1_CA.h.
```

**9.13.6.4.281 DELTAT\_LS\_LSB**

```
double DELTAT_LS_LSB = 1 [constexpr]
Definition at line 177 of file GPS_L1_CA.h.
```

**9.13.6.4.282 DELTAT\_LSF\_LSB**

```
double DELTAT_LSF_LSB = 1 [constexpr]
Definition at line 183 of file GPS_L1_CA.h.
```

**9.13.6.4.283 DFRQ1\_GLO**

```
double DFRQ1_GLO = 0.56250e6 [constexpr]
GLONASS G1 bias frequency (Hz/n)
Definition at line 39 of file gnss_frequencies.h.
```

**9.13.6.4.284 DFRQ2\_GLO**

```
double DFRQ2_GLO = 0.43750e6 [constexpr]  
GLONASS G2 bias frequency (Hz/n)  
Definition at line 41 of file gnss\_frequencies.h.
```

**9.13.6.4.285 DN\_LSB**

```
double DN_LSB = 1 [constexpr]  
Definition at line 181 of file GPS\_L1\_CA.h.
```

**9.13.6.4.286 E\_1\_LSB**

```
double E_1_LSB = TWO_N33 [constexpr]  
Definition at line 61 of file Galileo\_INAV.h.
```

**9.13.6.4.287 E\_7\_LSB**

```
double E_7_LSB = TWO_N16 [constexpr]  
Definition at line 167 of file Galileo\_INAV.h.
```

**9.13.6.4.288 E\_8\_LSB**

```
double E_8_LSB = TWO_N16 [constexpr]  
Definition at line 192 of file Galileo\_INAV.h.
```

**9.13.6.4.289 E\_9\_LSB**

```
double E_9_LSB = TWO_N16 [constexpr]  
Definition at line 220 of file Galileo\_INAV.h.
```

**9.13.6.4.290 ECCENTRICITY\_LSB**

```
double ECCENTRICITY_LSB = TWO_N33 [constexpr]  
Definition at line 115 of file GPS\_L1\_CA.h.
```

**9.13.6.4.291 FIRST\_RS\_BIT**

```
int32_t FIRST_RS_BIT = 7 [constexpr]  
Definition at line 274 of file Galileo\_INAV.h.
```

**9.13.6.4.292 FIRST\_RS\_BIT\_AFTER\_IODNAV**

```
int32_t FIRST_RS_BIT_AFTER_IODNAV = 17 [constexpr]  
Definition at line 275 of file Galileo\_INAV.h.
```

**9.13.6.4.293 FNAV\_A0\_4\_LSB**

```
double FNAV_A0_4_LSB = TWO_N30 [constexpr]  
Definition at line 109 of file Galileo\_FNAV.h.
```

**9.13.6.4.294 FNAV\_A0G\_4\_LSB**

```
double FNAV_A0G_4_LSB = TWO_N35 [constexpr]  
Definition at line 122 of file Galileo\_FNAV.h.
```

**9.13.6.4.295 FNAV\_A12\_2\_LSB**

```
double FNAV_A12_2_LSB = TWO_N19 [constexpr]  
Definition at line 73 of file Galileo\_FNAV.h.
```

**9.13.6.4.296 FNAV\_A1\_4\_LSB**

double FNAV\_A1\_4\_LSB = TWO\_N50 [constexpr]  
Definition at line 111 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.297 FNAV\_A1G\_4\_LSB**

double FNAV\_A1G\_4\_LSB = TWO\_N51 [constexpr]  
Definition at line 124 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.298 FNAV\_AF0\_1\_LSB**

double FNAV\_AF0\_1\_LSB = TWO\_N34 [constexpr]  
Definition at line 40 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.299 FNAV\_AF0\_5\_LSB**

double FNAV\_AF0\_5\_LSB = TWO\_N19 [constexpr]  
Definition at line 149 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.300 FNAV\_AF1\_1\_LSB**

double FNAV\_AF1\_1\_LSB = TWO\_N46 [constexpr]  
Definition at line 42 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.301 FNAV\_AF1\_5\_LSB**

double FNAV\_AF1\_5\_LSB = TWO\_N38 [constexpr]  
Definition at line 151 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.302 FNAV\_AF2\_1\_LSB**

double FNAV\_AF2\_1\_LSB = TWO\_N59 [constexpr]  
Definition at line 44 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.303 FNAV\_AI0\_1\_LSB**

double FNAV\_AI0\_1\_LSB = TWO\_N2 [constexpr]  
Definition at line 47 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.304 FNAV\_AI1\_1\_LSB**

double FNAV\_AI1\_1\_LSB = TWO\_N8 [constexpr]  
Definition at line 49 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.305 FNAV\_AI2\_1\_LSB**

double FNAV\_AI2\_1\_LSB = TWO\_N15 [constexpr]  
Definition at line 51 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.306 FNAV\_BGD\_1\_LSB**

double FNAV\_BGD\_1\_LSB = TWO\_N32 [constexpr]  
Definition at line 58 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.307 FNAV\_CIC\_4\_LSB**

double FNAV\_CIC\_4\_LSB = TWO\_N29 [constexpr]  
Definition at line 105 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.308 FNAV\_CIS\_4\_LSB**

double FNAV\_CIS\_4\_LSB = TWO\_N29 [constexpr]  
Definition at line 107 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.309 FNAV\_CRC\_3\_LSB**

double FNAV\_CRC\_3\_LSB = TWO\_N5 [constexpr]  
Definition at line 94 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.310 FNAV\_CRS\_3\_LSB**

double FNAV\_CRS\_3\_LSB = TWO\_N5 [constexpr]  
Definition at line 96 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.311 FNAV\_CUC\_3\_LSB**

double FNAV\_CUC\_3\_LSB = TWO\_N29 [constexpr]  
Definition at line 90 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.312 FNAV\_CUS\_3\_LSB**

double FNAV\_CUS\_3\_LSB = TWO\_N29 [constexpr]  
Definition at line 92 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.313 FNAV\_DELTA12\_5\_LSB**

double FNAV\_DELTA12\_5\_LSB = TWO\_N9 [constexpr]  
Definition at line 135 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.314 FNAV\_DELTAI\_5\_LSB**

double FNAV\_DELTAI\_5\_LSB = TWO\_N14 [constexpr]  
Definition at line 141 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.315 FNAV\_DELTAN\_3\_LSB**

double FNAV\_DELTAN\_3\_LSB = PI\_TWO\_N43 [constexpr]  
Definition at line 88 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.316 FNAV\_E\_2\_LSB**

double FNAV\_E\_2\_LSB = TWO\_N33 [constexpr]  
Definition at line 71 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.317 FNAV\_E\_5\_LSB**

double FNAV\_E\_5\_LSB = TWO\_N16 [constexpr]  
Definition at line 137 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.318 FNAV\_I0\_3\_LSB**

double FNAV\_I0\_3\_LSB = PI\_TWO\_N31 [constexpr]  
Definition at line 84 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.319 FNAV\_IDOT\_2\_LSB**

double FNAV\_IDOT\_2\_LSB = PI\_TWO\_N43 [constexpr]  
Definition at line 77 of file [Galileo\\_FNAV.h](#).



**9.13.6.4.320 FNAV\_M0\_2\_LSB**

double FNAV\_M0\_2\_LSB = [PI\\_TWO\\_N31](#) [constexpr]  
Definition at line 67 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.321 FNAV\_M0\_5\_LSB**

double FNAV\_M0\_5\_LSB = [TWO\\_N15](#) [constexpr]  
Definition at line 147 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.322 FNAV\_OMEGA0\_2\_LSB**

double FNAV\_OMEGA0\_2\_LSB = [PI\\_TWO\\_N31](#) [constexpr]  
Definition at line 75 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.323 FNAV\_OMEGA0\_5\_LSB**

double FNAV\_OMEGA0\_5\_LSB = [TWO\\_N15](#) [constexpr]  
Definition at line 143 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.324 FNAV\_OMEGADOT\_2\_LSB**

double FNAV\_OMEGADOT\_2\_LSB = [PI\\_TWO\\_N43](#) [constexpr]  
Definition at line 69 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.325 FNAV\_OMEGADOT\_5\_LSB**

double FNAV\_OMEGADOT\_5\_LSB = [TWO\\_N33](#) [constexpr]  
Definition at line 145 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.326 FNAV\_T0A\_5\_LSB**

int32\_t FNAV\_T0A\_5\_LSB = 600 [constexpr]  
Definition at line 132 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.327 FNAV\_T0C\_1\_LSB**

int32\_t FNAV\_T0C\_1\_LSB = 60 [constexpr]  
Definition at line 38 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.328 FNAV\_T0E\_3\_LSB**

int32\_t FNAV\_T0E\_3\_LSB = 60 [constexpr]  
Definition at line 98 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.329 FNAV\_T0G\_4\_LSB**

int32\_t FNAV\_T0G\_4\_LSB = 3600 [constexpr]  
Definition at line 120 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.330 FNAV\_T0T\_4\_LSB**

int32\_t FNAV\_T0T\_4\_LSB = 3600 [constexpr]  
Definition at line 114 of file [Galileo\\_FNAV.h](#).

**9.13.6.4.331 FNAV\_W\_3\_LSB**

double FNAV\_W\_3\_LSB = [PI\\_TWO\\_N31](#) [constexpr]  
Definition at line 86 of file [Galileo\\_FNAV.h](#).

#### 9.13.6.4.332 FNAV\_W\_5\_LSB

double FNAV\_W\_5\_LSB = TWO\_N15 [constexpr]  
Definition at line 139 of file [Galileo\\_FNAV.h](#).

#### 9.13.6.4.333 FREQ1

double FREQ1 = 1.57542e9 [constexpr]  
L1/E1 frequency (Hz)  
Definition at line 31 of file [gnss\\_frequencies.h](#).

#### 9.13.6.4.334 FREQ1\_BDS

double FREQ1\_BDS = 1.561098e9 [constexpr]  
BeiDou B1 frequency (Hz)  
Definition at line 43 of file [gnss\\_frequencies.h](#).

#### 9.13.6.4.335 FREQ1\_GLO

double FREQ1\_GLO = 1.60200e9 [constexpr]  
GLONASS G1 base frequency (Hz)  
Definition at line 38 of file [gnss\\_frequencies.h](#).

#### 9.13.6.4.336 FREQ2

double FREQ2 = 1.22760e9 [constexpr]  
L2 frequency (Hz)  
Definition at line 32 of file [gnss\\_frequencies.h](#).

#### 9.13.6.4.337 FREQ2\_BDS

double FREQ2\_BDS = 1.20714e9 [constexpr]  
BeiDou B2 frequency (Hz)  
Definition at line 44 of file [gnss\\_frequencies.h](#).

#### 9.13.6.4.338 FREQ2\_GLO

double FREQ2\_GLO = 1.24600e9 [constexpr]  
GLONASS G2 base frequency (Hz)  
Definition at line 40 of file [gnss\\_frequencies.h](#).

#### 9.13.6.4.339 FREQ3\_BDS

double FREQ3\_BDS = 1.26852e9 [constexpr]  
BeiDou B3 frequency (Hz)  
Definition at line 45 of file [gnss\\_frequencies.h](#).

#### 9.13.6.4.340 FREQ3\_GLO

double FREQ3\_GLO = 1.202025e9 [constexpr]  
GLONASS G3 frequency (Hz)  
Definition at line 42 of file [gnss\\_frequencies.h](#).

#### 9.13.6.4.341 FREQ5

double FREQ5 = 1.17645e9 [constexpr]  
L5/E5a frequency (Hz)  
Definition at line 33 of file [gnss\\_frequencies.h](#).

**9.13.6.4.342   FREQ6**

double FREQ6 = 1.27875e9   [constexpr]  
E6/LEX frequency (Hz)  
Definition at line 34 of file [gnss\\_frequencies.h](#).

**9.13.6.4.343   FREQ7**

double FREQ7 = 1.20714e9   [constexpr]  
E5b frequency (Hz)  
Definition at line 35 of file [gnss\\_frequencies.h](#).

**9.13.6.4.344   FREQ8**

double FREQ8 = 1.191795e9   [constexpr]  
E5a+b frequency (Hz)  
Definition at line 36 of file [gnss\\_frequencies.h](#).

**9.13.6.4.345   FREQ9**

double FREQ9 = 2.492028e9   [constexpr]  
S frequency (Hz)  
Definition at line 37 of file [gnss\\_frequencies.h](#).

**9.13.6.4.346   GALILEO\_CNAV\_BITS\_FOR\_CRC**

int32\_t GALILEO\_CNAV\_BITS\_FOR\_CRC = GALILEO\_CNAV\_HAS\_PAGE\_DATA\_BITS + GALILEO\_CNAV\_PAGE\_←  
RESERVED\_BITS   [constexpr]  
Definition at line 78 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.347   GALILEO\_CNAV\_BYTES\_FOR\_CRC**

int32\_t GALILEO\_CNAV\_BYTES\_FOR\_CRC = 58   [constexpr]  
Definition at line 68 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.348   GALILEO\_CNAV\_CRC\_LENGTH**

int32\_t GALILEO\_CNAV\_CRC\_LENGTH = 24   [constexpr]  
Definition at line 69 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.349   GALILEO\_CNAV\_HAS\_PAGE\_DATA\_BITS**

int32\_t GALILEO\_CNAV\_HAS\_PAGE\_DATA\_BITS = 448   [constexpr]  
Definition at line 66 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.350   GALILEO\_CNAV\_INFORMATION\_VECTOR\_LENGTH**

int32\_t GALILEO\_CNAV\_INFORMATION\_VECTOR\_LENGTH = 32   [constexpr]  
Definition at line 76 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.351   GALILEO\_CNAV\_INTERLEAVER\_COLS**

int32\_t GALILEO\_CNAV\_INTERLEAVER\_COLS = 123   [constexpr]  
Definition at line 64 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.352   GALILEO\_CNAV\_INTERLEAVER\_ROWS**

int32\_t GALILEO\_CNAV\_INTERLEAVER\_ROWS = 8   [constexpr]  
Definition at line 63 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.353 GALILEO\_CNAV\_MAX\_NUMBER\_SYMBOLS\_ENCODED\_BLOCK**

```
int32_t GALILEO_CNAV_MAX_NUMBER_SYMBOLS_ENCODED_BLOCK = 255 [constexpr]
```

Definition at line 73 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.354 GALILEO\_CNAV\_MESSAGE\_BITS\_PER\_PAGE**

```
int32_t GALILEO_CNAV_MESSAGE_BITS_PER_PAGE = 424 [constexpr]
```

Definition at line 70 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.355 GALILEO\_CNAV\_MT1\_HEADER\_BITS**

```
int32_t GALILEO_CNAV_MT1_HEADER_BITS = 32 [constexpr]
```

Definition at line 74 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.356 GALILEO\_CNAV\_OCTETS\_IN\_SUBPAGE**

```
int32_t GALILEO_CNAV_OCTETS_IN_SUBPAGE = 53 [constexpr]
```

Definition at line 75 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.357 GALILEO\_CNAV\_PAGE\_HEADER\_BITS**

```
int32_t GALILEO_CNAV_PAGE_HEADER_BITS = 24 [constexpr]
```

Definition at line 71 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.358 GALILEO\_CNAV\_PAGE\_MS**

```
int32_t GALILEO_CNAV_PAGE_MS = 1 [constexpr]
```

Definition at line 62 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.359 GALILEO\_CNAV\_PAGE\_RESERVED\_BITS**

```
int32_t GALILEO_CNAV_PAGE_RESERVED_BITS = 14 [constexpr]
```

Definition at line 67 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.360 GALILEO\_CNAV\_PREAMBLE**

```
char GALILEO_CNAV_PREAMBLE[17] = "1011011101110000" [constexpr]
```

Definition at line 99 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.361 GALILEO\_CNAV\_PREAMBLE\_LENGTH\_BITS**

```
int32_t GALILEO_CNAV_PREAMBLE_LENGTH_BITS = 16 [constexpr]
```

Definition at line 72 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.362 GALILEO\_CNAV\_PREAMBLE\_PERIOD\_SYMBOLS**

```
int32_t GALILEO_CNAV_PREAMBLE_PERIOD_SYMBOLS = 1000 [constexpr]
```

Definition at line 61 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.363 GALILEO\_CNAV\_SYMBOLS\_PER\_PAGE**

```
int32_t GALILEO_CNAV_SYMBOLS_PER_PAGE = 1000 [constexpr]
```

Definition at line 60 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.364 GALILEO\_CNAV\_TELEMETRY\_RATE\_BITS\_SECOND**

```
int32_t GALILEO_CNAV_TELEMETRY_RATE_BITS_SECOND = 1000 [constexpr]
```

Definition at line 65 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.365 GALILEO\_DATA\_FRAME\_BITS**

`int32_t GALILEO_DATA_FRAME_BITS = 196 [constexpr]`  
Definition at line 47 of file [Galileo\\_INAV.h](#).

**9.13.6.4.366 GALILEO\_DATA\_FRAME\_BYTES**

`int32_t GALILEO_DATA_FRAME_BYTES = 25 [constexpr]`  
Definition at line 48 of file [Galileo\\_INAV.h](#).

**9.13.6.4.367 GALILEO\_DATA\_JK\_BITS**

`int32_t GALILEO_DATA_JK_BITS = 128 [constexpr]`  
Definition at line 46 of file [Galileo\\_INAV.h](#).

**9.13.6.4.368 GALILEO\_E1\_B\_CODE\_LENGTH\_CHIPS**

`double GALILEO_E1_B_CODE_LENGTH_CHIPS = 4092.0 [constexpr]`  
Galileo E1-B code length [chips].  
Definition at line 39 of file [Galileo\\_E1.h](#).

**9.13.6.4.369 GALILEO\_E1\_B\_PRIMARY\_CODE**

`char GALILEO_E1_B_PRIMARY_CODE[GALILEO_E1_NUMBER_OF_CODES][1024] [constexpr]`  
Definition at line 56 of file [Galileo\\_E1.h](#).

**9.13.6.4.370 GALILEO\_E1\_B\_PRIMARY\_CODE\_STR\_LENGTH**

`size_t GALILEO_E1_B_PRIMARY_CODE_STR_LENGTH = 1023 [constexpr]`  
Definition at line 55 of file [Galileo\\_E1.h](#).

**9.13.6.4.371 GALILEO\_E1\_B\_SAMPLES\_PER\_SYMBOL**

`int32_t GALILEO_E1_B_SAMPLES_PER_SYMBOL = 1 [constexpr]`  
 $(\text{Galileo\_E1\_CODE\_CHIP\_RATE\_HZ} / \text{Galileo\_E1\_B\_CODE\_LENGTH\_CHIPS}) / \text{Galileo\_E1\_B\_SYMBOL\_RATE\_BPS}$   
Definition at line 42 of file [Galileo\\_E1.h](#).

**9.13.6.4.372 GALILEO\_E1\_B\_SYMBOL\_RATE\_BPS**

`double GALILEO_E1_B_SYMBOL_RATE_BPS = 250.0 [constexpr]`  
Galileo E1-B symbol rate [bits/second].  
Definition at line 40 of file [Galileo\\_E1.h](#).

**9.13.6.4.373 GALILEO\_E1\_C\_PRIMARY\_CODE**

`char GALILEO_E1_C_PRIMARY_CODE[GALILEO_E1_NUMBER_OF_CODES][1024] [constexpr]`  
Definition at line 760 of file [Galileo\\_E1.h](#).

**9.13.6.4.374 GALILEO\_E1\_C\_PRIMARY\_CODE\_STR\_LENGTH**

`size_t GALILEO_E1_C_PRIMARY_CODE_STR_LENGTH = 1023 [constexpr]`  
Definition at line 759 of file [Galileo\\_E1.h](#).

**9.13.6.4.375 GALILEO\_E1\_C\_SECONDARY\_CODE**

`char GALILEO_E1_C_SECONDARY_CODE[26] = "0011100000001010110110010" [constexpr]`  
Definition at line 52 of file [Galileo\\_E1.h](#).

**9.13.6.4.376 GALILEO\_E1\_C\_SECONDARY\_CODE\_LENGTH**

```
int32_t GALILEO_E1_C_SECONDARY_CODE_LENGTH = 25 [constexpr]
```

Galileo E1-C secondary code length [chips].

Definition at line 43 of file [Galileo\\_E1.h](#).

**9.13.6.4.377 GALILEO\_E1\_CODE\_CHIP\_RATE\_CPS**

```
double GALILEO_E1_CODE_CHIP_RATE_CPS = 1.023e6 [constexpr]
```

Galileo E1 code rate [chips/s].

Definition at line 35 of file [Galileo\\_E1.h](#).

**9.13.6.4.378 GALILEO\_E1\_CODE\_PERIOD\_MS**

```
uint32_t GALILEO_E1_CODE_PERIOD_MS = 4 [constexpr]
```

Galileo E1 code period [ms].

Definition at line 41 of file [Galileo\\_E1.h](#).

**9.13.6.4.379 GALILEO\_E1\_CODE\_PERIOD\_S**

```
double GALILEO_E1_CODE_PERIOD_S = 0.004 [constexpr]
```

Galileo E1 code period [s].

Definition at line 36 of file [Galileo\\_E1.h](#).

**9.13.6.4.380 GALILEO\_E1\_FREQ\_HZ**

```
double GALILEO_E1_FREQ_HZ = FREQ1 [constexpr]
```

Galileo E1 carrier frequency [Hz].

Definition at line 34 of file [Galileo\\_E1.h](#).

**9.13.6.4.381 GALILEO\_E1\_HISTORY\_DEEP**

```
int32_t GALILEO_E1_HISTORY_DEEP = 100 [constexpr]
```

Observable history length for interpolation.

Definition at line 50 of file [Galileo\\_E1.h](#).

**9.13.6.4.382 GALILEO\_E1\_NUMBER\_OF\_CODES**

```
int32_t GALILEO_E1_NUMBER_OF_CODES = 50 [constexpr]
```

Definition at line 44 of file [Galileo\\_E1.h](#).

**9.13.6.4.383 GALILEO\_E1\_OPT\_ACQ\_FS\_SPS**

```
uint32_t GALILEO_E1_OPT_ACQ_FS_SPS = 2000000 [constexpr]
```

Sampling frequency that maximizes the acquisition SNR while using a non-multiple of chip rate.

Definition at line 48 of file [Galileo\\_E1.h](#).

**9.13.6.4.384 GALILEO\_E1\_SUB\_CARRIER\_A\_RATE\_HZ**

```
double GALILEO_E1_SUB_CARRIER_A_RATE_HZ = 1.023e6 [constexpr]
```

Galileo E1 sub-carrier 'a' rate [Hz].

Definition at line 37 of file [Galileo\\_E1.h](#).

**9.13.6.4.385 GALILEO\_E1\_SUB\_CARRIER\_B\_RATE\_HZ**

```
double GALILEO_E1_SUB_CARRIER_B_RATE_HZ = 6.138e6 [constexpr]
```

Galileo E1 sub-carrier 'b' rate [Hz].

Definition at line 38 of file [Galileo\\_E1.h](#).

**9.13.6.4.386 GALILEO\_E5A\_CODE\_CHIP\_RATE\_CPS**

```
double GALILEO_E5A_CODE_CHIP_RATE_CPS = 1.023e7 [constexpr]
```

Galileo E5a code rate [chips/s].

Definition at line 33 of file [Galileo\\_E5a.h](#).

**9.13.6.4.387 GALILEO\_E5A\_CODE\_LENGTH\_CHIPS**

```
int32_t GALILEO_E5A_CODE_LENGTH_CHIPS = 10230 [constexpr]
```

Galileo E5a primary code length [chips].

Definition at line 37 of file [Galileo\\_E5a.h](#).

**9.13.6.4.388 GALILEO\_E5A\_CODE\_PERIOD\_MS**

```
int32_t GALILEO_E5A_CODE_PERIOD_MS = 1 [constexpr]
```

Galileo E5a primary code period [ms].

Definition at line 40 of file [Galileo\\_E5a.h](#).

**9.13.6.4.389 GALILEO\_E5A\_CODE\_PERIOD\_S**

```
double GALILEO_E5A_CODE_PERIOD_S = 0.001 [constexpr]
```

Galileo E5a primary code period [s].

Definition at line 36 of file [Galileo\\_E5a.h](#).

**9.13.6.4.390 GALILEO\_E5A\_CRC\_ERROR\_LIMIT**

```
int32_t GALILEO_E5A_CRC_ERROR_LIMIT = 6 [constexpr]
```

Definition at line 47 of file [Galileo\\_E5a.h](#).

**9.13.6.4.391 GALILEO\_E5A\_FREQ\_HZ**

```
double GALILEO_E5A_FREQ_HZ = FREQ5 [constexpr]
```

Galileo E5a carrier frequency [Hz].

Definition at line 32 of file [Galileo\\_E5a.h](#).

**9.13.6.4.392 GALILEO\_E5A\_HISTORY\_DEEP**

```
int32_t GALILEO_E5A_HISTORY_DEEP = 20 [constexpr]
```

Definition at line 46 of file [Galileo\\_E5a.h](#).

**9.13.6.4.393 GALILEO\_E5A\_I\_PRIMARY\_CODE**

```
char GALILEO_E5A_I_PRIMARY_CODE[GALILEO_E5A_NUMBER_OF_CODES][2559] [constexpr]
```

Definition at line 72 of file [Galileo\\_E5a.h](#).

**9.13.6.4.394 GALILEO\_E5A\_I\_PRIMARY\_CODE\_STR\_LENGTH**

```
size_t GALILEO_E5A_I_PRIMARY_CODE_STR_LENGTH = 2558 [constexpr]
```

Definition at line 71 of file [Galileo\\_E5a.h](#).

**9.13.6.4.395 GALILEO\_E5A\_I\_SECONDARY\_CODE**

```
char GALILEO_E5A_I_SECONDARY_CODE[] = "10000100001011101001" [constexpr]
```

Definition at line 3581 of file [Galileo\\_E5a.h](#).

**9.13.6.4.396 GALILEO\_E5A\_I\_SECONDARY\_CODE\_LENGTH**

```
int32_t GALILEO_E5A_I_SECONDARY_CODE_LENGTH = 20 [constexpr]
```

Galileo E5a-I secondary code length [chips].

Definition at line 38 of file [Galileo\\_E5a.h](#).

**9.13.6.4.397 GALILEO\_E5A\_I\_TIERED\_CODE\_PERIOD\_S**

```
double GALILEO_E5A_I_TIERED_CODE_PERIOD_S = 0.020 [constexpr]
```

Galileo E5a-I tiered code period [s].

Definition at line 34 of file [Galileo\\_E5a.h](#).

**9.13.6.4.398 GALILEO\_E5A\_NUMBER\_OF\_CODES**

```
int32_t GALILEO_E5A_NUMBER_OF_CODES = 50 [constexpr]
```

Definition at line 42 of file [Galileo\\_E5a.h](#).

**9.13.6.4.399 GALILEO\_E5A\_OPT\_ACQ\_FS\_SPS**

```
uint32_t GALILEO_E5A_OPT_ACQ_FS_SPS = 10000000 [constexpr]
```

Sampling frequency that maximizes the acquisition SNR while using a non-multiple of chip rate.

Definition at line 50 of file [Galileo\\_E5a.h](#).

**9.13.6.4.400 GALILEO\_E5A\_Q\_PRIMARY\_CODE**

```
char GALILEO_E5A_Q_PRIMARY_CODE[GALILEO_E5A_NUMBER_OF_CODES][2559] [constexpr]
```

Definition at line 1827 of file [Galileo\\_E5a.h](#).

**9.13.6.4.401 GALILEO\_E5A\_Q\_PRIMARY\_CODE\_STR\_LENGTH**

```
size_t GALILEO_E5A_Q_PRIMARY_CODE_STR_LENGTH = 2558 [constexpr]
```

Definition at line 1826 of file [Galileo\\_E5a.h](#).

**9.13.6.4.402 GALILEO\_E5A\_Q\_SECONDARY\_CODE**

```
char GALILEO_E5A_Q_SECONDARY_CODE[GALILEO_E5A_NUMBER_OF_CODES][101] [constexpr]
```

Definition at line 3585 of file [Galileo\\_E5a.h](#).

**9.13.6.4.403 GALILEO\_E5A\_Q\_SECONDARY\_CODE\_LENGTH**

```
int32_t GALILEO_E5A_Q_SECONDARY_CODE_LENGTH = 100 [constexpr]
```

Galileo E5a-Q secondary code length [chips].

Definition at line 39 of file [Galileo\\_E5a.h](#).

**9.13.6.4.404 GALILEO\_E5A\_Q\_SECONDARY\_CODE\_STR\_LENGTH**

```
size_t GALILEO_E5A_Q_SECONDARY_CODE_STR_LENGTH = 100 [constexpr]
```

Definition at line 3584 of file [Galileo\\_E5a.h](#).

**9.13.6.4.405 GALILEO\_E5A\_Q\_TIERED\_CODE\_PERIOD\_S**

```
double GALILEO_E5A_Q_TIERED_CODE_PERIOD_S = 0.100 [constexpr]
```

Galileo E5a-Q tiered code period [s].

Definition at line 35 of file [Galileo\\_E5a.h](#).

**9.13.6.4.406 GALILEO\_E5A\_SYMBOL\_RATE\_BPS**

```
int32_t GALILEO_E5A_SYMBOL_RATE_BPS = 50 [constexpr]
```

Galileo E5a symbol rate [bits/second].

Definition at line 41 of file [Galileo\\_E5a.h](#).

**9.13.6.4.407 GALILEO\_E5B\_CODE\_CHIP\_RATE\_CPS**

```
double GALILEO_E5B_CODE_CHIP_RATE_CPS = 1.023e7 [constexpr]
```

Galileo E5b code rate [chips/s].

Definition at line 34 of file [Galileo\\_E5b.h](#).



**9.13.6.4.408 GALILEO\_E5B\_CODE\_LENGTH\_CHIPS**

```
int32_t GALILEO_E5B_CODE_LENGTH_CHIPS = 10230 [constexpr]
```

Galileo E5b primary code length [chips].

Definition at line 39 of file [Galileo\\_E5b.h](#).

**9.13.6.4.409 GALILEO\_E5B\_CODE\_PERIOD\_MS**

```
int32_t GALILEO_E5B_CODE_PERIOD_MS = 1 [constexpr]
```

Galileo E5b primary code period [ms].

Definition at line 38 of file [Galileo\\_E5b.h](#).

**9.13.6.4.410 GALILEO\_E5B\_CODE\_PERIOD\_S**

```
double GALILEO_E5B_CODE_PERIOD_S = 0.001 [constexpr]
```

Galileo E5b primary code period [s].

Definition at line 37 of file [Galileo\\_E5b.h](#).

**9.13.6.4.411 GALILEO\_E5B\_FREQ\_HZ**

```
double GALILEO_E5B_FREQ_HZ = FREQ7 [constexpr]
```

Galileo E5b carrier frequency [Hz].

Definition at line 33 of file [Galileo\\_E5b.h](#).

**9.13.6.4.412 GALILEO\_E5B\_HISTORY\_DEEP**

```
int32_t GALILEO_E5B_HISTORY_DEEP = 100 [constexpr]
```

Definition at line 47 of file [Galileo\\_E5b.h](#).

**9.13.6.4.413 GALILEO\_E5B\_I\_PRIMARY\_CODE**

```
char GALILEO_E5B_I_PRIMARY_CODE[GALILEO_E5B_NUMBER_OF_CODES][2559] [constexpr]
```

Definition at line 57 of file [Galileo\\_E5b.h](#).

**9.13.6.4.414 GALILEO\_E5B\_I\_PRIMARY\_CODE\_STR\_LENGTH**

```
size_t GALILEO_E5B_I_PRIMARY_CODE_STR_LENGTH = 2558 [constexpr]
```

Definition at line 56 of file [Galileo\\_E5b.h](#).

**9.13.6.4.415 GALILEO\_E5B\_I\_SECONDARY\_CODE**

```
char GALILEO_E5B_I_SECONDARY_CODE[5] = "1110" [constexpr]
```

Definition at line 53 of file [Galileo\\_E5b.h](#).

**9.13.6.4.416 GALILEO\_E5B\_I\_SECONDARY\_CODE\_LENGTH**

```
int32_t GALILEO_E5B_I_SECONDARY_CODE_LENGTH = 4 [constexpr]
```

Galileo E5b-I secondary code length [chips].

Definition at line 40 of file [Galileo\\_E5b.h](#).

**9.13.6.4.417 GALILEO\_E5B\_I\_TIERED\_CODE\_PERIOD\_S**

```
double GALILEO_E5B_I_TIERED_CODE_PERIOD_S = 0.004 [constexpr]
```

Galileo E5b-I tiered code period [s].

Definition at line 35 of file [Galileo\\_E5b.h](#).

**9.13.6.4.418 GALILEO\_E5B\_NUMBER\_OF\_CODES**

```
int32_t GALILEO_E5B_NUMBER_OF_CODES = 50 [constexpr]
```

Definition at line 43 of file [Galileo\\_E5b.h](#).

**9.13.6.4.419 GALILEO\_E5B\_OPT\_ACQ\_FS\_SPS**

```
uint32_t GALILEO_E5B_OPT_ACQ_FS_SPS = 10000000 [constexpr]
```

Sampling frequency that maximizes the acquisition SNR while using a non-multiple of chip rate.

Definition at line 50 of file [Galileo\\_E5b.h](#).

**9.13.6.4.420 GALILEO\_E5B\_Q\_PRIMARY\_CODE**

```
char GALILEO_E5B_Q_PRIMARY_CODE[GALILEO_E5B_NUMBER_OF_CODES][2559] [constexpr]
```

Definition at line 1812 of file [Galileo\\_E5b.h](#).

**9.13.6.4.421 GALILEO\_E5B\_Q\_PRIMARY\_CODE\_STR\_LENGTH**

```
size_t GALILEO_E5B_Q_PRIMARY_CODE_STR_LENGTH = 2558 [constexpr]
```

Definition at line 1811 of file [Galileo\\_E5b.h](#).

**9.13.6.4.422 GALILEO\_E5B\_Q\_SECONDARY\_CODE**

```
char GALILEO_E5B_Q_SECONDARY_CODE[GALILEO_E5B_NUMBER_OF_CODES][101] [constexpr]
```

Definition at line 3568 of file [Galileo\\_E5b.h](#).

**9.13.6.4.423 GALILEO\_E5B\_Q\_SECONDARY\_CODE\_LENGTH**

```
int32_t GALILEO_E5B_Q_SECONDARY_CODE_LENGTH = 100 [constexpr]
```

Galileo E5b-Q secondary code length [chips].

Definition at line 41 of file [Galileo\\_E5b.h](#).

**9.13.6.4.424 GALILEO\_E5B\_Q\_SECONDARY\_CODE\_STR\_LENGTH**

```
size_t GALILEO_E5B_Q_SECONDARY_CODE_STR_LENGTH = 100 [constexpr]
```

Definition at line 3567 of file [Galileo\\_E5b.h](#).

**9.13.6.4.425 GALILEO\_E5B\_Q\_TIERED\_CODE\_PERIOD\_S**

```
double GALILEO_E5B_Q_TIERED_CODE_PERIOD_S = 0.100 [constexpr]
```

Galileo E5b-Q tiered code period [s].

Definition at line 36 of file [Galileo\\_E5b.h](#).

**9.13.6.4.426 GALILEO\_E5B\_SYMBOL\_RATE\_BPS**

```
int32_t GALILEO_E5B_SYMBOL_RATE_BPS = 250 [constexpr]
```

Galileo E5b symbol rate [bits/second].

Definition at line 42 of file [Galileo\\_E5b.h](#).

**9.13.6.4.427 GALILEO\_E6\_B\_CODE\_CHIP\_RATE\_CPS**

```
double GALILEO_E6_B_CODE_CHIP_RATE_CPS = 5.115e6 [constexpr]
```

Galileo E6 B code rate [chips/s].

Definition at line 31 of file [Galileo\\_E6.h](#).

**9.13.6.4.428 GALILEO\_E6\_B\_CODE\_LENGTH\_CHIPS**

```
double GALILEO_E6_B_CODE_LENGTH_CHIPS = 5115.0 [constexpr]
```

Galileo E6 B code length [chips].

Definition at line 35 of file [Galileo\\_E6.h](#).

**9.13.6.4.429 GALILEO\_E6\_B\_PRIMARY\_CODE**

```
char GALILEO_E6_B_PRIMARY_CODE[GALILEO_E6_NUMBER_OF_CODES][1280] [constexpr]
```

Definition at line 45 of file [Galileo\\_E6.h](#).

**9.13.6.4.430 GALILEO\_E6\_B\_PRIMARY\_CODE\_STR\_LENGTH**

size\_t GALILEO\_E6\_B\_PRIMARY\_CODE\_STR\_LENGTH = 1279 [constexpr]  
Definition at line 44 of file [Galileo\\_E6.h](#).

**9.13.6.4.431 GALILEO\_E6\_C\_CODE\_CHIP\_RATE\_CPS**

double GALILEO\_E6\_C\_CODE\_CHIP\_RATE\_CPS = 5.115e6 [constexpr]  
Galileo E6 C code rate [chips/s].  
Definition at line 32 of file [Galileo\\_E6.h](#).

**9.13.6.4.432 GALILEO\_E6\_C\_CODE\_LENGTH\_CHIPS**

double GALILEO\_E6\_C\_CODE\_LENGTH\_CHIPS = 5115.0 [constexpr]  
Galileo E6 C code length [chips].  
Definition at line 36 of file [Galileo\\_E6.h](#).

**9.13.6.4.433 GALILEO\_E6\_C\_PRIMARY\_CODE**

char GALILEO\_E6\_C\_PRIMARY\_CODE[GALILEO\_E6\_NUMBER\_OF\_CODES][1280] [constexpr]  
Definition at line 950 of file [Galileo\\_E6.h](#).

**9.13.6.4.434 GALILEO\_E6\_C\_PRIMARY\_CODE\_STR\_LENGTH**

size\_t GALILEO\_E6\_C\_PRIMARY\_CODE\_STR\_LENGTH = 1279 [constexpr]  
Definition at line 949 of file [Galileo\\_E6.h](#).

**9.13.6.4.435 GALILEO\_E6\_C\_SECONDARY\_CODE**

char GALILEO\_E6\_C\_SECONDARY\_CODE[GALILEO\_E6\_NUMBER\_OF\_CODES][26] [constexpr]  
Definition at line 1855 of file [Galileo\\_E6.h](#).

**9.13.6.4.436 GALILEO\_E6\_C\_SECONDARY\_CODE\_LENGTH\_CHIPS**

double GALILEO\_E6\_C\_SECONDARY\_CODE\_LENGTH\_CHIPS = 100.0 [constexpr]  
Galileo E6 C secondary code length [chips].  
Definition at line 37 of file [Galileo\\_E6.h](#).

**9.13.6.4.437 GALILEO\_E6\_C\_SECONDARY\_CODE\_STR\_LENGTH**

size\_t GALILEO\_E6\_C\_SECONDARY\_CODE\_STR\_LENGTH = 25 [constexpr]  
Definition at line 1854 of file [Galileo\\_E6.h](#).

**9.13.6.4.438 GALILEO\_E6\_CODE\_PERIOD\_MS**

uint32\_t GALILEO\_E6\_CODE\_PERIOD\_MS = 1 [constexpr]  
Galileo E& B/C code period [ms].  
Definition at line 38 of file [Galileo\\_E6.h](#).

**9.13.6.4.439 GALILEO\_E6\_CODE\_PERIOD\_S**

double GALILEO\_E6\_CODE\_PERIOD\_S = 0.001 [constexpr]  
Galileo E6 code period [s].  
Definition at line 33 of file [Galileo\\_E6.h](#).

**9.13.6.4.440 GALILEO\_E6\_FREQ\_HZ**

double GALILEO\_E6\_FREQ\_HZ = [FREQ6](#) [constexpr]  
Galileo E6 carrier frequency [Hz].  
Definition at line 30 of file [Galileo\\_E6.h](#).

**9.13.6.4.441 GALILEO\_E6\_NUMBER\_OF\_CODES**

`int32_t GALILEO_E6_NUMBER_OF_CODES = 50 [constexpr]`  
Definition at line 40 of file [Galileo\\_E6.h](#).

**9.13.6.4.442 GALILEO\_E6\_OPT\_ACQ\_FS\_SPS**

`uint32_t GALILEO_E6_OPT_ACQ_FS_SPS = 10000000 [constexpr]`  
Definition at line 42 of file [Galileo\\_E6.h](#).

**9.13.6.4.443 GALILEO\_F**

`double GALILEO_F = -4.442807309e-10 [constexpr]`  
Constant,  $[s/(m)^{(1/2)}]$ . OS SIS ICD v2.0, pag. 47.  
Definition at line 36 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.444 GALILEO\_FNAV\_CODES\_PER\_PAGE**

`int32_t GALILEO_FNAV_CODES_PER_PAGE = 10000 [constexpr]`  
Definition at line 59 of file [Galileo\\_E5a.h](#).

**9.13.6.4.445 GALILEO\_FNAV\_CODES\_PER\_PREAMBLE**

`int32_t GALILEO_FNAV_CODES_PER_PREAMBLE = 240 [constexpr]`  
Definition at line 56 of file [Galileo\\_E5a.h](#).

**9.13.6.4.446 GALILEO\_FNAV\_CODES\_PER\_SYMBOL**

`int32_t GALILEO_FNAV_CODES_PER_SYMBOL = 20 [constexpr]`  
Definition at line 55 of file [Galileo\\_E5a.h](#).

**9.13.6.4.447 GALILEO\_FNAV\_DATA\_FRAME\_BITS**

`int32_t GALILEO_FNAV_DATA_FRAME_BITS = 214 [constexpr]`  
Definition at line 65 of file [Galileo\\_E5a.h](#).

**9.13.6.4.448 GALILEO\_FNAV\_DATA\_FRAME\_BYTES**

`int32_t GALILEO_FNAV_DATA_FRAME_BYTES = 27 [constexpr]`  
Definition at line 66 of file [Galileo\\_E5a.h](#).

**9.13.6.4.449 GALILEO\_FNAV\_INTERLEAVER\_COLS**

`int32_t GALILEO_FNAV_INTERLEAVER_COLS = 61 [constexpr]`  
Definition at line 62 of file [Galileo\\_E5a.h](#).

**9.13.6.4.450 GALILEO\_FNAV\_INTERLEAVER\_ROWS**

`int32_t GALILEO_FNAV_INTERLEAVER_ROWS = 8 [constexpr]`  
Definition at line 61 of file [Galileo\\_E5a.h](#).

**9.13.6.4.451 GALILEO\_FNAV\_PAGE\_TYPE\_BITS**

`int32_t GALILEO_FNAV_PAGE_TYPE_BITS = 6 [constexpr]`  
Definition at line 63 of file [Galileo\\_E5a.h](#).

**9.13.6.4.452 GALILEO\_FNAV\_PREAMBLE**

`char GALILEO_FNAV_PREAMBLE[13] = "101101110000" [constexpr]`  
Definition at line 68 of file [Galileo\\_E5a.h](#).

**9.13.6.4.453 GALILEO\_FNAV\_PREAMBLE\_LENGTH\_BITS**

```
int32_t GALILEO_FNAV_PREAMBLE_LENGTH_BITS = 12 [constexpr]
```

Definition at line 54 of file [Galileo\\_E5a.h](#).

**9.13.6.4.454 GALILEO\_FNAV\_SECONDS\_PER\_PAGE**

```
int32_t GALILEO_FNAV_SECONDS_PER_PAGE = 10 [constexpr]
```

Definition at line 58 of file [Galileo\\_E5a.h](#).

**9.13.6.4.455 GALILEO\_FNAV\_SYMBOLS\_PER\_PAGE**

```
int32_t GALILEO_FNAV_SYMBOLS_PER_PAGE = 500 [constexpr]
```

Definition at line 57 of file [Galileo\\_E5a.h](#).

**9.13.6.4.456 GALILEO\_GM**

```
double GALILEO_GM = 3.986004418e14 [constexpr]
```

Geocentric gravitational constant[m<sup>3</sup>/s<sup>2</sup>], OS SIS ICD v2.0, pag. 44.  
Definition at line 35 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.457 GALILEO\_INAV\_ENCODED\_SSP1**

```
char GALILEO_INAV_ENCODED_SSP1[17] = "1110100100100101" [constexpr]
```

Definition at line 303 of file [Galileo\\_INAV.h](#).

**9.13.6.4.458 GALILEO\_INAV\_ENCODED\_SSP2**

```
char GALILEO_INAV_ENCODED_SSP2[17] = "0110110001001110" [constexpr]
```

Definition at line 304 of file [Galileo\\_INAV.h](#).

**9.13.6.4.459 GALILEO\_INAV\_ENCODED\_SSP3**

```
char GALILEO_INAV_ENCODED_SSP3[17] = "1101000000111110" [constexpr]
```

Definition at line 305 of file [Galileo\\_INAV.h](#).

**9.13.6.4.460 GALILEO\_INAV\_INTERLEAVER\_COLS**

```
int32_t GALILEO_INAV_INTERLEAVER_COLS = 30 [constexpr]
```

Definition at line 43 of file [Galileo\\_INAV.h](#).

**9.13.6.4.461 GALILEO\_INAV\_INTERLEAVER\_ROWS**

```
int32_t GALILEO_INAV_INTERLEAVER_ROWS = 8 [constexpr]
```

Definition at line 42 of file [Galileo\\_INAV.h](#).

**9.13.6.4.462 GALILEO\_INAV\_PAGE\_PART\_MS**

```
int32_t GALILEO_INAV_PAGE_PART_MS = 1000 [constexpr]
```

Definition at line 40 of file [Galileo\\_INAV.h](#).

**9.13.6.4.463 GALILEO\_INAV\_PAGE\_PART\_SECONDS**

```
int32_t GALILEO_INAV_PAGE_PART_SECONDS = 1 [constexpr]
```

Definition at line 39 of file [Galileo\\_INAV.h](#).

**9.13.6.4.464 GALILEO\_INAV\_PAGE\_PART\_SYMBOLS**

```
int32_t GALILEO_INAV_PAGE_PART_SYMBOLS = 250 [constexpr]
```

Each Galileo INAV pages are composed of two parts (even and odd) each of 250 symbols, including preamble. See Galileo ICD 4.3.2.

Definition at line 38 of file [Galileo\\_INAV.h](#).

**9.13.6.4.465 GALILEO\_INAV\_PAGE\_PART\_WITH\_PREABLE\_SECONDS**

```
double GALILEO_INAV_PAGE_PART_WITH_PREABLE_SECONDS = 2.04 [constexpr]
```

Page Duration + (Galileo I/NAV Preamble bits)\*(Galileo E5b-I tiered Code Period(seconds))

Definition at line 34 of file [Galileo\\_INAV.h](#).

**9.13.6.4.466 GALILEO\_INAV\_PAGE\_SECONDS**

```
int32_t GALILEO_INAV_PAGE_SECONDS = 2 [constexpr]
```

Definition at line 41 of file [Galileo\\_INAV.h](#).

**9.13.6.4.467 GALILEO\_INAV\_PAGE\_SYMBOLS**

```
uint32_t GALILEO_INAV_PAGE_SYMBOLS = 500 [constexpr]
```

The complete Galileo INAV page length.

Definition at line 35 of file [Galileo\\_INAV.h](#).

**9.13.6.4.468 GALILEO\_INAV\_PLAIN\_SSP1**

```
char GALILEO_INAV_PLAIN_SSP1[9] = "00000100" [constexpr]
```

Definition at line 300 of file [Galileo\\_INAV.h](#).

**9.13.6.4.469 GALILEO\_INAV\_PLAIN\_SSP2**

```
char GALILEO_INAV_PLAIN_SSP2[9] = "00101011" [constexpr]
```

Definition at line 301 of file [Galileo\\_INAV.h](#).

**9.13.6.4.470 GALILEO\_INAV\_PLAIN\_SSP3**

```
char GALILEO_INAV_PLAIN_SSP3[9] = "00101111" [constexpr]
```

Definition at line 302 of file [Galileo\\_INAV.h](#).

**9.13.6.4.471 GALILEO\_INAV\_PREAMBLE**

```
char GALILEO_INAV_PREAMBLE[11] = "0101100000" [constexpr]
```

Definition at line 49 of file [Galileo\\_INAV.h](#).

**9.13.6.4.472 GALILEO\_INAV\_PREAMBLE\_LENGTH\_BITS**

```
int32_t GALILEO_INAV_PREAMBLE_LENGTH_BITS = 10 [constexpr]
```

Definition at line 36 of file [Galileo\\_INAV.h](#).

**9.13.6.4.473 GALILEO\_INAV\_PREAMBLE\_PERIOD\_SYMBOLS**

```
int32_t GALILEO_INAV_PREAMBLE_PERIOD_SYMBOLS = 250 [constexpr]
```

Definition at line 37 of file [Galileo\\_INAV.h](#).

**9.13.6.4.474 GALILEO\_ISM\_CRC\_DATA\_BITS**

```
int32_t GALILEO_ISM_CRC_DATA_BITS = 96 [constexpr]
```

Definition at line 291 of file [Galileo\\_INAV.h](#).

**9.13.6.4.475 GALILEO\_ISM\_CRC\_DATA\_BYTES**

`int32_t GALILEO_ISM_CRC_DATA_BYTES = 12 [constexpr]`  
Definition at line 292 of file [Galileo\\_INAV.h](#).

**9.13.6.4.476 GALILEO\_PAGE\_TYPE\_BITS**

`int32_t GALILEO_PAGE_TYPE_BITS = 6 [constexpr]`  
Definition at line 45 of file [Galileo\\_INAV.h](#).

**9.13.6.4.477 GALILEO\_TELEMETRY\_RATE\_BITS\_SECOND**

`int32_t GALILEO_TELEMETRY_RATE_BITS_SECOND = 250 [constexpr]`  
Definition at line 44 of file [Galileo\\_INAV.h](#).

**9.13.6.4.478 GLONASS\_C20**

`double GLONASS_C20 = -1082.63e-6 [constexpr]`  
Second zonal coefficient of spherical harmonic expansion.  
Definition at line 45 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.479 GLONASS\_CA\_NBR\_SATS**

`int32_t GLONASS_CA_NBR_SATS = 24 [constexpr]`  
Definition at line 84 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.480 GLONASS\_EARTH\_INCLINATION**

`double GLONASS_EARTH_INCLINATION = 0.000409148809899e3 [constexpr]`  
Mean inclination of ecliptic to equator (23 deg 26 min 33 sec) [rad].  
Definition at line 47 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.481 GLONASS\_EARTH\_RADIUS**

`double GLONASS_EARTH_RADIUS = 6378.136 [constexpr]`  
Equatorial radius of Earth [km].  
Definition at line 46 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.482 GLONASS\_F\_M\_A**

`double GLONASS_F_M_A = 0.35e9 [constexpr]`  
Gravitational constant of atmosphere [ $\text{m}^3/\text{s}^2$ ].  
Definition at line 35 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.483 GLONASS\_FLATTENING**

`double GLONASS_FLATTENING = 1.0 / 29825784.0 [constexpr]`  
Flattening parameter.  
Definition at line 37 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.484 GLONASS\_GM**

`double GLONASS_GM = 398600.44e9 [constexpr]`  
Universal gravitational constant times the mass of the Earth, [ $\text{m}^3/\text{s}^2$ ].  
Definition at line 40 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.485 GLONASS\_GNAV\_CRC\_I\_INDEX**

```
const std::vector<int32_t> GLONASS_GNAV_CRC_I_INDEX {9, 10, 12, 13, 15, 17, 19, 20, 22, 24,
26, 28, 30, 32, 34, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 66, 68,
70, 72, 74, 76, 78, 80, 82, 84}
```

Definition at line 232 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.486 GLONASS\_GNAV\_CRC\_J\_INDEX**

```
const std::vector<int32_t> GLONASS_GNAV_CRC_J_INDEX {9, 11, 12, 14, 15, 18, 19, 21, 22, 25,
26, 29, 30, 33, 34, 36, 37, 40, 41, 44, 45, 48, 49, 52, 53, 56, 57, 60, 61, 64, 65, 67, 68,
71, 72, 75, 76, 79, 80, 83, 84}
```

Definition at line 233 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.487 GLONASS\_GNAV\_CRC\_K\_INDEX**

```
const std::vector<int32_t> GLONASS_GNAV_CRC_K_INDEX {10, 11, 12, 16, 17, 18, 19, 23, 24, 25,
26, 31, 32, 33, 34, 38, 39, 40, 41, 46, 47, 48, 49, 54, 55, 56, 57, 62, 63, 64, 65, 69, 70,
71, 72, 77, 78, 79, 80, 85}
```

Definition at line 234 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.488 GLONASS\_GNAV\_CRC\_L\_INDEX**

```
const std::vector<int32_t> GLONASS_GNAV_CRC_L_INDEX {13, 14, 15, 16, 17, 18, 19, 27, 28, 29,
30, 31, 32, 33, 34, 42, 43, 44, 45, 46, 47, 48, 49, 58, 59, 60, 61, 62, 63, 64, 65, 73, 74,
75, 76, 77, 78, 79, 80}
```

Definition at line 235 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.489 GLONASS\_GNAV\_CRC\_M\_INDEX**

```
const std::vector<int32_t> GLONASS_GNAV_CRC_M_INDEX {20, 21, 22, 23, 24, 25, 26, 27, 28, 29,
30, 31, 32, 33, 34, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 81, 82,
83, 84, 85}
```

Definition at line 236 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.490 GLONASS\_GNAV\_CRC\_N\_INDEX**

```
const std::vector<int32_t> GLONASS_GNAV_CRC_N_INDEX {35, 36, 37, 38, 39, 40, 41, 42, 43, 44,
45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65}
```

Definition at line 237 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.491 GLONASS\_GNAV\_CRC\_P\_INDEX**

```
const std::vector<int32_t> GLONASS_GNAV_CRC_P_INDEX {66, 67, 68, 69, 70, 71, 72, 73, 74, 75,
76, 77, 78, 79, 80, 81, 82, 83, 84, 85}
```

Definition at line 238 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.492 GLONASS\_GNAV\_CRC\_Q\_INDEX**

```
const std::vector<int32_t> GLONASS_GNAV_CRC_Q_INDEX {9, 10, 11, 12, 13, 14, 15, 16, 17, 18,
19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41,
42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64,
65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85}
```

Definition at line 239 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.493 GLONASS\_GNAV\_DATA\_SYMBOLS**

```
int32_t GLONASS_GNAV_DATA_SYMBOLS = 1700 [constexpr]
```

Definition at line 104 of file [GLONASS\\_L1\\_L2\\_CA.h](#).



**9.13.6.4.494 GLONASS\_GNAV\_ECC\_LOCATOR**

```
const std::vector<int32_t> GLONASS_GNAV_ECC_LOCATOR {0, 0, 1, 8, 2, 9, 10, 11, 3, 12, 13, 14,
15, 16, 17, 18, 4, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 5, 34, 35, 36,
37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59,
60, 61, 62, 63, 64, 6, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82,
83, 84}
```

Definition at line 240 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.495 GLONASS\_GNAV\_HAMMING\_CODE\_BITS**

```
int32_t GLONASS_GNAV_HAMMING_CODE_BITS = 8 [constexpr]
```

Number of bits in hamming code sequence of GNAV message.

Definition at line 103 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.496 GLONASS\_GNAV\_PREAMBLE\_DURATION\_S**

```
double GLONASS_GNAV_PREAMBLE_DURATION_S = 0.300 [constexpr]
```

Definition at line 93 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.497 GLONASS\_GNAV\_PREAMBLE\_LENGTH\_BITS**

```
int32_t GLONASS_GNAV_PREAMBLE_LENGTH_BITS = 30 [constexpr]
```

Definition at line 94 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.498 GLONASS\_GNAV\_PREAMBLE\_LENGTH\_SYMBOLS**

```
int32_t GLONASS_GNAV_PREAMBLE_LENGTH_SYMBOLS = 300 [constexpr]
```

Definition at line 95 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.499 GLONASS\_GNAV\_PREAMBLE\_PERIOD\_SYMBOLS**

```
int32_t GLONASS_GNAV_PREAMBLE_PERIOD_SYMBOLS = 2000 [constexpr]
```

Definition at line 96 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.500 GLONASS\_GNAV\_STRING\_BITS**

```
int32_t GLONASS_GNAV_STRING_BITS = 85 [constexpr]
```

Number of bits per string in the GNAV message (85 data bits + 30 time mark bits) [bits].

Definition at line 102 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.501 GLONASS\_GNAV\_STRING\_SYMBOLS**

```
int32_t GLONASS_GNAV_STRING_SYMBOLS = 2000 [constexpr]
```

Number of bits per string in the GNAV message (85 data bits + 30 time mark bits) [bits].

Definition at line 101 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.502 GLONASS\_GNAV\_TELEMETRY\_RATE\_BITS\_SECOND**

```
int32_t GLONASS_GNAV_TELEMETRY_RATE_BITS_SECOND = 50 [constexpr]
```

NAV message bit rate [bits/s].

Definition at line 97 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.503 GLONASS\_GNAV\_TELEMETRY\_RATE\_SYMBOLS\_SECOND**

```
int32_t GLONASS_GNAV_TELEMETRY_RATE_SYMBOLS_SECOND = GLONASS_GNAV_TELEMETRY_RATE_BITS_SECOND *
GLONASS_GNAV_TELEMETRY_SYMBOLS_PER_BIT [constexpr]
```

NAV message bit rate [symbols/s].

Definition at line 100 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.504 GLONASS\_GNAV\_TELEMETRY\_SYMBOLS\_PER\_BIT**

```
int32_t GLONASS_GNAV_TELEMETRY_SYMBOLS_PER_BIT = 10 [constexpr]
```

Definition at line 98 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.505 GLONASS\_GNAV\_TELEMETRY\_SYMBOLS\_PER\_PREAMBLE\_BIT**

```
int32_t GLONASS_GNAV_TELEMETRY_SYMBOLS_PER_PREAMBLE_BIT = 10 [constexpr]
```

Definition at line 99 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.506 GLONASS\_GRAVITY**

```
double GLONASS_GRAVITY = 97803284.0 [constexpr]
```

Equatorial acceleration of gravity [mGal].

Definition at line 38 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.507 GLONASS\_GRAVITY\_CORRECTION**

```
double GLONASS_GRAVITY_CORRECTION = 0.87 [constexpr]
```

Correction to acceleration of gravity at sea-level due to Atmosphere[uGal].

Definition at line 39 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.508 GLONASS\_J2**

```
double GLONASS_J2 = 1082625.75e-9 [constexpr]
```

Second zonal harmonic of the geopotential.

Definition at line 40 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.509 GLONASS\_J4**

```
double GLONASS_J4 = -2370.89e-9 [constexpr]
```

Fourth zonal harmonic of the geopotential.

Definition at line 41 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.510 GLONASS\_J6**

```
double GLONASS_J6 = 6.08e-9 [constexpr]
```

Sixth zonal harmonic of the geopotential.

Definition at line 42 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.511 GLONASS\_J8**

```
double GLONASS_J8 = 1.40e-11 [constexpr]
```

Eighth zonal harmonic of the geopotential.

Definition at line 43 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.512 GLONASS\_L1\_CA\_CHIP\_PERIOD\_S**

```
double GLONASS_L1_CA_CHIP_PERIOD_S = 1.9569e-06 [constexpr]
```

GLONASS L1 C/A chip period [seconds].

Definition at line 81 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.513 GLONASS\_L1\_CA\_CODE\_LENGTH\_CHIPS**

```
double GLONASS_L1_CA_CODE_LENGTH_CHIPS = 511.0 [constexpr]
```

GLONASS L1 C/A code length [chips].

Definition at line 79 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.514 GLONASS\_L1\_CA\_CODE\_PERIOD\_S**

double GLONASS\_L1\_CA\_CODE\_PERIOD\_S = 0.001 [constexpr]  
GLONASS L1 C/A code period [seconds].  
Definition at line 80 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.515 GLONASS\_L1\_CA\_CODE\_RATE\_CPS**

double GLONASS\_L1\_CA\_CODE\_RATE\_CPS = 0.511e6 [constexpr]  
GLONASS L1 C/A code rate [chips/s].  
Definition at line 78 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.516 GLONASS\_L1\_CA\_DFREQ\_HZ**

double GLONASS\_L1\_CA\_DFREQ\_HZ = DFRQ1\_GLO [constexpr]  
Freq Bias for GLONASS L1 [Hz].  
Definition at line 77 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.517 GLONASS\_L1\_CA\_FREQ\_HZ**

double GLONASS\_L1\_CA\_FREQ\_HZ = FREQ1\_GLO [constexpr]  
L1 [Hz].  
Definition at line 76 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.518 GLONASS\_L1\_CA\_HISTORY\_DEEP**

int32\_t GLONASS\_L1\_CA\_HISTORY\_DEEP = 100 [constexpr]  
Definition at line 87 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.519 GLONASS\_L1\_CA\_SYMBOL\_RATE\_BPS**

double GLONASS\_L1\_CA\_SYMBOL\_RATE\_BPS = 1000.0 [constexpr]  
Definition at line 82 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.520 GLONASS\_L2\_CA\_CHIP\_PERIOD\_S**

double GLONASS\_L2\_CA\_CHIP\_PERIOD\_S = 1.9569e-06 [constexpr]  
GLONASS L1 C/A chip period [seconds].  
Definition at line 73 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.521 GLONASS\_L2\_CA\_CODE\_LENGTH\_CHIPS**

double GLONASS\_L2\_CA\_CODE\_LENGTH\_CHIPS = 511.0 [constexpr]  
GLONASS L1 C/A code length [chips].  
Definition at line 71 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.522 GLONASS\_L2\_CA\_CODE\_PERIOD\_S**

double GLONASS\_L2\_CA\_CODE\_PERIOD\_S = 0.001 [constexpr]  
GLONASS L1 C/A code period [seconds].  
Definition at line 72 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.523 GLONASS\_L2\_CA\_CODE\_RATE\_CPS**

double GLONASS\_L2\_CA\_CODE\_RATE\_CPS = 0.511e6 [constexpr]  
GLONASS L1 C/A code rate [chips/s].  
Definition at line 70 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.524 GLONASS\_L2\_CA\_DFREQ\_HZ**

```
double GLONASS_L2_CA_DFREQ_HZ = DFRQ2_GLO [constexpr]
```

Freq Bias for GLONASS L1 [Hz].

Definition at line 69 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.525 GLONASS\_L2\_CA\_FREQ\_HZ**

```
double GLONASS_L2_CA_FREQ_HZ = FREQ2_GLO [constexpr]
```

L2 [Hz].

Definition at line 68 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.526 GLONASS\_L2\_CA\_SYMBOL\_RATE\_BPS**

```
double GLONASS_L2_CA_SYMBOL_RATE_BPS = 1000.0 [constexpr]
```

Definition at line 74 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.527 GLONASS\_LEAP\_SECONDS**

```
double GLONASS_LEAP_SECONDS[19][7] [constexpr]
```

**Initial value:**

```
= {
    {2017, 1, 1, 0, 0, 0, -18},
    {2015, 7, 1, 0, 0, 0, -17},
    {2012, 7, 1, 0, 0, 0, -16},
    {2009, 1, 1, 0, 0, 0, -15},
    {2006, 1, 1, 0, 0, 0, -14},
    {1999, 1, 1, 0, 0, 0, -13},
    {1997, 7, 1, 0, 0, 0, -12},
    {1996, 1, 1, 0, 0, 0, -11},
    {1994, 7, 1, 0, 0, 0, -10},
    {1993, 7, 1, 0, 0, 0, -9},
    {1992, 7, 1, 0, 0, 0, -8},
    {1991, 1, 1, 0, 0, 0, -7},
    {1990, 1, 1, 0, 0, 0, -6},
    {1988, 1, 1, 0, 0, 0, -5},
    {1985, 7, 1, 0, 0, 0, -4},
    {1983, 7, 1, 0, 0, 0, -3},
    {1982, 7, 1, 0, 0, 0, -2},
    {1981, 7, 1, 0, 0, 0, -1},
    {}
}
```

Record of leap seconds definition for GLOT to GPST conversion and vice versa.

Each entry is defined by an array of 7 elements consisting of yr,month,day,hr,min,sec,utc-gpst

**Note**

Ideally should use leap seconds definitions of rtklib GLONASS SV's orbital slots PRN = (orbital\_slot - 1)

Definition at line 111 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.528 GLONASS\_MOON\_ECCENTRICITY**

```
double GLONASS_MOON_ECCENTRICITY = 0.054900489 [constexpr]
```

Eccentricity of lunar orbit.

Definition at line 58 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.529 GLONASS\_MOON\_GM**

```
double GLONASS_MOON_GM = 4902.835 [constexpr]
```

Lunar gravitational constant [km<sup>3</sup>/s<sup>2</sup>].

Definition at line 56 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.530 GLONASS\_MOON\_INCLINATION**

```
double GLONASS_MOON_INCLINATION = 0.000089803977407e3 [constexpr]
```

Inclination of lunar orbit to ecliptic plane (5 deg 08 min 43.4 sec) [rad].

Definition at line 59 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

**9.13.6.4.531 GLONASS\_MOON\_OMEGA\_0**

```
double GLONASS_MOON_OMEGA_0 = 0.004523601514852e3 [constexpr]
(259 deg 10 min 59.79 sec) [rad]
Definition at line 54 of file GLONASS\_L1\_L2\_CA.h.
```

**9.13.6.4.532 GLONASS\_MOON\_OMEGA\_1**

```
double GLONASS_MOON_OMEGA_1 = -0.033757146246552e3 [constexpr]
(-1934 deg 08 min 31.23 sec) [rad]
Definition at line 55 of file GLONASS\_L1\_L2\_CA.h.
```

**9.13.6.4.533 GLONASS\_MOON\_Q0**

```
double GLONASS_MOON_Q0 = -0.001115184961435e3 [constexpr]
(-63 deg 53 min 43.41 sec) [rad]
Definition at line 52 of file GLONASS\_L1\_L2\_CA.h.
```

**9.13.6.4.534 GLONASS\_MOON\_Q1**

```
double GLONASS_MOON_Q1 = 8.328691103668023e3 [constexpr]
(477198 deg 50 min 56.79 sec) [rad]
Definition at line 53 of file GLONASS\_L1\_L2\_CA.h.
```

**9.13.6.4.535 GLONASS\_MOON\_SEMI\_MAJOR\_AXIS**

```
double GLONASS_MOON_SEMI_MAJOR_AXIS = 3.84385243e5 [constexpr]
Semi-major axis of lunar orbit [km];.
Definition at line 57 of file GLONASS\_L1\_L2\_CA.h.
```

**9.13.6.4.536 GLONASS\_OMEGA\_EARTH\_DOT**

```
double GLONASS_OMEGA_EARTH_DOT = 7.292115e-5 [constexpr]
Earth rotation rate, [rad/s] ICD L1, L2 GLONASS Edition 5.1 2008 pag. 55.
Definition at line 39 of file MATH\_CONSTANTS.h.
```

**9.13.6.4.537 GLONASS\_PRN**

```
const std::map<uint32_t, int32_t> GLONASS_PRN
Definition at line 133 of file GLONASS\_L1\_L2\_CA.h.
```

**9.13.6.4.538 GLONASS\_SEMI\_MAJOR\_AXIS**

```
double GLONASS_SEMI_MAJOR_AXIS = 6378136 [constexpr]
Semi-major axis of Earth [m].
Definition at line 36 of file GLONASS\_L1\_L2\_CA.h.
```

**9.13.6.4.539 GLONASS\_SUN\_ECCENTRICITY**

```
double GLONASS_SUN_ECCENTRICITY = 0.016719 [constexpr]
Eccentricity of solar orbit.
Definition at line 66 of file GLONASS\_L1\_L2\_CA.h.
```

**9.13.6.4.540 GLONASS\_SUN\_GM**

```
double GLONASS_SUN_GM = 0.1325263e12 [constexpr]
Solar gravitational constant [km^3/s^2].
Definition at line 64 of file GLONASS\_L1\_L2\_CA.h.
```

#### 9.13.6.4.541 GLONASS\_SUN\_OMEGA

```
double GLONASS_SUN_OMEGA = 0.004908229466869e3 [constexpr]
```

TODO What is this operation in the seconds with T?(281 deg 13 min 15.0 + 6189.03 x T sec) [rad].

Definition at line 61 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

#### 9.13.6.4.542 GLONASS\_SUN\_Q0

```
double GLONASS_SUN_Q0 = 0.006256583774423e3 [constexpr]
```

(358 deg 28 min 33.04 sec) [rad]

Definition at line 62 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

#### 9.13.6.4.543 GLONASS\_SUN\_Q1

```
double GLONASS_SUN_Q1 = 0e3 [constexpr]
```

TODO Why is the value greater than 60?(129596579.10 sec) [rad].

Definition at line 63 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

#### 9.13.6.4.544 GLONASS\_SUN\_SEMI\_MAJOR\_AXIS

```
double GLONASS_SUN_SEMI_MAJOR_AXIS = 1.49598e8 [constexpr]
```

Semi-major axis of solar orbit [km];.

Definition at line 65 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

#### 9.13.6.4.545 GLONASS\_TAU\_0

```
double GLONASS_TAU_0 = -0.005835151531174e3 [constexpr]
```

(-334 deg 19 min 46.40 sec) [rad];

Definition at line 49 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

#### 9.13.6.4.546 GLONASS\_TAU\_1

```
double GLONASS_TAU_1 = 0.071018041257371e3 [constexpr]
```

(4069 deg 02 min 02.52 sec) [rad];

Definition at line 50 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

#### 9.13.6.4.547 GLONASS\_U0

```
double GLONASS_U0 = 62636861.4 [constexpr]
```

Normal potential at surface of common terrestrial ellipsoid [m<sup>2</sup>/s<sup>2</sup>].

Definition at line 44 of file [GLONASS\\_L1\\_L2\\_CA.h](#).

#### 9.13.6.4.548 GNSS\_OMEGA\_EARTH\_DOT

```
double GNSS_OMEGA_EARTH_DOT = 7.2921151467e-5 [constexpr]
```

Default Earth rotation rate, [rad/s].

Definition at line 26 of file [MATH\\_CONSTANTS.h](#).

#### 9.13.6.4.549 GNSS\_PI

```
double GNSS_PI = 3.1415926535898 [constexpr]
```

pi constant as defined for GNSS

Definition at line 47 of file [MATH\\_CONSTANTS.h](#).

#### 9.13.6.4.550 GPS\_CA\_PREAMBLE

```
char GPS_CA_PREAMBLE[9] = "10001011" [constexpr]
```

Definition at line 72 of file [GPS\\_L1\\_CA.h](#).

int32\_t GPS\_CA\_PREAMBLE\_DURATION\_MS = 160 [constexpr]  
Definition at line 62 of file [GPS\\_L1\\_CA.h](#).

double GPS\_CA\_PREAMBLE\_DURATION\_S = 0.160 [constexpr]  
Definition at line 59 of file GPS\_L1\_CA.h.

int32\_t GPS\_CA\_PREAMBLE\_LENGTH\_BITS = 8 [constexpr]  
Definition at line 60 of file GPS\_L1\_CA.h.

int32\_t GPS\_CA\_PREAMBLE\_LENGTH\_SYMBOLS = 160 [constexpr]  
Definition at line 61 of file GPS\_L1\_CA.h.

[illegible]

int32\_t GPS\_CA\_TELEMETRY\_RATE\_BITS\_SECOND = 50 [constexpr]  
NAV message bit rate [bits/s].  
Definition at line 63 of file [GPS L1 CA.h](#).

```
int32_t GPS_CA_TELEMETRY_RATE_SYMBOLS_SECOND = GPS_CA_TELEMETRY_RATE_BITS_SECOND * GPS_CA_↵
TELEMETRY_SYMBOLS_PER_BIT [constexpr]
NAV message bit rate [symbols/s].
Definition at line 65 of file GPS L1 CA.h.
```

int32\_t GPS\_CA\_TELEMETRY\_SYMBOLS\_PER\_BIT = 20 [constexpr]  
Definition at line 64 of file GPS\_L1\_CA.h.

int32\_t GPS\_CNAV\_DATA\_PAGE\_BITS = 300 [constexpr]  
Definition at line 35 of file [GPS\\_CNAV.h](#).

```
double GPS_F = -4.442807633e-10 [constexpr]
Constant, [s/(m)^(1/2)], IS-GPS-200M, 20.3.3.3.1.
Definition at line 32 of file MATH_CONSTANTS.h.
```

```
double GPS_GM = 3.986005e14  [constexpr]
Universal gravitational constant times the mass of the Earth, [m^3/s^2] IS-GPS-200M, 20.3.3.3.1.
Definition at line 31 of file MATH_CONSTANTS.h.
```

#### 9.13.6.4.562 GPS\_L1\_CA\_BIT\_PERIOD\_MS

```
uint32_t GPS_L1_CA_BIT_PERIOD_MS = 20U [constexpr]
```

GPS L1 C/A bit period [ms].

Definition at line 40 of file [GPS\\_L1\\_CA.h](#).

#### 9.13.6.4.563 GPS\_L1\_CA\_CHIP\_PERIOD\_S

```
double GPS_L1_CA_CHIP_PERIOD_S = 9.7752e-07 [constexpr]
```

GPS L1 C/A chip period [seconds].

Definition at line 38 of file [GPS\\_L1\\_CA.h](#).

#### 9.13.6.4.564 GPS\_L1\_CA\_CODE\_LENGTH\_CHIPS

```
double GPS_L1_CA_CODE_LENGTH_CHIPS = 1023.0 [constexpr]
```

GPS L1 C/A code length [chips].

Definition at line 36 of file [GPS\\_L1\\_CA.h](#).

#### 9.13.6.4.565 GPS\_L1\_CA\_CODE\_PERIOD\_MS

```
uint32_t GPS_L1_CA_CODE_PERIOD_MS = 1U [constexpr]
```

GPS L1 C/A code period [ms].

Definition at line 39 of file [GPS\\_L1\\_CA.h](#).

#### 9.13.6.4.566 GPS\_L1\_CA\_CODE\_PERIOD\_S

```
double GPS_L1_CA_CODE_PERIOD_S = 0.001 [constexpr]
```

GPS L1 C/A code period [seconds].

Definition at line 37 of file [GPS\\_L1\\_CA.h](#).

#### 9.13.6.4.567 GPS\_L1\_CA\_CODE\_RATE\_CPS

```
double GPS_L1_CA_CODE_RATE_CPS = 1.023e6 [constexpr]
```

GPS L1 C/A code rate [chips/s].

Definition at line 35 of file [GPS\\_L1\\_CA.h](#).

#### 9.13.6.4.568 GPS\_L1\_CA\_HISTORY\_DEEP

```
int32_t GPS_L1_CA_HISTORY_DEEP = 100 [constexpr]
```

Definition at line 56 of file [GPS\\_L1\\_CA.h](#).

#### 9.13.6.4.569 GPS\_L1\_CA\_OPT\_ACQ\_FS\_SPS

```
uint32_t GPS_L1_CA_OPT_ACQ_FS_SPS = 2000000 [constexpr]
```

Sampling frequency that maximizes the acquisition SNR while using a non-multiple of chip rate.

Definition at line 53 of file [GPS\\_L1\\_CA.h](#).

#### 9.13.6.4.570 GPS\_L1\_FREQ\_HZ

```
double GPS_L1_FREQ_HZ = FREQ1 [constexpr]
```

L1 [Hz].

Definition at line 34 of file [GPS\\_L1\\_CA.h](#).

#### 9.13.6.4.571 GPS\_L2\_CNAV\_DATA\_PAGE\_BITS

```
int32_t GPS_L2_CNAV_DATA_PAGE_BITS = 300 [constexpr]
```

GPS L2 CNAV page length, including preamble and CRC [bits].

Definition at line 42 of file [GPS\\_L2C.h](#).



**9.13.6.4.572 GPS\_L2\_CNAV\_DATA\_PAGE\_DURATION\_S**

`int32_t GPS_L2_CNAV_DATA_PAGE_DURATION_S = 12 [constexpr]`  
Definition at line 46 of file [GPS\\_L2C.h](#).

**9.13.6.4.573 GPS\_L2\_CNAV\_DATA\_PAGE\_SYMBOLS**

`int32_t GPS_L2_CNAV_DATA_PAGE_SYMBOLS = 600 [constexpr]`  
Definition at line 45 of file [GPS\\_L2C.h](#).

**9.13.6.4.574 GPS\_L2\_FREQ\_HZ**

`double GPS_L2_FREQ_HZ = FREQ2 [constexpr]`  
L2 [Hz].  
Definition at line 35 of file [GPS\\_L2C.h](#).

**9.13.6.4.575 GPS\_L2\_L\_CODE\_LENGTH\_CHIPS**

`int32_t GPS_L2_L_CODE_LENGTH_CHIPS = 767250 [constexpr]`  
GPS L2 L code length [chips].  
Definition at line 41 of file [GPS\\_L2C.h](#).

**9.13.6.4.576 GPS\_L2\_L\_CODE\_RATE\_CPS**

`double GPS_L2_L_CODE_RATE_CPS = 0.5115e6 [constexpr]`  
GPS L2 L code rate [chips/s].  
Definition at line 39 of file [GPS\\_L2C.h](#).

**9.13.6.4.577 GPS\_L2\_L\_PERIOD\_S**

`double GPS_L2_L_PERIOD_S = 1.5 [constexpr]`  
GPS L2 L code period [seconds].  
Definition at line 36 of file [GPS\\_L2C.h](#).

**9.13.6.4.578 GPS\_L2\_M\_CODE\_LENGTH\_CHIPS**

`int32_t GPS_L2_M_CODE_LENGTH_CHIPS = 10230 [constexpr]`  
GPS L2 M code length [chips].  
Definition at line 40 of file [GPS\\_L2C.h](#).

**9.13.6.4.579 GPS\_L2\_M\_CODE\_RATE\_CPS**

`double GPS_L2_M_CODE_RATE_CPS = 0.5115e6 [constexpr]`  
GPS L2 M code rate [chips/s].  
Definition at line 37 of file [GPS\\_L2C.h](#).

**9.13.6.4.580 GPS\_L2\_M\_PERIOD\_S**

`double GPS_L2_M_PERIOD_S = 0.02 [constexpr]`  
GPS L2 M code period [seconds].  
Definition at line 38 of file [GPS\\_L2C.h](#).

**9.13.6.4.581 GPS\_L2\_SAMPLES\_PER\_SYMBOL**

`int32_t GPS_L2_SAMPLES_PER_SYMBOL = 1 [constexpr]`  
Definition at line 44 of file [GPS\\_L2C.h](#).

**9.13.6.4.582 GPS\_L2\_SYMBOLS\_PER\_BIT**

`int32_t GPS_L2_SYMBOLS_PER_BIT = 2 [constexpr]`  
Definition at line 43 of file [GPS\\_L2C.h](#).

**9.13.6.4.583 GPS\_L2C\_HISTORY\_DEEP**

```
int32_t GPS_L2C_HISTORY_DEEP = 5 [constexpr]
```

Definition at line 48 of file [GPS\\_L2C.h](#).

**9.13.6.4.584 GPS\_L2C\_M\_INIT\_REG**

```
int32_t GPS_L2C_M_INIT_REG[115] [constexpr]
```

**Initial value:**

```
=
{0742417664, 0756014035, 0002747144, 0066265724,
 0601403471, 0703232733, 0124510070, 0617316361,
 0047541621, 0733031046, 0713512145, 0024437606,
 0021264003, 0230655351, 0001314400, 0222021506,
 0540264026, 0205521705, 0064022144, 0120161274,
 0044023533, 0724744327, 0045743577, 0741201660,
 0700274134, 0010247261, 0713433445, 0737324162,
 0311627434, 0710452007, 0722462133, 0050172213,
 0500653703, 0755077436, 0136717361, 0756675453,
 0435506112, 0771353753, 0226107701, 0022025110,
 0402466344, 0752566114, 0702011164, 0041216771,
 0047457275, 0266333164, 0713167356, 0060546335,
 0355173035, 0617201036, 0157465571, 0767360553,
 0023127030, 0431343777, 0747317317, 0045706125,
 0002744276, 0060036467, 0217744147, 0603340174,
 0326616775, 0063240065, 0111460621,
 0604055104, 0157065232, 0013305707, 0603552017,
 0230461355, 0603653437, 0652346475, 0743107103,
 0401521277, 0167335110, 0014013575, 0362051132,
 0617753265, 0216363634, 0755561123, 0365304033,
 0625025543, 0054420334, 0415473671, 0662364360,
 0373446602, 0417564100, 0000526452, 0226631300,
 0113752074, 0706134401, 0041352546, 0664630154,
 0276524255, 0714720530, 0714051771, 0044526647,
 0207164322, 0262120161, 0204244652, 0202133131,
 0714351204, 0657127260, 0130567507, 0670517677,
 0607275514, 0045413633, 0212645405, 0613700455,
 0706202440, 0705056276, 0020373522, 0746013617,
 0132720621, 0434015513, 0566721727, 0140633660}
```

Definition at line 53 of file [GPS\\_L2C.h](#).

**9.13.6.4.585 GPS\_L2C\_OPT\_ACQ\_FS\_SPS**

```
uint32_t GPS_L2C_OPT_ACQ_FS_SPS = 2000000 [constexpr]
```

Sampling frequency that maximizes the acquisition SNR while using a non-multiple of chip rate.

Definition at line 51 of file [GPS\\_L2C.h](#).

**9.13.6.4.586 GPS\_L5\_CNAV\_DATA\_PAGE\_BITS**

```
int32_t GPS_L5_CNAV_DATA_PAGE_BITS = 300 [constexpr]
```

GPS L5 CNAV page length, including preamble and CRC [bits].

Definition at line 162 of file [GPS\\_L5.h](#).

**9.13.6.4.587 GPS\_L5\_CNAV\_DATA\_PAGE\_DURATION\_S**

```
int32_t GPS_L5_CNAV_DATA_PAGE_DURATION_S = 6 [constexpr]
```

Definition at line 166 of file [GPS\\_L5.h](#).

**9.13.6.4.588 GPS\_L5\_CNAV\_DATA\_PAGE\_SYMBOLS**

```
int32_t GPS_L5_CNAV_DATA_PAGE_SYMBOLS = 600 [constexpr]
```

Definition at line 165 of file [GPS\\_L5.h](#).

**9.13.6.4.589 GPS\_L5\_FREQ\_HZ**

```
double GPS_L5_FREQ_HZ = FREQ5 [constexpr]
```

L5 [Hz].

Definition at line 32 of file [GPS\\_L5.h](#).

**9.13.6.4.590 GPS\_L5\_HISTORY\_DEEP**

```
int32_t GPS_L5_HISTORY_DEEP = 5 [constexpr]
```

Definition at line 43 of file [GPS\\_L5.h](#).

**9.13.6.4.591 GPS\_L5\_OPT\_ACQ\_FS\_SPS**

```
uint32_t GPS_L5_OPT_ACQ_FS_SPS = 10000000 [constexpr]
```

Sampling frequency that maximizes the acquisition SNR while using a non-multiple of chip rate.

Definition at line 46 of file [GPS\\_L5.h](#).

**9.13.6.4.592 GPS\_L5\_SAMPLES\_PER\_SYMBOL**

```
int32_t GPS_L5_SAMPLES_PER_SYMBOL = 10 [constexpr]
```

Definition at line 164 of file [GPS\\_L5.h](#).

**9.13.6.4.593 GPS\_L5\_SYMBOLS\_PER\_BIT**

```
int32_t GPS_L5_SYMBOLS_PER_BIT = 2 [constexpr]
```

Definition at line 163 of file [GPS\\_L5.h](#).

**9.13.6.4.594 GPS\_L5I\_CODE\_LENGTH\_CHIPS**

```
int32_t GPS_L5I_CODE_LENGTH_CHIPS = 10230 [constexpr]
```

GPS L5I code length [chips].

Definition at line 39 of file [GPS\\_L5.h](#).

**9.13.6.4.595 GPS\_L5I\_CODE\_RATE\_CPS**

```
double GPS_L5I_CODE_RATE_CPS = 10.23e6 [constexpr]
```

GPS L5I code rate [chips/s].

Definition at line 33 of file [GPS\\_L5.h](#).

**9.13.6.4.596 GPS\_L5I\_INIT\_REG**

```
int32_t GPS_L5I_INIT_REG[210] [constexpr]
```

Definition at line 48 of file [GPS\\_L5.h](#).

**9.13.6.4.597 GPS\_L5I\_NH\_CODE**

```
int32_t GPS_L5I_NH_CODE[10] = {0, 0, 0, 0, 1, 1, 0, 1, 0, 1} [constexpr]
```

Definition at line 168 of file [GPS\\_L5.h](#).

**9.13.6.4.598 GPS\_L5I\_NH\_CODE\_LENGTH**

```
int32_t GPS_L5I_NH_CODE_LENGTH = 10 [constexpr]
```

Definition at line 167 of file [GPS\\_L5.h](#).

**9.13.6.4.599 GPS\_L5I\_NH\_CODE\_STR**

```
char GPS_L5I_NH_CODE_STR[11] = "0000110101" [constexpr]
```

Definition at line 171 of file [GPS\\_L5.h](#).

**9.13.6.4.600 GPS\_L5I\_PERIOD\_MS**

```
int32_t GPS_L5I_PERIOD_MS = 1 [constexpr]
```

GPS L5I code period [ms].

Definition at line 40 of file [GPS\\_L5.h](#).

#### 9.13.6.4.601 GPS\_L5I\_PERIOD\_S

```
double GPS_L5I_PERIOD_S = 0.001 [constexpr]
```

GPS L5I code period [seconds].

Definition at line 34 of file [GPS\\_L5.h](#).

#### 9.13.6.4.602 GPS\_L5I\_SYMBOL\_PERIOD\_MS

```
int32_t GPS_L5I_SYMBOL_PERIOD_MS = 10 [constexpr]
```

GPS L5I symbol period [ms].

Definition at line 41 of file [GPS\\_L5.h](#).

#### 9.13.6.4.603 GPS\_L5I\_SYMBOL\_PERIOD\_S

```
double GPS_L5I_SYMBOL_PERIOD_S = 0.01 [constexpr]
```

GPS L5I symbol period [seconds].

Definition at line 35 of file [GPS\\_L5.h](#).

#### 9.13.6.4.604 GPS\_L5Q\_CODE\_LENGTH\_CHIPS

```
int32_t GPS_L5Q_CODE_LENGTH_CHIPS = 10230 [constexpr]
```

GPS L5Q code length [chips].

Definition at line 38 of file [GPS\\_L5.h](#).

#### 9.13.6.4.605 GPS\_L5Q\_CODE\_RATE\_CPS

```
double GPS_L5Q_CODE_RATE_CPS = 10.23e6 [constexpr]
```

GPS L5Q code rate [chips/s].

Definition at line 36 of file [GPS\\_L5.h](#).

#### 9.13.6.4.606 GPS\_L5Q\_INIT\_REG

```
int32_t GPS_L5Q_INIT_REG[210] [constexpr]
```

Definition at line 105 of file [GPS\\_L5.h](#).

#### 9.13.6.4.607 GPS\_L5Q\_NH\_CODE

```
int32_t GPS_L5Q_NH_CODE[20] = {0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0}  
[constexpr]
```

Definition at line 170 of file [GPS\\_L5.h](#).

#### 9.13.6.4.608 GPS\_L5Q\_NH\_CODE\_LENGTH

```
int32_t GPS_L5Q_NH_CODE_LENGTH = 20 [constexpr]
```

Definition at line 169 of file [GPS\\_L5.h](#).

#### 9.13.6.4.609 GPS\_L5Q\_NH\_CODE\_STR

```
char GPS_L5Q_NH_CODE_STR[21] = "00000100110101001110" [constexpr]
```

Definition at line 172 of file [GPS\\_L5.h](#).

#### 9.13.6.4.610 GPS\_L5Q\_PERIOD\_S

```
double GPS_L5Q_PERIOD_S = 0.001 [constexpr]
```

GPS L5Q code period [seconds].

Definition at line 37 of file [GPS\\_L5.h](#).

**9.13.6.4.611 GPS\_SUBFRAME\_BITS**

```
int32_t GPS_SUBFRAME_BITS = 300 [constexpr]
```

Number of bits per subframe in the NAV message [bits].  
Definition at line 68 of file [GPS\\_L1\\_CA.h](#).

**9.13.6.4.612 GPS\_SUBFRAME\_LENGTH**

```
int32_t GPS_SUBFRAME_LENGTH = 40 [constexpr]
```

GPS\_WORD\_LENGTH x 10 = 40 bytes.  
Definition at line 67 of file [GPS\\_L1\\_CA.h](#).

**9.13.6.4.613 GPS\_SUBFRAME\_MS**

```
int32_t GPS_SUBFRAME_MS = 6000 [constexpr]
```

Subframe duration [seconds].  
Definition at line 70 of file [GPS\\_L1\\_CA.h](#).

**9.13.6.4.614 GPS\_SUBFRAME\_SECONDS**

```
int32_t GPS_SUBFRAME_SECONDS = 6 [constexpr]
```

Subframe duration [seconds].  
Definition at line 69 of file [GPS\\_L1\\_CA.h](#).

**9.13.6.4.615 GPS\_WORD\_BITS**

```
int32_t GPS_WORD_BITS = 30 [constexpr]
```

Number of bits per word in the NAV message [bits].  
Definition at line 71 of file [GPS\\_L1\\_CA.h](#).

**9.13.6.4.616 GPS\_WORD\_LENGTH**

```
int32_t GPS_WORD_LENGTH = 4 [constexpr]
```

CRC + GPS WORD (-2 -1 0 ... 29) Bits = 4 bytes.  
Definition at line 66 of file [GPS\\_L1\\_CA.h](#).

**9.13.6.4.617 HALF\_PI**

```
double HALF_PI = GNSS_PI / 2.0 [constexpr]
```

$\pi/2$   
Definition at line 48 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.618 HAS\_MSG\_CODE\_BIAS\_LENGTH**

```
size_t HAS_MSG_CODE_BIAS_LENGTH = 11 [constexpr]
```

Definition at line 53 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.619 HAS\_MSG\_CODE\_BIAS\_SCALE\_FACTOR**

```
float HAS_MSG_CODE_BIAS_SCALE_FACTOR = 0.02 [constexpr]
```

Definition at line 90 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.620 HAS\_MSG\_DELTA\_CLOCK\_CORRECTION\_LENGTH**

```
size_t HAS_MSG_DELTA_CLOCK_CORRECTION_LENGTH = 13 [constexpr]
```

Definition at line 48 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.621 HAS\_MSG\_DELTA\_CLOCK\_CORRECTION\_SUBSET\_LENGTH**

```
size_t HAS_MSG_DELTA_CLOCK_CORRECTION_SUBSET_LENGTH = 13 [constexpr]
```

Definition at line 52 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.622 HAS\_MSG\_DELTA\_CLOCK\_MULTIPLIER\_LENGTH**

size\_t HAS\_MSG\_DELTA\_CLOCK\_MULTIPLIER\_LENGTH = 2 [constexpr]  
Definition at line 47 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.623 HAS\_MSG\_DELTA\_CLOCK\_MULTIPLIER\_SUBSET\_LENGTH**

size\_t HAS\_MSG\_DELTA\_CLOCK\_MULTIPLIER\_SUBSET\_LENGTH = 2 [constexpr]  
Definition at line 51 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.624 HAS\_MSG\_DELTA\_CLOCK\_SCALE\_FACTOR**

float HAS\_MSG\_DELTA\_CLOCK\_SCALE\_FACTOR = 0.0025 [constexpr]  
Definition at line 89 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.625 HAS\_MSG\_DELTA\_CROSS\_TRACK\_LENGTH**

size\_t HAS\_MSG\_DELTA\_CROSS\_TRACK\_LENGTH = 12 [constexpr]  
Definition at line 46 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.626 HAS\_MSG\_DELTA\_CROSS\_TRACK\_SCALE\_FACTOR**

float HAS\_MSG\_DELTA\_CROSS\_TRACK\_SCALE\_FACTOR = 0.008 [constexpr]  
Definition at line 88 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.627 HAS\_MSG\_DELTA\_IN\_TRACK\_LENGTH**

size\_t HAS\_MSG\_DELTA\_IN\_TRACK\_LENGTH = 12 [constexpr]  
Definition at line 45 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.628 HAS\_MSG\_DELTA\_IN\_TRACK\_SCALE\_FACTOR**

float HAS\_MSG\_DELTA\_IN\_TRACK\_SCALE\_FACTOR = 0.008 [constexpr]  
Definition at line 87 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.629 HAS\_MSG\_DELTA\_RADIAL\_LENGTH**

size\_t HAS\_MSG\_DELTA\_RADIAL\_LENGTH = 13 [constexpr]  
Definition at line 44 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.630 HAS\_MSG\_DELTA\_RADIAL\_SCALE\_FACTOR**

float HAS\_MSG\_DELTA\_RADIAL\_SCALE\_FACTOR = 0.0025 [constexpr]  
Definition at line 86 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.631 HAS\_MSG\_GALILEO\_SYSTEM**

uint8\_t HAS\_MSG\_GALILEO\_SYSTEM = 2 [constexpr]  
Definition at line 96 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.632 HAS\_MSG\_GPS\_SYSTEM**

uint8\_t HAS\_MSG\_GPS\_SYSTEM = 0 [constexpr]  
Definition at line 95 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.633 HAS\_MSG\_ID\_CLOCK\_SUBSET\_LENGTH**

size\_t HAS\_MSG\_ID\_CLOCK\_SUBSET\_LENGTH = 4 [constexpr]  
Definition at line 50 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.634 HAS\_MSG\_ID\_MASK\_LENGTH**

size\_t HAS\_MSG\_ID\_MASK\_LENGTH = 4 [constexpr]  
Definition at line 36 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.635 HAS\_MSG\_IOD\_GAL\_LENGTH**

size\_t HAS\_MSG\_IOD\_GAL\_LENGTH = 10 [constexpr]  
Definition at line 43 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.636 HAS\_MSG\_IOD\_GPS\_LENGTH**

size\_t HAS\_MSG\_IOD\_GPS\_LENGTH = 8 [constexpr]  
Definition at line 42 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.637 HAS\_MSG\_MASK\_RESERVED\_LENGTH**

size\_t HAS\_MSG\_MASK\_RESERVED\_LENGTH = 6 [constexpr]  
Definition at line 40 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.638 HAS\_MSG\_NAV\_MESSAGE\_LENGTH**

size\_t HAS\_MSG\_NAV\_MESSAGE\_LENGTH = 3 [constexpr]  
Definition at line 39 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.639 HAS\_MSG\_NSYS\_LENGTH**

size\_t HAS\_MSG\_NSYS\_LENGTH = 4 [constexpr]  
Definition at line 35 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.640 HAS\_MSG\_NSYS\_SUB\_LENGTH**

size\_t HAS\_MSG\_NSYS\_SUB\_LENGTH = 4 [constexpr]  
Definition at line 49 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.641 HAS\_MSG\_NUMBER\_GNSS\_IDS**

int32\_t HAS\_MSG\_NUMBER\_GNSS\_IDS = 16 [constexpr]  
Definition at line 81 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.642 HAS\_MSG\_NUMBER\_MASK\_IDS**

int32\_t HAS\_MSG\_NUMBER\_MASK\_IDS = 32 [constexpr]  
Definition at line 80 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.643 HAS\_MSG\_NUMBER\_MAX\_TOH**

uint16\_t HAS\_MSG\_NUMBER\_MAX\_TOH = 3599 [constexpr]  
Definition at line 93 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.644 HAS\_MSG\_NUMBER\_MESSAGE\_IDS**

int32\_t HAS\_MSG\_NUMBER\_MESSAGE\_IDS = 32 [constexpr]  
Definition at line 82 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.645 HAS\_MSG\_NUMBER\_SATELLITE\_IDS**

int32\_t HAS\_MSG\_NUMBER\_SATELLITE\_IDS = 40 [constexpr]  
Definition at line 83 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.646 HAS\_MSG\_NUMBER\_SIGNAL\_MASKS**

`int32_t HAS_MSG_NUMBER_SIGNAL_MASKS = 16 [constexpr]`  
 Definition at line 84 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.647 HAS\_MSG\_PHASE\_BIAS\_LENGTH**

`size_t HAS_MSG_PHASE_BIAS_LENGTH = 11 [constexpr]`  
 Definition at line 54 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.648 HAS\_MSG\_PHASE\_BIAS\_SCALE\_FACTOR**

`float HAS_MSG_PHASE_BIAS_SCALE_FACTOR = 0.01 [constexpr]`  
 Definition at line 91 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.649 HAS\_MSG\_PHASE\_DISCONTINUITY\_INDICATOR\_LENGTH**

`size_t HAS_MSG_PHASE_DISCONTINUITY_INDICATOR_LENGTH = 2 [constexpr]`  
 Definition at line 55 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.650 HAS\_MSG\_SATELLITE\_MASK\_LENGTH**

`size_t HAS_MSG_SATELLITE_MASK_LENGTH = 40 [constexpr]`  
 Definition at line 37 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.651 HAS\_MSG\_SIGNAL\_MASK\_LENGTH**

`size_t HAS_MSG_SIGNAL_MASK_LENGTH = 16 [constexpr]`  
 Definition at line 38 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.652 HAS\_MSG\_VALIDITY\_INDEX\_LENGTH**

`size_t HAS_MSG_VALIDITY_INDEX_LENGTH = 4 [constexpr]`  
 Definition at line 41 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.653 HAS\_MSG\_WRONG\_SYSTEM**

`uint8_t HAS_MSG_WRONG_SYSTEM = 255 [constexpr]`  
 Definition at line 97 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.654 HAS\_SIGNAL\_INDEX\_TABLE**

`const std::unordered_map<uint8_t, std::unordered_map<uint8_t, std::string> > HAS_SIGNAL_INDEX_TABLE`  
 Definition at line 122 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.655 HAS\_VALIDITY\_INTERVALS**

`const std::unordered_map<uint8_t, uint16_t> HAS_VALIDITY_INTERVALS`

**Initial value:**

```
= {
    {0, 5},
    {1, 10},
    {2, 15},
    {3, 20},
    {4, 30},
    {5, 60},
    {6, 90},
    {7, 120},
    {8, 180},
    {9, 240},
    {10, 300},
    {11, 600},
    {12, 900},
    {13, 1800},
```



```
{14, 3600}}
```

Definition at line 161 of file [Galileo\\_CNAV.h](#).

#### 9.13.6.4.656 I\_0\_2\_LSB

```
double I_0_2_LSB = PI_TWO_N31 [constexpr]
```

Definition at line 72 of file [Galileo\\_INAV.h](#).

#### 9.13.6.4.657 I\_0\_LSB

```
double I_0_LSB = PI_TWO_N31 [constexpr]
```

Definition at line 134 of file [GPS\\_L1\\_CA.h](#).

#### 9.13.6.4.658 I\_DOT\_2\_LSB

```
double I_DOT_2_LSB = PI_TWO_N43 [constexpr]
```

Definition at line 76 of file [Galileo\\_INAV.h](#).

#### 9.13.6.4.659 I\_DOT\_LSB

```
double I_DOT_LSB = PI_TWO_N43 [constexpr]
```

Definition at line 143 of file [GPS\\_L1\\_CA.h](#).

#### 9.13.6.4.660 INAV\_RS\_BUFFER\_LENGTH

```
size_t INAV_RS_BUFFER_LENGTH = 118 [constexpr]
```

Definition at line 272 of file [Galileo\\_INAV.h](#).

#### 9.13.6.4.661 INAV\_RS\_INFO\_VECTOR\_LENGTH

```
size_t INAV_RS_INFO_VECTOR_LENGTH = 58 [constexpr]
```

Definition at line 271 of file [Galileo\\_INAV.h](#).

#### 9.13.6.4.662 INAV\_RS\_PARITY\_VECTOR\_LENGTH

```
size_t INAV_RS_PARITY_VECTOR_LENGTH = 60 [constexpr]
```

Definition at line 270 of file [Galileo\\_INAV.h](#).

#### 9.13.6.4.663 INAV\_RS\_SUBVECTOR\_LENGTH

```
size_t INAV_RS_SUBVECTOR_LENGTH = 15 [constexpr]
```

Definition at line 269 of file [Galileo\\_INAV.h](#).

#### 9.13.6.4.664 M0\_10\_LSB

```
double M0_10_LSB = TWO_N15 [constexpr]
```

Definition at line 234 of file [Galileo\\_INAV.h](#).

#### 9.13.6.4.665 M0\_1\_LSB

```
double M0_1_LSB = PI_TWO_N31 [constexpr]
```

Definition at line 59 of file [Galileo\\_INAV.h](#).

#### 9.13.6.4.666 M0\_7\_LSB

```
double M0_7_LSB = TWO_N15 [constexpr]
```

Definition at line 177 of file [Galileo\\_INAV.h](#).

**9.13.6.4.667 M0\_9\_LSB**

```
double M0_9_LSB = TWO_N15 [constexpr]
```

Definition at line 209 of file [Galileo\\_INAV.h](#).

**9.13.6.4.668 M\_0\_LSB**

```
double M_0_LSB = PI_TWO_N31 [constexpr]
```

Definition at line 111 of file [GPS\\_L1\\_CA.h](#).

**9.13.6.4.669 MAX\_SECONDS\_REMEMBERING\_MID**

```
uint64_t MAX_SECONDS_REMEMBERING_MID = 150 [constexpr]
```

Definition at line 57 of file [Galileo\\_CNAV.h](#).

**9.13.6.4.670 MAX\_TOA\_DELAY\_MS**

```
double MAX_TOA_DELAY_MS = 20.0 [constexpr]
```

Maximum Time-Of-Arrival (TOA) difference between satellites for a receiver operated on Earth surface is 20 ms. According to the GPS orbit model described in [1] Pag. 32. It should be taken into account to set the buffer size for the PRN start timestamp in the pseudoranges block. [1] J. Bao-Yen Tsui, Fundamentals of Global Positioning System Receivers. A Software Approach, John Wiley & Sons, Inc., Hoboken, NJ, 2nd edition, 2005.

Definition at line 50 of file [GPS\\_L1\\_CA.h](#).

**9.13.6.4.671 MAXCODE**

```
int32_t MAXCODE = 55 [constexpr]
```

max number of obs code

Definition at line 86 of file [gnss\\_obs\\_codes.h](#).

**9.13.6.4.672 OMEGA0\_10\_LSB**

```
double OMEGA0_10_LSB = TWO_N15 [constexpr]
```

Definition at line 230 of file [Galileo\\_INAV.h](#).

**9.13.6.4.673 OMEGA0\_7\_LSB**

```
double OMEGA0_7_LSB = TWO_N15 [constexpr]
```

Definition at line 173 of file [Galileo\\_INAV.h](#).

**9.13.6.4.674 OMEGA0\_8\_LSB**

```
double OMEGA0_8_LSB = TWO_N15 [constexpr]
```

Definition at line 198 of file [Galileo\\_INAV.h](#).

**9.13.6.4.675 OMEGA\_0\_2\_LSB**

```
double OMEGA_0_2_LSB = PI_TWO_N31 [constexpr]
```

Definition at line 70 of file [Galileo\\_INAV.h](#).

**9.13.6.4.676 OMEGA\_0\_LSB**

```
double OMEGA_0_LSB = PI_TWO_N31 [constexpr]
```

Definition at line 130 of file [GPS\\_L1\\_CA.h](#).

**9.13.6.4.677 OMEGA\_2\_LSB**

```
double OMEGA_2_LSB = PI_TWO_N31 [constexpr]
```

Definition at line 74 of file [Galileo\\_INAV.h](#).

**9.13.6.4.678 OMEGA\_7\_LSB**

double OMEGA\_7\_LSB = TWO\_N15 [constexpr]  
 Definition at line 169 of file [Galileo\\_INAV.h](#).

**9.13.6.4.679 OMEGA\_8\_LSB**

double OMEGA\_8\_LSB = TWO\_N15 [constexpr]  
 Definition at line 194 of file [Galileo\\_INAV.h](#).

**9.13.6.4.680 OMEGA\_9\_LSB**

double OMEGA\_9\_LSB = TWO\_N15 [constexpr]  
 Definition at line 222 of file [Galileo\\_INAV.h](#).

**9.13.6.4.681 OMEGA\_DOT\_10\_LSB**

double OMEGA\_DOT\_10\_LSB = TWO\_N33 [constexpr]  
 Definition at line 232 of file [Galileo\\_INAV.h](#).

**9.13.6.4.682 OMEGA\_DOT\_3\_LSB**

double OMEGA\_DOT\_3\_LSB = PI\_TWO\_N43 [constexpr]  
 Definition at line 82 of file [Galileo\\_INAV.h](#).

**9.13.6.4.683 OMEGA\_DOT\_7\_LSB**

double OMEGA\_DOT\_7\_LSB = TWO\_N33 [constexpr]  
 Definition at line 175 of file [Galileo\\_INAV.h](#).

**9.13.6.4.684 OMEGA\_DOT\_8\_LSB**

double OMEGA\_DOT\_8\_LSB = TWO\_N33 [constexpr]  
 Definition at line 200 of file [Galileo\\_INAV.h](#).

**9.13.6.4.685 OMEGA\_DOT\_LSB**

double OMEGA\_DOT\_LSB = PI\_TWO\_N43 [constexpr]  
 Definition at line 140 of file [GPS\\_L1\\_CA.h](#).

**9.13.6.4.686 OMEGA\_LSB**

double OMEGA\_LSB = PI\_TWO\_N31 [constexpr]  
 Definition at line 138 of file [GPS\\_L1\\_CA.h](#).

**9.13.6.4.687 OSNMA\_TABLE\_1**

const std::unordered\_map<uint8\_t, std::string> OSNMA\_TABLE\_1

**Initial value:**

```
= {
    {0, std::string("Reserved")},
    {1, std::string("Test")},
    {2, std::string("Operational")},
    {3, std::string("Don't use")}}
```

Definition at line 36 of file [Galileo\\_OSNMA.h](#).

**9.13.6.4.688 OSNMA\_TABLE\_10**

const std::unordered\_map<uint8\_t, uint16\_t> OSNMA\_TABLE\_10

**Initial value:**

```
= {
    {0, 96},
```

```

{1, 104},
{2, 112},
{3, 120},
{4, 128},
{5, 160},
{6, 192},
{7, 224},
{8, 256},
{9, 0},
{10, 0},
{11, 0},
{12, 0},
{13, 0},
{15, 0},
{15, 0}

```

Definition at line 120 of file [Galileo\\_OSNMA.h](#).

#### 9.13.6.4.689 OSNMA\_TABLE\_11

```
const std::unordered_map<uint8_t, uint8_t> OSNMA_TABLE_11
```

**Initial value:**

```

= {
    {0, 0},
    {1, 0},
    {2, 0},
    {3, 0},
    {4, 0},
    {5, 20},
    {6, 24},
    {7, 28},
    {8, 32},
    {9, 40},
    {10, 0},
    {11, 0},
    {12, 0},
    {13, 0},
    {14, 0},
    {15, 0},
}

```

Definition at line 138 of file [Galileo\\_OSNMA.h](#).

#### 9.13.6.4.690 OSNMA\_TABLE\_15

```
const std::unordered_map<std::string, uint16_t> OSNMA_TABLE_15
```

**Initial value:**

```

= {
    {std::string("ECDSA P-256"), 512},
    {std::string("ECDSA P-521"), 1056}
}

```

Definition at line 157 of file [Galileo\\_OSNMA.h](#).

#### 9.13.6.4.691 OSNMA\_TABLE\_16

```
const std::unordered_map<uint8_t, Mack\_lookup> OSNMA_TABLE_16
```

**Initial value:**

```

= {
    {27, {2, 6, {"00S", "00E", "00E", "00E", "12S", "00E"}, {"00S ", "00E", "00E", "04S", "12S", "00E"}}},
    {28, {2, 10, {"00S", "00E", "00E", "00E", "00S", "00E", "00E", "00E", "12S", "00E", "00E"}, {"00S", "00E", "00E", "00S", "00E", "04S", "12S", "00E", "00E"}}},
    {31, {2, 5, {"00S", "00E", "00E", "12S", "00E"}, {"00S", "00E", "00E", "12S", "04S"}}},
    {33, {2, 6, {"00S", "00E", "04S", "00E", "12S", "00E"}, {"00S", "00E", "00E", "12S", "00E", "12E"}}},
    {34, {2, 6, {"00S", "FLX", "04S", "FLX", "12S", "00E"}, {"00S", "FLX", "00E", "12S", "00E", "12E"}}},
    {35, {2, 6, {"00S", "FLX", "04S", "FLX", "12S", "FLX"}, {"00S", "FLX", "FLX", "12S", "FLX", "FLX"}}},
    {36, {2, 5, {"00S", "FLX", "04S", "FLX", "12S"}, {"00S", "FLX", "00E", "12S", "12E"}}},
    {37, {2, 5, {"00S", "00E", "04S", "00E", "12S"}, {"00S", "00E", "00E", "12S", "12E"}}},
    {38, {2, 5, {"00S", "FLX", "04S", "FLX", "12S"}, {"00S", "FLX", "FLX", "12S", "FLX"}}},
    {39, {2, 4, {"00S", "FLX", "04S", "FLX"}, {"00S", "FLX", "00E", "12S"}}},
    {40, {2, 4, {"00S", "00E", "04S", "12S"}, {"00S", "00E", "00E", "12E"}}},
    {41, {2, 4, {"00S", "FLX", "04S", "FLX"}, {"00S", "FLX", "FLX", "12S"}}}
}

```

Definition at line 183 of file [Galileo\\_OSNMA.h](#).

#### 9.13.6.4.692 OSNMA\_TABLE\_2

```
const std::unordered_map<uint8_t, std::string> OSNMA_TABLE_2
```

**Initial value:**

```
= {
    {0, std::string("Reserved")},
    {1, std::string("Nominal")},
    {2, std::string("End of Chain (EOC)")},
    {3, std::string("Chain Revoked (CREV)")},
    {4, std::string("New Public Key (NPK)")},
    {5, std::string("Public Key Revoked (PKREV)")},
    {6, std::string("New Merkle Tree (NMT)")},
    {7, std::string("Alert Message (AM)")}}
```

Definition at line 44 of file [Galileo\\_OSNMA.h](#).

#### 9.13.6.4.693 OSNMA\_TABLE\_3

```
const std::unordered_map<uint8_t, std::pair<uint16_t, uint16_t> > OSNMA_TABLE_3
```

**Initial value:**

```
= {
    {0, {0, 0}},
    {1, {0, 0}},
    {2, {0, 0}},
    {3, {0, 0}},
    {4, {0, 0}},
    {5, {0, 0}},
    {6, {0, 0}},
    {7, {13, 1352}},
    {8, {14, 1456}},
    {9, {15, 1560}},
    {10, {16, 1664}},
    {11, {0, 0}},
    {12, {0, 0}},
    {13, {0, 0}},
    {14, {0, 0}},
    {15, {0, 0}}}
```

Definition at line 55 of file [Galileo\\_OSNMA.h](#).

#### 9.13.6.4.694 OSNMA\_TABLE\_5

```
const std::unordered_map<uint8_t, std::string> OSNMA_TABLE_5
```

**Initial value:**

```
= {
    {0, std::string("Reserved")},
    {1, std::string("ECDSA P-256")},
    {2, std::string("Reserved")},
    {3, std::string("ECDSA P-521")},
    {4, std::string("OAM")},
    {5, std::string("Reserved")},
    {6, std::string("Reserved")},
    {7, std::string("Reserved")},
    {8, std::string("Reserved")},
    {9, std::string("Reserved")},
    {10, std::string("Reserved")},
    {11, std::string("Reserved")},
    {12, std::string("Reserved")},
    {13, std::string("Reserved")},
    {14, std::string("Reserved")},
    {15, std::string("Reserved")}}
```

Definition at line 73 of file [Galileo\\_OSNMA.h](#).

#### 9.13.6.4.695 OSNMA\_TABLE\_6

```
const std::unordered_map<std::string, uint16_t> OSNMA_TABLE_6
```

**Initial value:**

```
= {
    {std::string("ECDSA P-256"), 264},
    {std::string("ECDSA P-521"), 536}}
```

Definition at line 91 of file [Galileo\\_OSNMA.h](#).

#### 9.13.6.4.696 OSNMA\_TABLE\_7

```
const std::unordered_map<uint8_t, std::pair<uint16_t, uint16_t> > OSNMA_TABLE_7
```

**Initial value:**

```
= {
    {0, {0, 0}},
    {1, {7, 728}},
    {2, {8, 832}},
```

```

{3, {9, 936}},
{4, {10, 1040}},
{5, {11, 1144}},
{6, {12, 1248}},
{7, {13, 1352}},
{8, {14, 1456}},
{9, {0, 0}},
{10, {0, 0}},
{11, {0, 0}},
{12, {0, 0}},
{13, {0, 0}},
{14, {0, 0}},
{15, {0, 0}}

```

Definition at line 96 of file [Galileo\\_OSNMA.h](#).

#### 9.13.6.4.697 OSNMA\_TABLE\_8

```
const std::unordered_map<uint8_t, std::string> OSNMA_TABLE_8
```

**Initial value:**

```

= {
    {0, std::string("SHA-256")},
    {1, std::string("Reserved")},
    {2, std::string("SHA3-256")},
    {3, std::string("Reserved")}
}

```

Definition at line 114 of file [Galileo\\_OSNMA.h](#).

#### 9.13.6.4.698 PI\_TWO\_N19

```
double PI_TWO_N19 = 5.992112452678286e-006 [constexpr]
Pi*2-19.
```

Definition at line 117 of file [MATH\\_CONSTANTS.h](#).

#### 9.13.6.4.699 PI\_TWO\_N23

```
double PI_TWO_N23 = 3.745070282923929e-007 [constexpr]
Pi*2-23.
```

Definition at line 121 of file [MATH\\_CONSTANTS.h](#).

#### 9.13.6.4.700 PI\_TWO\_N31

```
double PI_TWO_N31 = 1.462918079267160e-009 [constexpr]
Pi*2-31.
```

Definition at line 119 of file [MATH\\_CONSTANTS.h](#).

#### 9.13.6.4.701 PI\_TWO\_N38

```
double PI_TWO_N38 = 1.142904749427469e-011 [constexpr]
Pi*2-38.
```

Definition at line 120 of file [MATH\\_CONSTANTS.h](#).

#### 9.13.6.4.702 PI\_TWO\_N43

```
double PI_TWO_N43 = 3.571577341960839e-013 [constexpr]
Pi*2-43.
```

Definition at line 118 of file [MATH\\_CONSTANTS.h](#).

#### 9.13.6.4.703 R2D

```
double R2D = 180.0 / GNSS_PI [constexpr]
rad to deg
```

Definition at line 124 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.704 SC2RAD**

double SC2RAD = GNSS\_PI [constexpr]  
 semi-circle to radian (IS-GPS)  
 Definition at line 125 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.705 SIGNAL\_FREQ\_MAP**

const std::unordered\_map<std::string, double> SIGNAL\_FREQ\_MAP  
 Initial value:

```
= {
    {"1C", FREQ1},
    {"2S", FREQ2},
    {"L5", FREQ5},
    {"1B", FREQ1},
    {"5X", FREQ5},
    {"E6", FREQ6},
    {"7X", FREQ7},
    {"1G", FREQ1_GLO},
    {"2G", FREQ2_GLO},
    {"B1", FREQ1_BDS},
    {"B2", FREQ2_BDS},
    {"B3", FREQ3_BDS},
}
```

Definition at line 47 of file [gnss\\_frequencies.h](#).

**9.13.6.4.706 SIZE\_DSM\_BLOCKS\_BYTES**

size\_t SIZE\_DSM\_BLOCKS\_BYTES = 13 [constexpr]  
 Definition at line 33 of file [Galileo\\_OSNMA.h](#).

**9.13.6.4.707 SPEED\_OF\_LIGHT\_M\_MS**

double SPEED\_OF\_LIGHT\_M\_MS = 299792.4580 [constexpr]  
 Speed of light in vacuum [m/ms].  
 Definition at line 28 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.708 SPEED\_OF\_LIGHT\_M\_S**

double SPEED\_OF\_LIGHT\_M\_S = 299792458.0 [constexpr]  
 Speed of light in vacuum [m/s].  
 Definition at line 27 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.709 SQRT\_A\_LSB**

double SQRT\_A\_LSB = TWO\_N19 [constexpr]  
 Definition at line 119 of file [GPS\\_L1\\_CA.h](#).

**9.13.6.4.710 T0A\_7\_LSB**

int32\_t T0A\_7\_LSB = 600 [constexpr]  
 Definition at line 162 of file [Galileo\\_INAV.h](#).

**9.13.6.4.711 T0A\_9\_LSB**

int32\_t T0A\_9\_LSB = 600 [constexpr]  
 Definition at line 207 of file [Galileo\\_INAV.h](#).

**9.13.6.4.712 T0C\_4\_LSB**

int32\_t T0C\_4\_LSB = 60 [constexpr]  
 Definition at line 104 of file [Galileo\\_INAV.h](#).

**9.13.6.4.713 T0E\_1\_LSB**

```
int32_t T0E_1_LSB = 60 [constexpr]
```

Definition at line 57 of file [Galileo\\_INAV.h](#).

**9.13.6.4.714 T0T\_6\_LSB**

```
int32_t T0T_6_LSB = 3600 [constexpr]
```

Definition at line 150 of file [Galileo\\_INAV.h](#).

**9.13.6.4.715 T\_0\_G\_10\_LSB**

```
int32_t T_0_G_10_LSB = 3600 [constexpr]
```

Definition at line 246 of file [Galileo\\_INAV.h](#).

**9.13.6.4.716 T\_GD\_LSB**

```
double T_GD_LSB = TWO_N31 [constexpr]
```

Definition at line 93 of file [GPS\\_L1\\_CA.h](#).

**9.13.6.4.717 T\_OA\_LSB**

```
int32_t T_OA_LSB = TWO_P12 [constexpr]
```

Definition at line 201 of file [GPS\\_L1\\_CA.h](#).

**9.13.6.4.718 T\_OC\_LSB**

```
int32_t T_OC_LSB = static_cast<int32_t>(TWO_P4) [constexpr]
```

Definition at line 96 of file [GPS\\_L1\\_CA.h](#).

**9.13.6.4.719 T\_OE\_LSB**

```
int32_t T_OE_LSB = static_cast<int32_t>(TWO_P4) [constexpr]
```

Definition at line 121 of file [GPS\\_L1\\_CA.h](#).

**9.13.6.4.720 T\_OT\_LSB**

```
double T_OT_LSB = TWO_P12 [constexpr]
```

Definition at line 173 of file [GPS\\_L1\\_CA.h](#).

**9.13.6.4.721 TEXT\_BLACK**

```
const std::string TEXT_BLACK = "\033[30m"
```

Definition at line 36 of file [display.h](#).

**9.13.6.4.722 TEXT\_BLUE**

```
const std::string TEXT_BLUE = "\033[34m"
```

Definition at line 40 of file [display.h](#).

**9.13.6.4.723 TEXT\_BOLD\_BLACK**

```
const std::string TEXT_BOLD_BLACK = "\033[1m\033[30m"
```

Definition at line 44 of file [display.h](#).

**9.13.6.4.724 TEXT\_BOLD\_BLUE**

```
const std::string TEXT_BOLD_BLUE = "\033[1m\033[34m"
```

Definition at line 48 of file [display.h](#).



**9.13.6.4.725 TEXT\_BOLD\_CYAN**

```
const std::string TEXT_BOLD_CYAN = "\033[1m\033[36m"
```

Definition at line 50 of file [display.h](#).

**9.13.6.4.726 TEXT\_BOLD\_GREEN**

```
const std::string TEXT_BOLD_GREEN = "\033[1m\033[32m"
```

Definition at line 46 of file [display.h](#).

**9.13.6.4.727 TEXT\_BOLD\_MAGENTA**

```
const std::string TEXT_BOLD_MAGENTA = "\033[1m\033[35m"
```

Definition at line 49 of file [display.h](#).

**9.13.6.4.728 TEXT\_BOLD\_RED**

```
const std::string TEXT_BOLD_RED = "\033[1m\033[31m"
```

Definition at line 45 of file [display.h](#).

**9.13.6.4.729 TEXT\_BOLD\_WHITE**

```
const std::string TEXT_BOLD_WHITE = "\033[1m\033[37m"
```

Definition at line 51 of file [display.h](#).

**9.13.6.4.730 TEXT\_BOLD\_YELLOW**

```
const std::string TEXT_BOLD_YELLOW = "\033[1m\033[33m"
```

Definition at line 47 of file [display.h](#).

**9.13.6.4.731 TEXT\_CYAN**

```
const std::string TEXT_CYAN = "\033[36m"
```

Definition at line 42 of file [display.h](#).

**9.13.6.4.732 TEXT\_GREEN**

```
const std::string TEXT_GREEN = "\033[32m"
```

Definition at line 38 of file [display.h](#).

**9.13.6.4.733 TEXT\_MAGENTA**

```
const std::string TEXT_MAGENTA = "\033[35m"
```

Definition at line 41 of file [display.h](#).

**9.13.6.4.734 TEXT\_RED**

```
const std::string TEXT_RED = "\033[31m"
```

Definition at line 37 of file [display.h](#).

**9.13.6.4.735 TEXT\_RESET**

```
const std::string TEXT_RESET = "\033[0m"
```

Definition at line 35 of file [display.h](#).

**9.13.6.4.736 TEXT\_WHITE**

```
const std::string TEXT_WHITE = "\033[37m"
```

Definition at line 43 of file [display.h](#).

**9.13.6.4.737 TEXT\_YELLOW**

```
const std::string TEXT_YELLOW = "\033[33m"
```

Definition at line 39 of file [display.h](#).

**9.13.6.4.738 TWO\_N10**

```
double TWO_N10 = 0.0009765625 [constexpr]  
2^-10
```

Definition at line 77 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.739 TWO\_N11**

```
double TWO_N11 = 4.882812500000000e-004 [constexpr]  
2^-11
```

Definition at line 78 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.740 TWO\_N14**

```
double TWO_N14 = 0.00006103515625 [constexpr]  
2^-14
```

Definition at line 79 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.741 TWO\_N15**

```
double TWO_N15 = 3.051757812500000e-005 [constexpr]  
2^-15
```

Definition at line 80 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.742 TWO\_N16**

```
double TWO_N16 = 1.525878906250000e-005 [constexpr]  
2^-16
```

Definition at line 81 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.743 TWO\_N17**

```
double TWO_N17 = 7.629394531250000e-006 [constexpr]  
2^-17
```

Definition at line 82 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.744 TWO\_N18**

```
double TWO_N18 = 3.814697265625000e-006 [constexpr]  
2^-18
```

Definition at line 83 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.745 TWO\_N19**

```
double TWO_N19 = 1.907348632812500e-006 [constexpr]  
2^-19
```

Definition at line 84 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.746 TWO\_N2**

```
double TWO_N2 = 0.25 [constexpr]  
2^-2
```

Definition at line 72 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.747 TWO\_N20**

```
double TWO_N20 = 9.536743164062500e-007 [constexpr]  
2^-20
```

Definition at line 85 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.748 TWO\_N21**

```
double TWO_N21 = 4.768371582031250e-007 [constexpr]  
2^-21
```

Definition at line 86 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.749 TWO\_N22**

```
double TWO_N22 = 2.384185791015625e-007 [constexpr]  
2^-22
```

Definition at line 87 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.750 TWO\_N23**

```
double TWO_N23 = 1.192092895507810e-007 [constexpr]  
2^-23
```

Definition at line 88 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.751 TWO\_N24**

```
double TWO_N24 = 5.960464477539063e-008 [constexpr]  
2^-24
```

Definition at line 89 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.752 TWO\_N25**

```
double TWO_N25 = 2.980232238769531e-008 [constexpr]  
2^-25
```

Definition at line 90 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.753 TWO\_N26**

```
double TWO_N26 = 1.490116119384765e-009 [constexpr]  
2^-26
```

Definition at line 91 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.754 TWO\_N27**

```
double TWO_N27 = 7.450580596923828e-009 [constexpr]  
2^-27
```

Definition at line 92 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.755 TWO\_N29**

```
double TWO_N29 = 1.862645149230957e-009 [constexpr]  
2^-29
```

Definition at line 93 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.756 TWO\_N30**

```
double TWO_N30 = 9.313225746154785e-010 [constexpr]  
2^-30
```

Definition at line 94 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.757 TWO\_N31**

```
double TWO_N31 = 4.656612873077393e-010 [constexpr]  
2^-31
```

Definition at line 95 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.758 TWO\_N32**

```
double TWO_N32 = 2.328306436538696e-010 [constexpr]  
2^-32
```

Definition at line 96 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.759 TWO\_N33**

```
double TWO_N33 = 1.164153218269348e-010 [constexpr]  
2^-33
```

Definition at line 97 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.760 TWO\_N34**

```
double TWO_N34 = 5.82076609134674e-011 [constexpr]  
2^-34
```

Definition at line 98 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.761 TWO\_N35**

```
double TWO_N35 = 2.91038304567337e-011 [constexpr]  
2^-35
```

Definition at line 99 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.762 TWO\_N38**

```
double TWO_N38 = 3.637978807091713e-012 [constexpr]  
2^-38
```

Definition at line 100 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.763 TWO\_N39**

```
double TWO_N39 = 1.818989403545856e-012 [constexpr]  
2^-39
```

Definition at line 101 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.764 TWO\_N40**

```
double TWO_N40 = 9.094947017729280e-013 [constexpr]  
2^-40
```

Definition at line 102 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.765 TWO\_N43**

```
double TWO_N43 = 1.136868377216160e-013 [constexpr]  
2^-43
```

Definition at line 103 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.766 TWO\_N44**

```
double TWO_N44 = 5.684341886080802e-14 [constexpr]  
2^-44
```

Definition at line 104 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.767 TWO\_N46**

```
double TWO_N46 = 1.4210854715202e-014 [constexpr]  
2^-46
```

Definition at line 105 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.768 TWO\_N48**

```
double TWO_N48 = 3.552713678800501e-15 [constexpr]  
2^-46
```

Definition at line 106 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.769 TWO\_N5**

```
double TWO_N5 = 0.03125 [constexpr]  
2^-5
```

Definition at line 73 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.770 TWO\_N50**

```
double TWO_N50 = 8.881784197001252e-016 [constexpr]  
2^-50
```

Definition at line 108 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.771 TWO\_N51**

```
double TWO_N51 = 4.44089209850063e-016 [constexpr]  
2^-51
```

Definition at line 109 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.772 TWO\_N55**

```
double TWO_N55 = 2.775557561562891e-017 [constexpr]  
2^-55
```

Definition at line 110 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.773 TWO\_N57**

```
double TWO_N57 = 6.938893903907228e-18 [constexpr]  
2^-57
```

Definition at line 111 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.774 TWO\_N59**

```
double TWO_N59 = 1.73472347597681e-018 [constexpr]  
2^-59
```

Definition at line 112 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.775 TWO\_N6**

```
double TWO_N6 = 0.015625 [constexpr]  
2^-6
```

Definition at line 74 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.776 TWO\_N60**

```
double TWO_N60 = 8.673617379884036e-19 [constexpr]  
2^-60
```

Definition at line 113 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.777 TWO\_N66**

```
double TWO_N66 = 1.3552527156068805425093160010874271392822265625e-20 [constexpr]
2^-66
```

Definition at line 114 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.778 TWO\_N68**

```
double TWO_N68 = 3.388131789017201e-21 [constexpr]
2^-68
```

Definition at line 115 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.779 TWO\_N8**

```
double TWO_N8 = 0.00390625 [constexpr]
2^-8
```

Definition at line 75 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.780 TWO\_N9**

```
double TWO_N9 = 0.001953125 [constexpr]
2^-9
```

Definition at line 76 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.781 TWO\_P11**

```
double TWO_P11 = 2048.0 [constexpr]
2^11
```

Definition at line 62 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.782 TWO\_P12**

```
double TWO_P12 = 4096.0 [constexpr]
2^12
```

Definition at line 63 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.783 TWO\_P14**

```
double TWO_P14 = 16384.0 [constexpr]
2^14
```

Definition at line 64 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.784 TWO\_P16**

```
double TWO_P16 = 65536.0 [constexpr]
2^16
```

Definition at line 65 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.785 TWO\_P19**

```
double TWO_P19 = 524288.0 [constexpr]
2^19
```

Definition at line 66 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.786 TWO\_P3**

```
double TWO_P3 = 8.0 [constexpr]
2^3
```

Definition at line 59 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.787 TWO\_P31**

```
double TWO_P31 = 2147483648.0 [constexpr]  
2^31
```

Definition at line 67 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.788 TWO\_P32**

```
double TWO_P32 = 4294967296.0 [constexpr]  
2^32
```

Definition at line 68 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.789 TWO\_P4**

```
double TWO_P4 = 16.0 [constexpr]  
2^4
```

Definition at line 60 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.790 TWO\_P56**

```
double TWO_P56 = 7.205759403792794e+016 [constexpr]  
2^56
```

Definition at line 69 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.791 TWO\_P57**

```
double TWO_P57 = 1.441151880758559e+017 [constexpr]  
2^57
```

Definition at line 70 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.792 TWO\_P8**

```
double TWO_P8 = 256.0 [constexpr]  
2^8
```

Definition at line 61 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.793 TWO\_PI**

```
double TWO_PI = 2.0 * GNSS_PI [constexpr]  
2 * pi
```

Definition at line 49 of file [MATH\\_CONSTANTS.h](#).

**9.13.6.4.794 WN\_LSF\_LSB**

```
double WN_LSF_LSB = 1 [constexpr]
```

Definition at line 179 of file [GPS\\_L1\\_CA.h](#).

**9.13.6.4.795 WN\_T\_LSB**

```
double WN_T_LSB = 1 [constexpr]
```

Definition at line 175 of file [GPS\\_L1\\_CA.h](#).





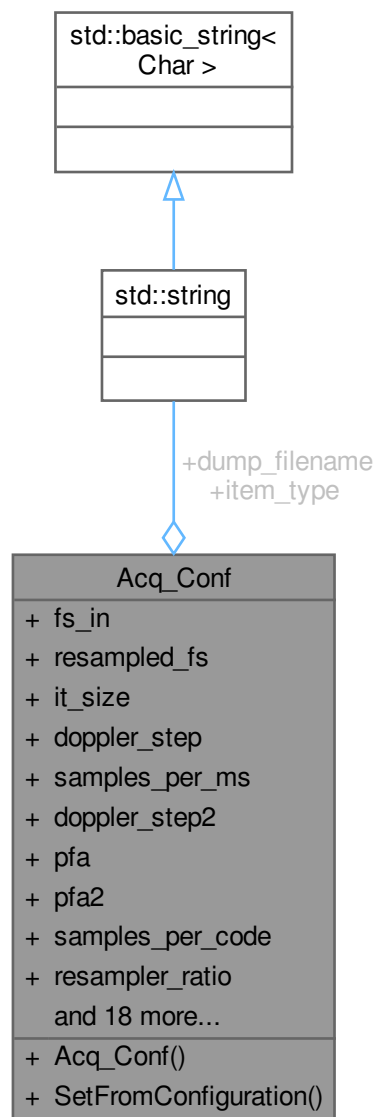


## Chapter 10

# Class Documentation

### 10.1 Acq\_Conf Class Reference

Collaboration diagram for Acq\_Conf:



## Public Member Functions

- void **SetFromConfiguration** (const [ConfigurationInterface](#) \*configuration, const std::string &role, double chip\_rate, double opt\_freq)

## Public Attributes

- std::string [item\\_type](#) {"gr\_complex"}
- std::string [dump\\_filename](#)
- int64\_t [fs\\_in](#) {4000000LL}
- int64\_t [resampled\\_fs](#) {0LL}
- size\_t [it\\_size](#) {sizeof(gr\_complex)}
- float [doppler\\_step](#) {250.0}
- float [samples\\_per\\_ms](#) {0.0}
- float [doppler\\_step2](#) {125.0}
- float [pfa](#) {0.0}
- float [pfa2](#) {0.0}
- float [samples\\_per\\_code](#) {0.0}
- float [resampler\\_ratio](#) {1.0}
- uint32\_t [sampled\\_ms](#) {1U}
- uint32\_t [ms\\_per\\_code](#) {1U}
- uint32\_t [samples\\_per\\_chip](#) {2U}
- uint32\_t [chips\\_per\\_second](#) {1023000U}
- uint32\_t [max\\_dwells](#) {1U}
- uint32\_t [num\\_doppler\\_bins\\_step2](#) {4U}
- uint32\_t [resampler\\_latency\\_samples](#) {0U}
- uint32\_t [dump\\_channel](#) {0U}
- int32\_t [doppler\\_max](#) {5000}
- int32\_t [doppler\\_min](#) {-5000}
- bool [bit\\_transition\\_flag](#) {false}
- bool [use\\_CFAR\\_algorithm\\_flag](#) {true}
- bool [dump](#) {false}
- bool [blocking](#) {true}
- bool [blocking\\_on\\_standby](#) {false}
- bool [make\\_2\\_steps](#) {false}
- bool [use\\_automatic\\_resampler](#) {false}
- bool [enable\\_monitor\\_output](#) {false}

### 10.1.1 Detailed Description

Definition at line 33 of file [acq\\_conf.h](#).

### 10.1.2 Member Data Documentation

#### 10.1.2.1 bit\_transition\_flag

```
bool Acq_Conf::bit_transition_flag {false}
```

Definition at line 68 of file [acq\\_conf.h](#).

#### 10.1.2.2 blocking

```
bool Acq_Conf::blocking {true}
```

Definition at line 71 of file [acq\\_conf.h](#).

#### 10.1.2.3 blocking\_on\_standby

```
bool Acq_Conf::blocking_on_standby {false}
```

Definition at line 72 of file [acq\\_conf.h](#).

#### 10.1.2.4 chips\_per\_second

uint32\_t Acq\_Conf::chips\_per\_second {1023000U}  
Definition at line 60 of file [acq\\_conf.h](#).

#### 10.1.2.5 doppler\_max

int32\_t Acq\_Conf::doppler\_max {5000}  
Definition at line 65 of file [acq\\_conf.h](#).

#### 10.1.2.6 doppler\_min

int32\_t Acq\_Conf::doppler\_min {-5000}  
Definition at line 66 of file [acq\\_conf.h](#).

#### 10.1.2.7 doppler\_step

float Acq\_Conf::doppler\_step {250.0}  
Definition at line 49 of file [acq\\_conf.h](#).

#### 10.1.2.8 doppler\_step2

float Acq\_Conf::doppler\_step2 {125.0}  
Definition at line 51 of file [acq\\_conf.h](#).

#### 10.1.2.9 dump

bool Acq\_Conf::dump {false}  
Definition at line 70 of file [acq\\_conf.h](#).

#### 10.1.2.10 dump\_channel

uint32\_t Acq\_Conf::dump\_channel {0U}  
Definition at line 64 of file [acq\\_conf.h](#).

#### 10.1.2.11 dump\_filename

std::string Acq\_Conf::dump\_filename  
Definition at line 42 of file [acq\\_conf.h](#).

#### 10.1.2.12 enable\_monitor\_output

bool Acq\_Conf::enable\_monitor\_output {false}  
Definition at line 75 of file [acq\\_conf.h](#).

#### 10.1.2.13 fs\_in

int64\_t Acq\_Conf::fs\_in {4000000LL}  
Definition at line 44 of file [acq\\_conf.h](#).

#### 10.1.2.14 it\_size

size\_t Acq\_Conf::it\_size {sizeof(gr\_complex)}  
Definition at line 47 of file [acq\\_conf.h](#).

#### 10.1.2.15 item\_type

std::string Acq\_Conf::item\_type {"gr\_complex"}  
Definition at line 41 of file [acq\\_conf.h](#).

#### 10.1.2.16 make\_2\_steps

`bool Acq_Conf::make_2_steps {false}`  
Definition at line 73 of file [acq\\_conf.h](#).

#### 10.1.2.17 max\_dwells

`uint32_t Acq_Conf::max_dwells {1U}`  
Definition at line 61 of file [acq\\_conf.h](#).

#### 10.1.2.18 ms\_per\_code

`uint32_t Acq_Conf::ms_per_code {1U}`  
Definition at line 58 of file [acq\\_conf.h](#).

#### 10.1.2.19 num\_doppler\_bins\_step2

`uint32_t Acq_Conf::num_doppler_bins_step2 {4U}`  
Definition at line 62 of file [acq\\_conf.h](#).

#### 10.1.2.20 pfa

`float Acq_Conf::pfa {0.0}`  
Definition at line 52 of file [acq\\_conf.h](#).

#### 10.1.2.21 pfa2

`float Acq_Conf::pfa2 {0.0}`  
Definition at line 53 of file [acq\\_conf.h](#).

#### 10.1.2.22 resampled\_fs

`int64_t Acq_Conf::resampled_fs {0LL}`  
Definition at line 45 of file [acq\\_conf.h](#).

#### 10.1.2.23 resampler\_latency\_samples

`uint32_t Acq_Conf::resampler_latency_samples {0U}`  
Definition at line 63 of file [acq\\_conf.h](#).

#### 10.1.2.24 resampler\_ratio

`float Acq_Conf::resampler_ratio {1.0}`  
Definition at line 55 of file [acq\\_conf.h](#).

#### 10.1.2.25 sampled\_ms

`uint32_t Acq_Conf::sampled_ms {1U}`  
Definition at line 57 of file [acq\\_conf.h](#).

#### 10.1.2.26 samples\_per\_chip

`uint32_t Acq_Conf::samples_per_chip {2U}`  
Definition at line 59 of file [acq\\_conf.h](#).

#### 10.1.2.27 samples\_per\_code

`float Acq_Conf::samples_per_code {0.0}`  
Definition at line 54 of file [acq\\_conf.h](#).

#### 10.1.2.28 samples\_per\_ms

```
float Acq_Conf::samples_per_ms {0.0}
```

Definition at line 50 of file [acq\\_conf.h](#).

#### 10.1.2.29 use\_automatic\_resampler

```
bool Acq_Conf::use_automatic_resampler {false}
```

Definition at line 74 of file [acq\\_conf.h](#).

#### 10.1.2.30 use\_CFAR\_algorithm\_flag

```
bool Acq_Conf::use_CFAR_algorithm_flag {true}
```

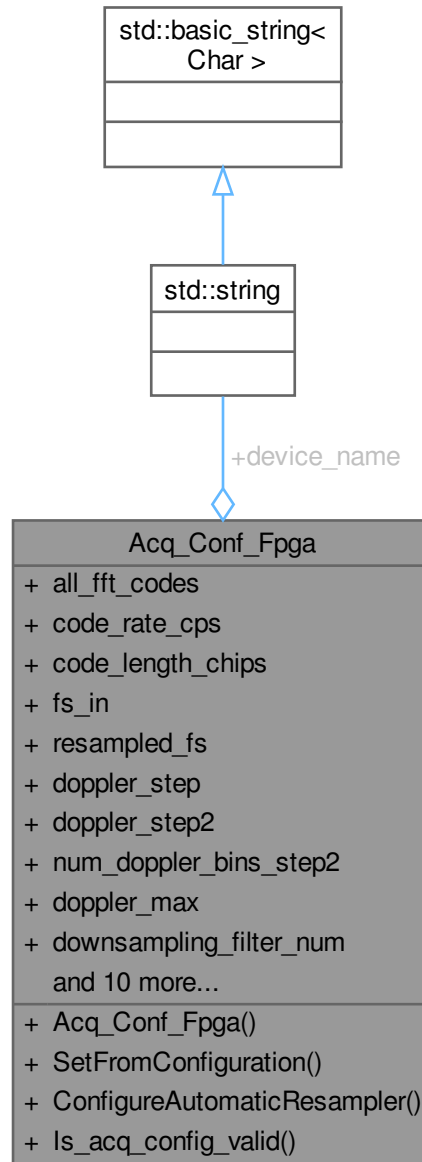
Definition at line 69 of file [acq\\_conf.h](#).

The documentation for this class was generated from the following file:

- [acq\\_conf.h](#)

## 10.2 Acq\_Conf\_Fpga Class Reference

Collaboration diagram for Acq\_Conf\_Fpga:



### Public Member Functions

- void **SetFromConfiguration** (const [ConfigurationInterface](#) \*configuration, const std::string &role, uint32\_t blk\_exp, double code\_chips\_per\_sec, double num\_chips\_per\_code)
- bool **ConfigureAutomaticResampler** (std::vector< std::pair< uint32\_t, uint32\_t > > downsampling\_filter\_specs, uint32\_t max\_FFT\_size, double opt\_freq)
- bool **Is\_acq\_config\_valid** (uint32\_t max\_FFT\_size)

## Public Attributes

- `std::string device_name` = "uio0"
- `uint32_t * all_fft_codes` = NULL
- `double code_rate_cps`
- `double code_length_chips`
- `int64_t fs_in` {4000000LL}
- `int64_t resampled_fs` {4000000LL}
- `float doppler_step` {250.0}
- `float doppler_step2` {125.0}
- `uint32_t num_doppler_bins_step2` {4U}
- `int32_t doppler_max` {5000}
- `uint32_t downsampling_filter_num` {0U}
- `uint32_t downsampling_factor` {1U}
- `uint32_t downsampling_filter_delay` {0U}
- `uint32_t total_block_exp` {13U}
- `uint32_t excludelimit` {5U}
- `uint32_t max_num_acqs` {2U}
- `uint32_t fft_size` {1U}
- `uint32_t code_length` {16000U}
- `bool make_2_steps` {false}
- `bool enable_zero_padding` {false}
- `bool repeat_satellite` {false}

### 10.2.1 Detailed Description

Definition at line 35 of file [acq\\_conf\\_fpga.h](#).

### 10.2.2 Member Data Documentation

#### 10.2.2.1 all\_fft\_codes

```
uint32_t* Acq_Conf_Fpga::all_fft_codes = NULL
```

Definition at line 48 of file [acq\\_conf\\_fpga.h](#).

#### 10.2.2.2 code\_length

```
uint32_t Acq_Conf_Fpga::code_length {16000U}
```

Definition at line 64 of file [acq\\_conf\\_fpga.h](#).

#### 10.2.2.3 code\_length\_chips

```
double Acq_Conf_Fpga::code_length_chips
```

Definition at line 50 of file [acq\\_conf\\_fpga.h](#).

#### 10.2.2.4 code\_rate\_cps

```
double Acq_Conf_Fpga::code_rate_cps
```

Definition at line 49 of file [acq\\_conf\\_fpga.h](#).

#### 10.2.2.5 device\_name

```
std::string Acq_Conf_Fpga::device_name = "uio0"
```

Definition at line 47 of file [acq\\_conf\\_fpga.h](#).

#### 10.2.2.6 doppler\_max

```
int32_t Acq_Conf_Fpga::doppler_max {5000}
```

Definition at line 56 of file [acq\\_conf\\_fpga.h](#).



### 10.2.2.7 doppler\_step

float Acq\_Conf\_Fpga::doppler\_step {250.0}  
Definition at line 53 of file [acq\\_conf\\_fpga.h](#).

### 10.2.2.8 doppler\_step2

float Acq\_Conf\_Fpga::doppler\_step2 {125.0}  
Definition at line 54 of file [acq\\_conf\\_fpga.h](#).

### 10.2.2.9 downsampling\_factor

uint32\_t Acq\_Conf\_Fpga::downsampling\_factor {1U}  
Definition at line 58 of file [acq\\_conf\\_fpga.h](#).

### 10.2.2.10 downsampling\_filter\_delay

uint32\_t Acq\_Conf\_Fpga::downsampling\_filter\_delay {0U}  
Definition at line 59 of file [acq\\_conf\\_fpga.h](#).

### 10.2.2.11 downsampling\_filter\_num

uint32\_t Acq\_Conf\_Fpga::downsampling\_filter\_num {0U}  
Definition at line 57 of file [acq\\_conf\\_fpga.h](#).

### 10.2.2.12 enable\_zero\_padding

bool Acq\_Conf\_Fpga::enable\_zero\_padding {false}  
Definition at line 66 of file [acq\\_conf\\_fpga.h](#).

### 10.2.2.13 excludelimit

uint32\_t Acq\_Conf\_Fpga::excludelimit {5U}  
Definition at line 61 of file [acq\\_conf\\_fpga.h](#).

### 10.2.2.14 fft\_size

uint32\_t Acq\_Conf\_Fpga::fft\_size {1U}  
Definition at line 63 of file [acq\\_conf\\_fpga.h](#).

### 10.2.2.15 fs\_in

int64\_t Acq\_Conf\_Fpga::fs\_in {4000000LL}  
Definition at line 51 of file [acq\\_conf\\_fpga.h](#).

### 10.2.2.16 make\_2\_steps

bool Acq\_Conf\_Fpga::make\_2\_steps {false}  
Definition at line 65 of file [acq\\_conf\\_fpga.h](#).

### 10.2.2.17 max\_num\_acqs

uint32\_t Acq\_Conf\_Fpga::max\_num\_acqs {2U}  
Definition at line 62 of file [acq\\_conf\\_fpga.h](#).

### 10.2.2.18 num\_doppler\_bins\_step2

uint32\_t Acq\_Conf\_Fpga::num\_doppler\_bins\_step2 {4U}  
Definition at line 55 of file [acq\\_conf\\_fpga.h](#).

### 10.2.2.19 repeat\_satellite

```
bool Acq_Conf_Fpga::repeat_satellite {false}
```

Definition at line 67 of file [acq\\_conf\\_fpga.h](#).

### 10.2.2.20 resampled\_fs

```
int64_t Acq_Conf_Fpga::resampled_fs {4000000LL}
```

Definition at line 52 of file [acq\\_conf\\_fpga.h](#).

### 10.2.2.21 total\_block\_exp

```
uint32_t Acq_Conf_Fpga::total_block_exp {13U}
```

Definition at line 60 of file [acq\\_conf\\_fpga.h](#).

The documentation for this class was generated from the following file:

- [acq\\_conf\\_fpga.h](#)

## 10.3 AcquisitionInterface Class Reference

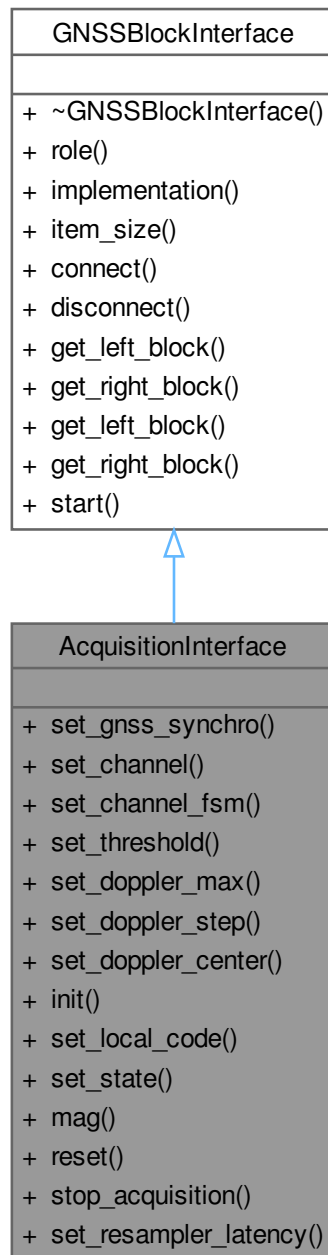
This abstract class represents an interface to an acquisition GNSS block.

```
#include <acquisition_interface.h>
```

Inheritance diagram for AcquisitionInterface:



Collaboration diagram for AcquisitionInterface:



### Public Member Functions

- virtual void [set\\_gnss\\_synchro](#) ([Gnss\\_Synchro](#) \*gnss\_synchro)=0
- virtual void [set\\_channel](#) (unsigned int channel\_id)=0
- virtual void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm)=0
- virtual void [set\\_threshold](#) (float threshold)=0
- virtual void [set\\_doppler\\_max](#) (unsigned int doppler\_max)=0
- virtual void [set\\_doppler\\_step](#) (unsigned int doppler\_step)=0

- virtual void [set\\_doppler\\_center](#) (int doppler\_center \_\_attribute\_\_((unused)))
- virtual void [init](#) ()=0
- virtual void [set\\_local\\_code](#) ()=0
- virtual void [set\\_state](#) (int state)=0
- virtual signed int [mag](#) ()=0
- virtual void [reset](#) ()=0
- virtual void [stop\\_acquisition](#) ()=0
- virtual void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples)=0

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual std::string [role](#) ()=0
- virtual std::string [implementation](#) ()=0
- virtual size\_t [item\\_size](#) ()=0
- virtual void [connect](#) (gr::top\_block\_sptr top\_block)=0
- virtual void [disconnect](#) (gr::top\_block\_sptr top\_block)=0
- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) ()=0
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) ()=0
- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.3.1 Detailed Description

This abstract class represents an interface to an acquisition GNSS block.

Abstract class for acquisition algorithms. Since all its methods are virtual, this class cannot be instantiated directly, and a subclass can only be instantiated directly if all inherited pure virtual methods have been implemented by that class or a parent class.

Definition at line 50 of file [acquisition\\_interface.h](#).

### 10.3.2 Member Function Documentation

#### 10.3.2.1 [init\(\)](#)

```
virtual void AcquisitionInterface::init () [pure virtual]
```

Implemented in [BeidouB1iPcpsAcquisition](#), [BeidouB3iPcpsAcquisition](#), [GalileoE1Pcps8msAmbiguousAcquisition](#), [GalileoE1PcpsAmbiguousAcquisition](#), [GalileoE1PcpsAmbiguousAcquisitionFpga](#), [GalileoE1PcpsCccwsrAmbiguousAcquisition](#), [GalileoE1PcpsQuickSyncAmbiguousAcquisition](#), [GalileoE1PcpsTongAmbiguousAcquisition](#), [GalileoE5aNoncoherentIQAcquisitionCa](#), [GalileoE5aPcpsAcquisition](#), [GalileoE5aPcpsAcquisitionFpga](#), [GalileoE5bPcpsAcquisition](#), [GalileoE5bPcpsAcquisitionFpga](#), [GalileoE6PcpsAcquisition](#), [GlonassL1CaPcpsAcquisition](#), [GlonassL2CaPcpsAcquisition](#), [GpsL1CaPcpsAcquisition](#), [GpsL1CaPcpsAcquisitionFineDoppler](#), [GpsL1CaPcpsAcquisitionFpga](#), [GpsL1CaPcpsAssistedAcquisition](#), [GpsL1CaPcpsOpenCIAcquisition](#), [GpsL1CaPcpsQuickSyncAcquisition](#), [GpsL1CaPcpsTongAcquisition](#), [GpsL2MPcpsAcquisition](#), [GpsL2MPcpsAcquisitionFpga](#), [GpsL5iPcpsAcquisition](#), and [GpsL5iPcpsAcquisitionFpga](#).

#### 10.3.2.2 [mag\(\)](#)

```
virtual signed int AcquisitionInterface::mag () [pure virtual]
```

Implemented in [BeidouB1iPcpsAcquisition](#), [BeidouB3iPcpsAcquisition](#), [GalileoE1Pcps8msAmbiguousAcquisition](#), [GalileoE1PcpsAmbiguousAcquisition](#), [GalileoE1PcpsAmbiguousAcquisitionFpga](#), [GalileoE1PcpsCccwsrAmbiguousAcquisition](#), [GalileoE1PcpsQuickSyncAmbiguousAcquisition](#), [GalileoE1PcpsTongAmbiguousAcquisition](#), [GalileoE5aNoncoherentIQAcquisitionCa](#), [GalileoE5aPcpsAcquisition](#), [GalileoE5aPcpsAcquisitionFpga](#), [GalileoE5bPcpsAcquisition](#), [GalileoE5bPcpsAcquisitionFpga](#), [GalileoE6PcpsAcquisition](#), [GlonassL1CaPcpsAcquisition](#), [GlonassL2CaPcpsAcquisition](#), [GpsL1CaPcpsAcquisition](#), [GpsL1CaPcpsAcquisitionFineDoppler](#), [GpsL1CaPcpsAcquisitionFpga](#), [GpsL1CaPcpsAssistedAcquisition](#), [GpsL1CaPcpsOpenCIAcquisition](#), [GpsL1CaPcpsQuickSyncAcquisition](#), [GpsL1CaPcpsTongAcquisition](#), [GpsL2MPcpsAcquisition](#), [GpsL2MPcpsAcquisitionFpga](#), [GpsL5iPcpsAcquisition](#), and [GpsL5iPcpsAcquisitionFpga](#).

### 10.3.2.3 reset()

```
virtual void AcquisitionInterface::reset () [pure virtual]
```

Implemented in [BeidouB1iPcpsAcquisition](#), [BeidouB3iPcpsAcquisition](#), [GalileoE1Pcps8msAmbiguousAcquisition](#), [GalileoE1PcpsAmbiguousAcquisition](#), [GalileoE1PcpsAmbiguousAcquisitionFpga](#), [GalileoE1PcpsCccwsrAmbiguousAcquisition](#), [GalileoE1PcpsQuickSyncAmbiguousAcquisition](#), [GalileoE1PcpsTongAmbiguousAcquisition](#), [GalileoE5aNoncoherentIQAquisitionCaf](#), [GalileoE5aPcpsAcquisition](#), [GalileoE5aPcpsAcquisitionFpga](#), [GalileoE5bPcpsAcquisition](#), [GalileoE5bPcpsAcquisitionFpga](#), [GalileoE6PcpsAcquisition](#), [GlonassL1CaPcpsAcquisition](#), [GlonassL2CaPcpsAcquisition](#), [GpsL1CaPcpsAcquisition](#), [GpsL1CaPcpsAcquisitionFineDoppler](#), [GpsL1CaPcpsAcquisitionFpga](#), [GpsL1CaPcpsAssistedAcquisition](#), [GpsL1CaPcpsOpenCIAquisition](#), [GpsL1CaPcpsQuickSyncAcquisition](#), [GpsL1CaPcpsTongAcquisition](#), [GpsL2MPcpsAcquisition](#), [GpsL2MPcpsAcquisitionFpga](#), [GpsL5iPcpsAcquisition](#), and [GpsL5iPcpsAcquisitionFpga](#).

### 10.3.2.4 set\_channel()

```
virtual void AcquisitionInterface::set_channel (
    unsigned int channel_id) [pure virtual]
```

Implemented in [BeidouB1iPcpsAcquisition](#), [BeidouB3iPcpsAcquisition](#), [GalileoE1Pcps8msAmbiguousAcquisition](#), [GalileoE1PcpsAmbiguousAcquisition](#), [GalileoE1PcpsAmbiguousAcquisitionFpga](#), [GalileoE1PcpsCccwsrAmbiguousAcquisition](#), [GalileoE1PcpsQuickSyncAmbiguousAcquisition](#), [GalileoE1PcpsTongAmbiguousAcquisition](#), [GalileoE5aNoncoherentIQAquisitionCaf](#), [GalileoE5aPcpsAcquisition](#), [GalileoE5aPcpsAcquisitionFpga](#), [GalileoE5bPcpsAcquisition](#), [GalileoE5bPcpsAcquisitionFpga](#), [GalileoE6PcpsAcquisition](#), [GlonassL1CaPcpsAcquisition](#), [GlonassL2CaPcpsAcquisition](#), [GpsL1CaPcpsAcquisition](#), [GpsL1CaPcpsAcquisitionFineDoppler](#), [GpsL1CaPcpsAcquisitionFpga](#), [GpsL1CaPcpsAssistedAcquisition](#), [GpsL1CaPcpsOpenCIAquisition](#), [GpsL1CaPcpsQuickSyncAcquisition](#), [GpsL1CaPcpsTongAcquisition](#), [GpsL2MPcpsAcquisition](#), [GpsL2MPcpsAcquisitionFpga](#), [GpsL5iPcpsAcquisition](#), and [GpsL5iPcpsAcquisitionFpga](#).

### 10.3.2.5 set\_channel\_fsm()

```
virtual void AcquisitionInterface::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [pure virtual]
```

Implemented in [BeidouB1iPcpsAcquisition](#), [BeidouB3iPcpsAcquisition](#), [GalileoE1Pcps8msAmbiguousAcquisition](#), [GalileoE1PcpsAmbiguousAcquisition](#), [GalileoE1PcpsAmbiguousAcquisitionFpga](#), [GalileoE1PcpsCccwsrAmbiguousAcquisition](#), [GalileoE1PcpsQuickSyncAmbiguousAcquisition](#), [GalileoE1PcpsTongAmbiguousAcquisition](#), [GalileoE5aNoncoherentIQAquisitionCaf](#), [GalileoE5aPcpsAcquisition](#), [GalileoE5aPcpsAcquisitionFpga](#), [GalileoE5bPcpsAcquisition](#), [GalileoE5bPcpsAcquisitionFpga](#), [GalileoE6PcpsAcquisition](#), [GlonassL1CaPcpsAcquisition](#), [GlonassL2CaPcpsAcquisition](#), [GpsL1CaPcpsAcquisition](#), [GpsL1CaPcpsAcquisitionFineDoppler](#), [GpsL1CaPcpsAcquisitionFpga](#), [GpsL1CaPcpsAssistedAcquisition](#), [GpsL1CaPcpsOpenCIAquisition](#), [GpsL1CaPcpsQuickSyncAcquisition](#), [GpsL1CaPcpsTongAcquisition](#), [GpsL2MPcpsAcquisition](#), [GpsL2MPcpsAcquisitionFpga](#), [GpsL5iPcpsAcquisition](#), and [GpsL5iPcpsAcquisitionFpga](#).

### 10.3.2.6 set\_doppler\_center()

```
virtual void AcquisitionInterface::set_doppler_center (
    int doppler_center __attribute__((unused))) [inline], [virtual]
```

Definition at line 59 of file [acquisition\\_interface.h](#).

### 10.3.2.7 set\_doppler\_max()

```
virtual void AcquisitionInterface::set_doppler_max (
    unsigned int doppler_max) [pure virtual]
```

Implemented in [BeidouB3iPcpsAcquisition](#), [GalileoE1Pcps8msAmbiguousAcquisition](#), [GalileoE1PcpsAmbiguousAcquisition](#), [GalileoE1PcpsAmbiguousAcquisitionFpga](#), [GalileoE1PcpsCccwsrAmbiguousAcquisition](#), [GalileoE1PcpsQuickSyncAmbiguousAcquisition](#), [GalileoE1PcpsTongAmbiguousAcquisition](#), [GalileoE5aNoncoherentIQAquisitionCaf](#), [GalileoE5aPcpsAcquisition](#), [GalileoE5aPcpsAcquisitionFpga](#), [GalileoE5bPcpsAcquisition](#), [GalileoE5bPcpsAcquisitionFpga](#), [GalileoE6PcpsAcquisition](#), [GlonassL1CaPcpsAcquisition](#), [GlonassL2CaPcpsAcquisition](#), [GpsL1CaPcpsAcquisition](#), [GpsL1CaPcpsAcquisitionFineDoppler](#), [GpsL1CaPcpsAcquisitionFpga](#), [GpsL1CaPcpsAssistedAcquisition](#), [GpsL1CaPcpsOpenCIAquisition](#), [GpsL1CaPcpsQuickSyncAcquisition](#), [GpsL1CaPcpsTongAcquisition](#), [GpsL2MPcpsAcquisition](#), [GpsL2MPcpsAcquisitionFpga](#), [GpsL5iPcpsAcquisition](#), and [GpsL5iPcpsAcquisitionFpga](#).

### 10.3.2.8 set\_doppler\_step()

```
virtual void AcquisitionInterface::set_doppler_step (
```

```
    unsigned int doppler_step) [pure virtual]
```

Implemented in [BeidouB3iPcpsAcquisition](#), [GalileoE1Pcps8msAmbiguousAcquisition](#), [GalileoE1PcpsAmbiguousAcquisition](#), [GalileoE1PcpsAmbiguousAcquisitionFpga](#), [GalileoE1PcpsCccwsrAmbiguousAcquisition](#), [GalileoE1PcpsQuickSyncAmbiguousAcquisition](#), [GalileoE1PcpsTongAmbiguousAcquisition](#), [GalileoE5aNoncoherentIQAcquisitionCaf](#), [GalileoE5aPcpsAcquisition](#), [GalileoE5aPcpsAcquisitionFpga](#), [GalileoE5bPcpsAcquisition](#), [GalileoE5bPcpsAcquisitionFpga](#), [GalileoE6PcpsAcquisition](#), [GlonassL1CaPcpsAcquisition](#), [GlonassL2CaPcpsAcquisition](#), [GpsL1CaPcpsAcquisition](#), [GpsL1CaPcpsAcquisitionFineDoppler](#), [GpsL1CaPcpsAcquisitionFpga](#), [GpsL1CaPcpsAssistedAcquisition](#), [GpsL1CaPcpsOpenCIAcquisition](#), [GpsL1CaPcpsQuickSyncAcquisition](#), [GpsL1CaPcpsTongAcquisition](#), [GpsL2MPcpsAcquisition](#), [GpsL2MPcpsAcquisitionFpga](#), [GpsL5iPcpsAcquisition](#), and [GpsL5iPcpsAcquisitionFpga](#).

### 10.3.2.9 set\_gnss\_synchro()

```
virtual void AcquisitionInterface::set_gnss_synchro (
    Gnss_Synchro * gnss_synchro) [pure virtual]
```

Implemented in [BeidouB1iPcpsAcquisition](#), [BeidouB3iPcpsAcquisition](#), [GalileoE1Pcps8msAmbiguousAcquisition](#), [GalileoE1PcpsAmbiguousAcquisition](#), [GalileoE1PcpsAmbiguousAcquisitionFpga](#), [GalileoE1PcpsCccwsrAmbiguousAcquisition](#), [GalileoE1PcpsQuickSyncAmbiguousAcquisition](#), [GalileoE1PcpsTongAmbiguousAcquisition](#), [GalileoE5aNoncoherentIQAcquisitionCaf](#), [GalileoE5aPcpsAcquisition](#), [GalileoE5aPcpsAcquisitionFpga](#), [GalileoE5bPcpsAcquisition](#), [GalileoE5bPcpsAcquisitionFpga](#), [GalileoE6PcpsAcquisition](#), [GlonassL1CaPcpsAcquisition](#), [GlonassL2CaPcpsAcquisition](#), [GpsL1CaPcpsAcquisition](#), [GpsL1CaPcpsAcquisitionFineDoppler](#), [GpsL1CaPcpsAcquisitionFpga](#), [GpsL1CaPcpsAssistedAcquisition](#), [GpsL1CaPcpsOpenCIAcquisition](#), [GpsL1CaPcpsQuickSyncAcquisition](#), [GpsL1CaPcpsTongAcquisition](#), [GpsL2MPcpsAcquisition](#), [GpsL2MPcpsAcquisitionFpga](#), [GpsL5iPcpsAcquisition](#), and [GpsL5iPcpsAcquisitionFpga](#).

### 10.3.2.10 set\_local\_code()

```
virtual void AcquisitionInterface::set_local_code () [pure virtual]
```

Implemented in [BeidouB1iPcpsAcquisition](#), [BeidouB3iPcpsAcquisition](#), [GalileoE1Pcps8msAmbiguousAcquisition](#), [GalileoE1PcpsAmbiguousAcquisition](#), [GalileoE1PcpsAmbiguousAcquisitionFpga](#), [GalileoE1PcpsQuickSyncAmbiguousAcquisition](#), [GalileoE1PcpsTongAmbiguousAcquisition](#), [GalileoE5aNoncoherentIQAcquisitionCaf](#), [GalileoE5aPcpsAcquisition](#), [GalileoE5aPcpsAcquisitionFpga](#), [GalileoE5bPcpsAcquisition](#), [GalileoE5bPcpsAcquisitionFpga](#), [GalileoE6PcpsAcquisition](#), [GlonassL1CaPcpsAcquisition](#), [GlonassL2CaPcpsAcquisition](#), [GpsL1CaPcpsAcquisition](#), [GpsL1CaPcpsAcquisitionFpga](#), [GpsL1CaPcpsOpenCIAcquisition](#), [GpsL1CaPcpsQuickSyncAcquisition](#), [GpsL1CaPcpsTongAcquisition](#), [GpsL2MPcpsAcquisition](#), [GpsL2MPcpsAcquisitionFpga](#), [GpsL5iPcpsAcquisition](#), and [GpsL5iPcpsAcquisitionFpga](#).

### 10.3.2.11 set\_resampler\_latency()

```
virtual void AcquisitionInterface::set_resampler_latency (
    uint32_t latency_samples) [pure virtual]
```

Implemented in [BeidouB1iPcpsAcquisition](#), [BeidouB3iPcpsAcquisition](#), [GalileoE1PcpsAmbiguousAcquisition](#), [GalileoE5aPcpsAcquisition](#), [GalileoE5bPcpsAcquisition](#), [GalileoE6PcpsAcquisition](#), [GpsL1CaPcpsAcquisition](#), [GpsL2MPcpsAcquisition](#), and [GpsL5iPcpsAcquisition](#).

### 10.3.2.12 set\_state()

```
virtual void AcquisitionInterface::set_state (
    int state) [pure virtual]
```

Implemented in [BeidouB1iPcpsAcquisition](#), [BeidouB3iPcpsAcquisition](#), [GalileoE1PcpsAmbiguousAcquisition](#), [GalileoE1PcpsAmbiguousAcquisitionFpga](#), [GalileoE1PcpsCccwsrAmbiguousAcquisition](#), [GalileoE1PcpsQuickSyncAmbiguousAcquisition](#), [GalileoE1PcpsTongAmbiguousAcquisition](#), [GalileoE5aNoncoherentIQAcquisitionCaf](#), [GalileoE5aPcpsAcquisition](#), [GalileoE5aPcpsAcquisitionFpga](#), [GalileoE5bPcpsAcquisition](#), [GalileoE5bPcpsAcquisitionFpga](#), [GalileoE6PcpsAcquisition](#), [GlonassL1CaPcpsAcquisition](#), [GlonassL2CaPcpsAcquisition](#), [GpsL1CaPcpsAcquisition](#), [GpsL1CaPcpsAcquisitionFineDoppler](#), [GpsL1CaPcpsAcquisitionFpga](#), [GpsL1CaPcpsQuickSyncAcquisition](#), [GpsL1CaPcpsTongAcquisition](#), [GpsL2MPcpsAcquisition](#), [GpsL2MPcpsAcquisitionFpga](#), [GpsL5iPcpsAcquisition](#), and [GpsL5iPcpsAcquisitionFpga](#).

### 10.3.2.13 set\_threshold()

```
virtual void AcquisitionInterface::set_threshold (
    float threshold) [pure virtual]
```

Implemented in [BeidouB1iPcpsAcquisition](#), [BeidouB3iPcpsAcquisition](#), [GalileoE1Pcps8msAmbiguousAcquisition](#), [GalileoE1PcpsAmbiguousAcquisition](#), [GalileoE1PcpsAmbiguousAcquisitionFpga](#), [GalileoE1PcpsCccwsrAmbiguousAcquisition](#), [GalileoE1PcpsQuickSyncAmbiguousAcquisition](#), [GalileoE1PcpsTongAmbiguousAcquisition](#), [GalileoE5aNoncoherentIQAquisitionCa](#), [GalileoE5aPcpsAcquisition](#), [GalileoE5aPcpsAcquisitionFpga](#), [GalileoE5bPcpsAcquisition](#), [GalileoE5bPcpsAcquisitionFpga](#), [GalileoE6PcpsAcquisition](#), [GlonassL1CaPcpsAcquisition](#), [GlonassL2CaPcpsAcquisition](#), [GpsL1CaPcpsAcquisition](#), [GpsL1CaPcpsAcquisitionFineDoppler](#), [GpsL1CaPcpsAssistedAcquisition](#), [GpsL1CaPcpsOpenCIAquisition](#), [GpsL1CaPcpsQuickSyncAcquisition](#), [GpsL1CaPcpsTongAcquisition](#), [GpsL2MPcpsAcquisition](#), [GpsL2MPcpsAcquisitionFpga](#), [GpsL5iPcpsAcquisition](#), and [GpsL5iPcpsAcquisitionFpga](#).

#### 10.3.2.14 stop\_acquisition()

```
virtual void AcquisitionInterface::stop_acquisition () [pure virtual]
```

Implemented in [BeidouB1iPcpsAcquisition](#), [BeidouB3iPcpsAcquisition](#), [GalileoE1Pcps8msAmbiguousAcquisition](#), [GalileoE1PcpsAmbiguousAcquisition](#), [GalileoE1PcpsAmbiguousAcquisitionFpga](#), [GalileoE1PcpsCccwsrAmbiguousAcquisition](#), [GalileoE1PcpsQuickSyncAmbiguousAcquisition](#), [GalileoE1PcpsTongAmbiguousAcquisition](#), [GalileoE5aNoncoherentIQAquisitionCa](#), [GalileoE5aPcpsAcquisition](#), [GalileoE5aPcpsAcquisitionFpga](#), [GalileoE5bPcpsAcquisition](#), [GalileoE5bPcpsAcquisitionFpga](#), [GalileoE6PcpsAcquisition](#), [GlonassL1CaPcpsAcquisition](#), [GlonassL2CaPcpsAcquisition](#), [GpsL1CaPcpsAcquisition](#), [GpsL1CaPcpsAcquisitionFineDoppler](#), [GpsL1CaPcpsAcquisitionFpga](#), [GpsL1CaPcpsAssistedAcquisition](#), [GpsL1CaPcpsOpenCIAquisition](#), [GpsL1CaPcpsQuickSyncAcquisition](#), [GpsL1CaPcpsTongAcquisition](#), [GpsL2MPcpsAcquisition](#), [GpsL2MPcpsAcquisitionFpga](#), [GpsL5iPcpsAcquisition](#), and [GpsL5iPcpsAcquisitionFpga](#).

The documentation for this class was generated from the following file:

- [acquisition\\_interface.h](#)

## 10.4 ad936x\_iio\_custom Class Reference

Collaboration diagram for ad936x\_iio\_custom:

ad936x_iio_custom
+ n_channels
+ ad936x_iio_custom()
+ ~ad936x_iio_custom()
+ initialize_device()
+ init_config_ad9361_rx()
+ calibrate()
+ get_rx_gain()
+ setRXGain()
+ set_antenna_port()
+ get_frequency()
+ set_frequency()
and 9 more...

### Public Member Functions

- **ad936x\_iio\_custom** (int debug\_level\_, int log\_level\_)
- **bool initialize\_device** (const std::string &pluto\_device\_uri, const std::string &board\_type)

- bool **init\_config\_ad9361\_rx** (long long bandwidth\_, long long sample\_rate\_, long long freq\_, const std::string &rf\_port\_select\_, const std::string &rf\_filter, const std::string &gain\_mode\_rx0\_, const std::string &gain\_mode\_rx1\_, double rf\_gain\_rx0\_, double rf\_gain\_rx1\_, bool enable\_ch0, bool enable\_ch1, long long freq\_2ch, double lo\_attenuation\_db\_, bool high\_side\_lo\_, int tx\_lo\_channel\_)
- bool **calibrate** (int ch, double bw\_hz)
- double **get\_rx\_gain** (int ch\_num)
- bool **setRXGain** (int ch\_num, const std::string &gain\_mode, double gain\_dB)
- bool **set\_antenna\_port** (int ch, int antenna\_idx)
- double **get\_frequency** (int ch)
- bool **set\_frequency** (int ch, double freq\_hz)
- bool **start\_sample\_rx** (bool ppsmode)
- void **stop\_record** ()
- void **set\_gnss\_time\_queue** (std::shared\_ptr< [Concurrent\\_Queue](#)< [GnssTime](#) > > queue)
- void **set\_pps\_samplestamp\_queue** (std::shared\_ptr< [Concurrent\\_Queue](#)< [PpsSamplestamp](#) > > queue)
- bool **get\_rx\_frequency** (long long &freq\_hz)
- bool **set\_rx\_frequency** (long long freq\_hz)
- bool **read\_die\_temp** (double &temp\_c)
- void **pop\_sample\_buffer** (std::shared\_ptr< [ad936x\\_iio\\_samples](#) > &current\_buffer)
- void **push\_sample\_buffer** (std::shared\_ptr< [ad936x\\_iio\\_samples](#) > &current\_buffer)

#### Public Attributes

- int [n\\_channels](#)

### 10.4.1 Detailed Description

Definition at line 39 of file [ad936x\\_iio\\_custom.h](#).

### 10.4.2 Member Data Documentation

#### 10.4.2.1 n\_channels

```
int ad936x_iio_custom::n_channels
```

Definition at line 84 of file [ad936x\\_iio\\_custom.h](#).

The documentation for this class was generated from the following file:

- [ad936x\\_iio\\_custom.h](#)

## 10.5 ad936x\_iio\_samples Class Reference

Collaboration diagram for [ad936x\\_iio\\_samples](#):

ad936x_iio_samples
+ n_bytes
+ n_interleaved_iq_samples
+ n_channels
+ step_bytes
+ buffer
+ ad936x_iio_samples()



**Public Attributes**

- uint32\_t [n\\_bytes](#) {0}
- uint32\_t [n\\_interleaved\\_iq\\_samples](#) {0}
- uint16\_t [n\\_channels](#) {0}
- uint16\_t [step\\_bytes](#) {0}
- char [buffer](#) [IIO\_DEFAULTAD936XAPIFIFO\_SIZE\_SAMPLES \*4 \*4]

**10.5.1 Detailed Description**

Definition at line 33 of file [ad936x\\_iio\\_samples.h](#).

**10.5.2 Member Data Documentation****10.5.2.1 buffer**

```
char ad936x_iio_samples::buffer[IIO_DEFAULTAD936XAPIFIFO_SIZE_SAMPLES *4 *4]
```

Definition at line 41 of file [ad936x\\_iio\\_samples.h](#).

**10.5.2.2 n\_bytes**

```
uint32_t ad936x_iio_samples::n_bytes {0}
```

Definition at line 37 of file [ad936x\\_iio\\_samples.h](#).

**10.5.2.3 n\_channels**

```
uint16_t ad936x_iio_samples::n_channels {0}
```

Definition at line 39 of file [ad936x\\_iio\\_samples.h](#).

**10.5.2.4 n\_interleaved\_iq\_samples**

```
uint32_t ad936x_iio_samples::n_interleaved_iq_samples {0}
```

Definition at line 38 of file [ad936x\\_iio\\_samples.h](#).

**10.5.2.5 step\_bytes**

```
uint16_t ad936x_iio_samples::step_bytes {0}
```

Definition at line 40 of file [ad936x\\_iio\\_samples.h](#).

The documentation for this class was generated from the following file:

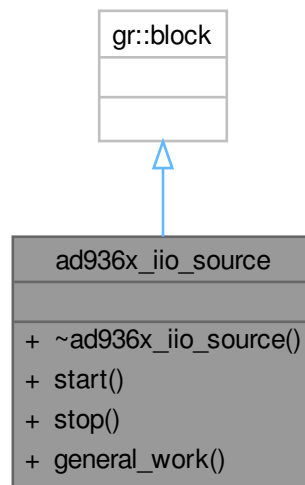
- [ad936x\\_iio\\_samples.h](#)

**10.6 ad936x\_iio\_source Class Reference**

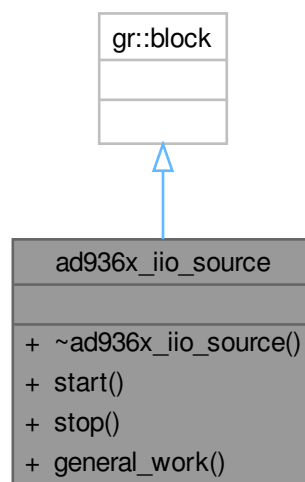
This class implements conversion between Labsat 2, 3 and 3 Wideband formats to gr\_complex.

```
#include <ad936x_iio_source.h>
```

Inheritance diagram for `ad936x_iio_source`:



Collaboration diagram for `ad936x_iio_source`:



### Public Member Functions

- `bool start ()`  
*start the sample transmission*
- `bool stop ()`  
*stop the sample transmission*

- int **general\_work** (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)

### Friends

- ad936x\_iio\_source\_sptr **ad936x\_iio\_make\_source\_sptr** (const std::string &pluto\_uri\_, const std::string &board\_type\_, int64\_t bandwidth\_, int64\_t sample\_rate\_, int64\_t freq\_, const std::string &rf\_port\_select\_, const std::string &rf\_filter, const std::string &gain\_mode\_rx0\_, const std::string &gain\_mode\_rx1\_, double rf\_gain\_rx0\_, double rf\_gain\_rx1\_, bool enable\_ch0, bool enable\_ch1, int64\_t freq\_2ch, bool ppsmode\_, bool customsamplesize\_, const std::string &fe\_ip\_, int fe\_ctlport\_, int ssize\_, int bshift\_, bool spattern\_, double lo\_attenuation\_db\_, bool high\_side\_lo\_, int tx\_lo\_channel\_)

## 10.6.1 Detailed Description

This class implements conversion between Labsat 2, 3 and 3 Wideband formats to gr\_complex.

Definition at line 77 of file [ad936x\\_iio\\_source.h](#).

## 10.6.2 Member Function Documentation

### 10.6.2.1 start()

```
bool ad936x_iio_source::start ()
start the sample transmission
```

### 10.6.2.2 stop()

```
bool ad936x_iio_source::stop ()
stop the sample transmission
```

The documentation for this class was generated from the following file:

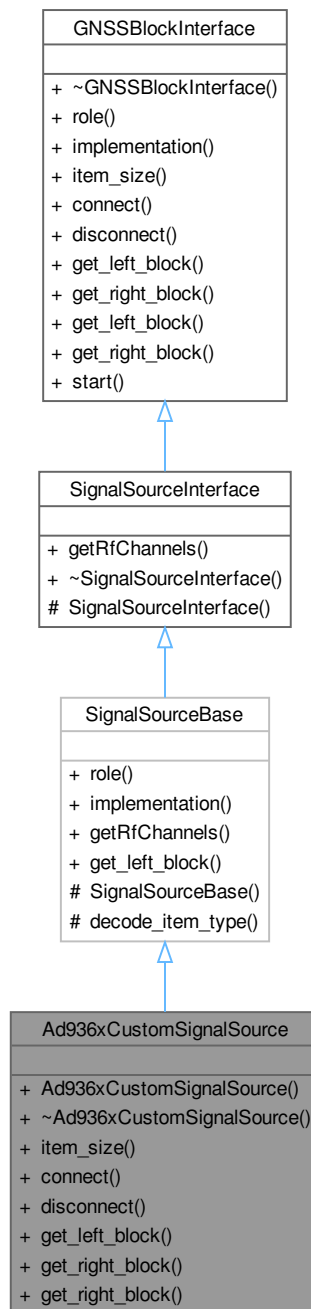
- [ad936x\\_iio\\_source.h](#)

## 10.7 Ad936xCustomSignalSource Class Reference

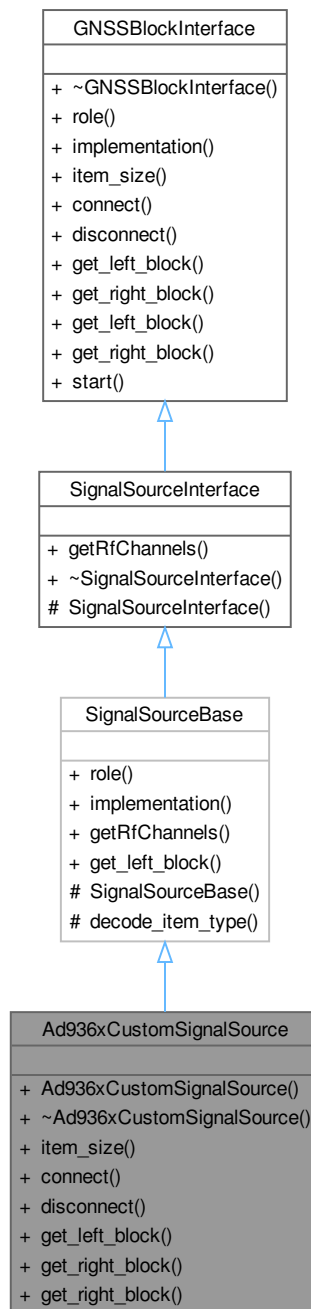
This class instantiates the Ad936xCustom gnuradio signal source. It has support also for a customized Ad936xCustom firmware and signal source to support PPS samplestamp reading.

```
#include <ad936x_custom_signal_source.h>
```

Inheritance diagram for Ad936xCustomSignalSource:



Collaboration diagram for Ad936xCustomSignalSource:



### Public Member Functions

- **Ad936xCustomSignalSource** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_stream, unsigned int out\_stream, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)
- `size_t item_size ()` override
- void `connect` (gr::top\_block\_sptr top\_block) override
- void `disconnect` (gr::top\_block\_sptr top\_block) override
- `gr::basic_block_sptr get_left_block ()` override

- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel) override

### Public Member Functions inherited from [SignalSourceBase](#)

- std::string [role](#) () final
- std::string [implementation](#) () final
- size\_t [getRfChannels](#) () const override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### Additional Inherited Members

### Protected Member Functions inherited from [SignalSourceBase](#)

- [SignalSourceBase](#) ([ConfigurationInterface](#) const \*configuration, std::string role, std::string impl)  
*Constructor.*
- size\_t [decode\\_item\\_type](#) (std::string const &item\_type, bool \*is\_interleaved=nullptr, bool throw\_on\_↵ error=false)  
*utility for decoding passed ".item\_type" values*

## 10.7.1 Detailed Description

This class instantiates the Ad936xCustom gnuradio signal source. It has support also for a customized Ad936x↵ Custom firmware and signal source to support PPS samplestamp reading.  
Definition at line 51 of file [ad936x\\_custom\\_signal\\_source.h](#).

## 10.7.2 Member Function Documentation

### 10.7.2.1 connect()

```
void Ad936xCustomSignalSource::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.7.2.2 disconnect()

```
void Ad936xCustomSignalSource::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.7.2.3 get\_left\_block()

```
gr::basic_block_sptr Ad936xCustomSignalSource::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.7.2.4 get\_right\_block() [1/2]

```
gr::basic_block_sptr Ad936xCustomSignalSource::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.7.2.5 `get_right_block()` [2/2]

```
gr::basic_block_sptr Ad936xCustomSignalSource::get_right_block (  
    int RF_channel) [override], [virtual]
```

Reimplemented from [GNSSBlockInterface](#).

### 10.7.2.6 `item_size()`

```
size_t Ad936xCustomSignalSource::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

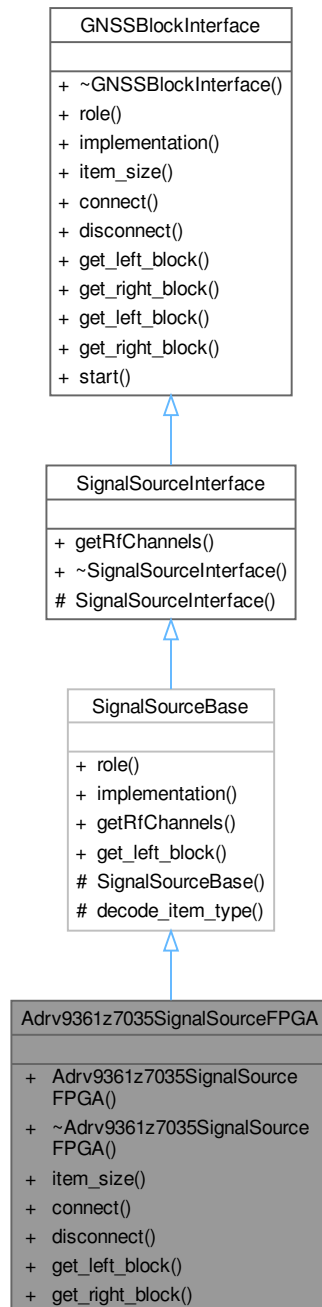
Definition at line 60 of file [ad936x\\_custom\\_signal\\_source.h](#).

The documentation for this class was generated from the following file:

- [ad936x\\_custom\\_signal\\_source.h](#)

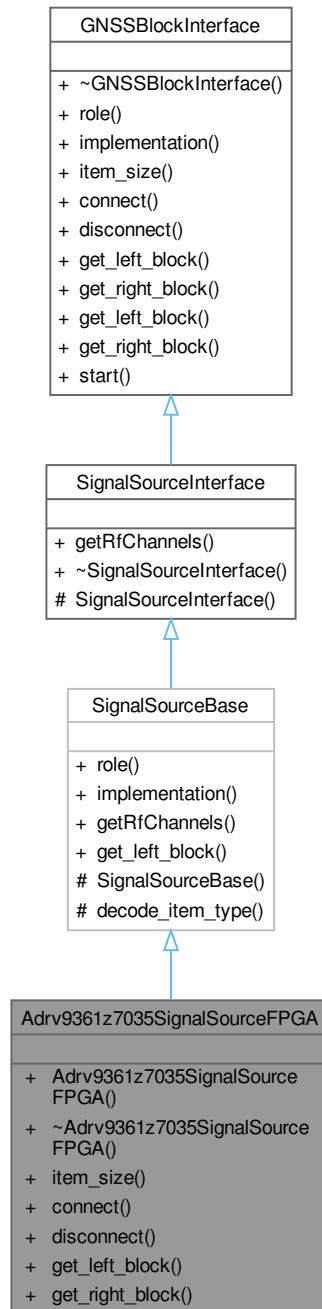
## 10.8 Adrv9361z7035SignalSourceFPGA Class Reference

Inheritance diagram for Adrv9361z7035SignalSourceFPGA:





Collaboration diagram for Adrv9361z7035SignalSourceFPGA:



## Public Member Functions

- **Adrv9361z7035SignalSourceFPGA** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_stream, unsigned int out\_stream, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override

## Public Member Functions inherited from [SignalSourceBase](#)

- `std::string` [role](#) () final
- `std::string` [implementation](#) () final
- `size_t` [getRfChannels](#) () const override
- `gr::basic_block_sptr` [get\\_left\\_block](#) () override

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual `gr::basic_block_sptr` [get\\_left\\_block](#) (int RF\_channel)
- virtual `gr::basic_block_sptr` [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

## Additional Inherited Members

## Protected Member Functions inherited from [SignalSourceBase](#)

- [SignalSourceBase](#) ([ConfigurationInterface](#) const \*configuration, `std::string` role, `std::string` impl)  
*Constructor.*
- `size_t` [decode\\_item\\_type](#) (`std::string` const &item\_type, `bool` \*is\_interleaved=nullptr, `bool` throw\_on\_↵  
error=false)  
*utility for decoding passed ".item\_type" values*

## 10.8.1 Detailed Description

Definition at line 46 of file [adv9361\\_z7035\\_signal\\_source\\_fpga.h](#).

## 10.8.2 Member Function Documentation

### 10.8.2.1 connect()

```
void Adv9361z7035SignalSourceFPGA::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.8.2.2 disconnect()

```
void Adv9361z7035SignalSourceFPGA::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.8.2.3 get\_left\_block()

```
gr::basic_block_sptr Adv9361z7035SignalSourceFPGA::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.8.2.4 get\_right\_block()

```
gr::basic_block_sptr Adv9361z7035SignalSourceFPGA::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.8.2.5 item\_size()

```
size_t Adv9361z7035SignalSourceFPGA::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 55 of file [adv9361\\_z7035\\_signal\\_source\\_fpga.h](#).

The documentation for this class was generated from the following file:

- [adv9361\\_z7035\\_signal\\_source\\_fpga.h](#)

## 10.9 Agnss\_Ref\_Location Class Reference

Interface of an Assisted GNSS REFERENCE LOCATION storage.

```
#include <agnss_ref_location.h>
```

Collaboration diagram for Agnss\_Ref\_Location:

Agnss_Ref_Location
+ lat
+ lon
+ uncertainty
+ valid
+ Agnss_Ref_Location()
+ serialize()

### Public Member Functions

- [Agnss\\_Ref\\_Location](#) ()=default
- `template<class Archive>`  
void [serialize](#) (Archive &archive, const unsigned int version)

*Serialize is a boost standard method to be called by the boost XML serialization. Here is used to save the Ref location on disk file.*

### Public Attributes

- double [lat](#) {}
- double [lon](#) {}
- double [uncertainty](#) {}
- bool [valid](#) {}

### 10.9.1 Detailed Description

Interface of an Assisted GNSS REFERENCE LOCATION storage.

Definition at line 33 of file [agnss\\_ref\\_location.h](#).

### 10.9.2 Constructor & Destructor Documentation

#### 10.9.2.1 Agnss\_Ref\_Location()

```
Agnss_Ref_Location::Agnss_Ref_Location () [default]
```

Default constructor

### 10.9.3 Member Function Documentation

#### 10.9.3.1 serialize()

```
template<class Archive>
void Agnss_Ref_Location::serialize (
    Archive & archive,
    const unsigned int version) [inline]
```

Serialize is a boost standard method to be called by the boost XML serialization. Here is used to save the Ref location on disk file.

Definition at line 52 of file [agnss\\_ref\\_location.h](#).

## 10.9.4 Member Data Documentation

### 10.9.4.1 lat

```
double Agnss_Ref_Location::lat {}
```

Definition at line 41 of file [agnss\\_ref\\_location.h](#).

### 10.9.4.2 lon

```
double Agnss_Ref_Location::lon {}
```

Definition at line 42 of file [agnss\\_ref\\_location.h](#).

### 10.9.4.3 uncertainty

```
double Agnss_Ref_Location::uncertainty {}
```

Definition at line 43 of file [agnss\\_ref\\_location.h](#).

### 10.9.4.4 valid

```
bool Agnss_Ref_Location::valid {}
```

Definition at line 44 of file [agnss\\_ref\\_location.h](#).

The documentation for this class was generated from the following file:

- [agnss\\_ref\\_location.h](#)

## 10.10 Agnss\_Ref\_Time Class Reference

Interface of an Assisted GNSS REFERENCE TIME storage.

```
#include <agnss_ref_time.h>
```

Collaboration diagram for Agnss\_Ref\_Time:

Agnss_Ref_Time
+ tow
+ week
+ seconds
+ microseconds
+ valid
+ Agnss_Ref_Time()
+ serialize()

### Public Member Functions

- [Agnss\\_Ref\\_Time](#) ()=default
- `template<class Archive>`  
void [serialize](#) (Archive &archive, const unsigned int version)

*Serialize is a boost standard method to be called by the boost XML serialization. Here is used to save the ref time data on disk file.*

### Public Attributes

- double [tow](#) {}
- double [week](#) {}
- double [seconds](#) {}
- double [microseconds](#) {}
- bool [valid](#) {}

## 10.10.1 Detailed Description

Interface of an Assisted GNSS REFERENCE TIME storage.

Definition at line 33 of file [agnss\\_ref\\_time.h](#).

## 10.10.2 Constructor & Destructor Documentation

### 10.10.2.1 Agnss\_Ref\_Time()

```
Agnss_Ref_Time::Agnss_Ref_Time () [default]
```

Default constructor

## 10.10.3 Member Function Documentation

### 10.10.3.1 serialize()

```
template<class Archive>
void Agnss_Ref_Time::serialize (
    Archive & archive,
    const unsigned int version) [inline]
```

Serialize is a boost standard method to be called by the boost XML serialization. Here is used to save the ref time data on disk file.

Definition at line 53 of file [agnss\\_ref\\_time.h](#).

## 10.10.4 Member Data Documentation

### 10.10.4.1 microseconds

```
double Agnss_Ref_Time::microseconds {}
```

Definition at line 44 of file [agnss\\_ref\\_time.h](#).

### 10.10.4.2 seconds

```
double Agnss_Ref_Time::seconds {}
```

Definition at line 43 of file [agnss\\_ref\\_time.h](#).

### 10.10.4.3 tow

```
double Agnss_Ref_Time::tow {}
```

Definition at line 41 of file [agnss\\_ref\\_time.h](#).

### 10.10.4.4 valid

```
bool Agnss_Ref_Time::valid {}
```

Definition at line 45 of file [agnss\\_ref\\_time.h](#).

#### 10.10.4.5 week

double Agnss\_Ref\_Time::week {}

Definition at line 42 of file [agnss\\_ref\\_time.h](#).

The documentation for this class was generated from the following file:

- [agnss\\_ref\\_time.h](#)

## 10.11 Allocator Struct Reference

Collaboration diagram for Allocator:

Allocator
+ countwords_
+ buffer_
+ Allocator()
+ operator()()

### Public Member Functions

- [Allocator](#) (size\_t countwords, void \*&buffer)
- [template<typename WordType>](#)  
void [operator](#)() () const

### Public Attributes

- size\_t [countwords\\_](#)
- void \*& [buffer\\_](#)

#### 10.11.1 Detailed Description

Definition at line 57 of file [ion\\_gsms\\_chunk\\_data.h](#).

#### 10.11.2 Constructor & Destructor Documentation

##### 10.11.2.1 Allocator()

```
Allocator::Allocator (
    size_t countwords,
    void *& buffer) [inline]
```

Definition at line 62 of file [ion\\_gsms\\_chunk\\_data.h](#).

#### 10.11.3 Member Function Documentation

##### 10.11.3.1 operator()()

```
template<typename WordType>
void Allocator::operator() () const [inline]
```

Definition at line 66 of file [ion\\_gsms\\_chunk\\_data.h](#).

## 10.11.4 Member Data Documentation

### 10.11.4.1 buffer\_

`void*& Allocator::buffer_`

Definition at line 60 of file [ion\\_gsms\\_chunk\\_data.h](#).

### 10.11.4.2 countwords\_

`size_t Allocator::countwords_`

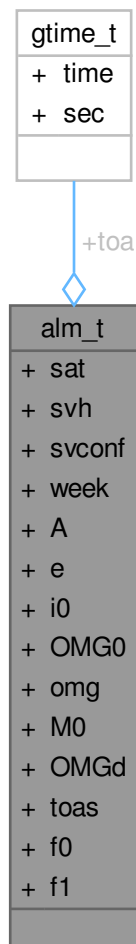
Definition at line 59 of file [ion\\_gsms\\_chunk\\_data.h](#).

The documentation for this struct was generated from the following file:

- [ion\\_gsms\\_chunk\\_data.h](#)

## 10.12 alm\_t Struct Reference

Collaboration diagram for `alm_t`:



## Public Attributes

- int [sat](#)
- int [svh](#)
- int [svconf](#)
- int [week](#)
- [gtime\\_t](#) [toa](#)
- double [A](#)
- double [e](#)
- double [i0](#)
- double [OMG0](#)
- double [omg](#)
- double [M0](#)
- double [OMGd](#)
- double [toas](#)
- double [f0](#)
- double [f1](#)

### 10.12.1 Detailed Description

Definition at line [418](#) of file [rtklib.h](#).

### 10.12.2 Member Data Documentation

#### 10.12.2.1 A

`double alm_t::A`

Definition at line [426](#) of file [rtklib.h](#).

#### 10.12.2.2 e

`double alm_t::e`

Definition at line [426](#) of file [rtklib.h](#).

#### 10.12.2.3 f0

`double alm_t::f0`

Definition at line [428](#) of file [rtklib.h](#).

#### 10.12.2.4 f1

`double alm_t::f1`

Definition at line [428](#) of file [rtklib.h](#).

#### 10.12.2.5 i0

`double alm_t::i0`

Definition at line [426](#) of file [rtklib.h](#).

#### 10.12.2.6 M0

`double alm_t::M0`

Definition at line [426](#) of file [rtklib.h](#).

#### 10.12.2.7 omg

`double alm_t::omg`

Definition at line [426](#) of file [rtklib.h](#).



#### 10.12.2.8 OMG0

```
double alm_t::OMG0
```

Definition at line 426 of file [rtklib.h](#).

#### 10.12.2.9 OMGd

```
double alm_t::OMGd
```

Definition at line 426 of file [rtklib.h](#).

#### 10.12.2.10 sat

```
int alm_t::sat
```

Definition at line 420 of file [rtklib.h](#).

#### 10.12.2.11 svconf

```
int alm_t::svconf
```

Definition at line 422 of file [rtklib.h](#).

#### 10.12.2.12 svh

```
int alm_t::svh
```

Definition at line 421 of file [rtklib.h](#).

#### 10.12.2.13 toa

```
gtime_t alm_t::toa
```

Definition at line 424 of file [rtklib.h](#).

#### 10.12.2.14 toas

```
double alm_t::toas
```

Definition at line 427 of file [rtklib.h](#).

#### 10.12.2.15 week

```
int alm_t::week
```

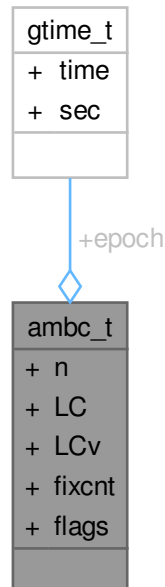
Definition at line 423 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.13 ambc\_t Struct Reference

Collaboration diagram for ambc\_t:



### Public Attributes

- [gtime\\_t epoch](#) [4]
- int [n](#) [4]
- double [LC](#) [4]
- double [LCv](#) [4]
- int [fixcnt](#)
- char [flags](#) [MAXSAT]

### 10.13.1 Detailed Description

Definition at line [1055](#) of file [rtklib.h](#).

### 10.13.2 Member Data Documentation

#### 10.13.2.1 epoch

[gtime\\_t](#) [ambc\\_t::epoch](#)[4]

Definition at line [1057](#) of file [rtklib.h](#).

#### 10.13.2.2 fixcnt

int [ambc\\_t::fixcnt](#)

Definition at line [1061](#) of file [rtklib.h](#).

#### 10.13.2.3 flags

char [ambc\\_t::flags](#)[MAXSAT]

Definition at line [1062](#) of file [rtklib.h](#).

### 10.13.2.4 LC

```
double ambc_t::LC[4]
```

Definition at line 1059 of file [rtklib.h](#).

### 10.13.2.5 LCv

```
double ambc_t::LCv[4]
```

Definition at line 1060 of file [rtklib.h](#).

### 10.13.2.6 n

```
int ambc_t::n[4]
```

Definition at line 1058 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

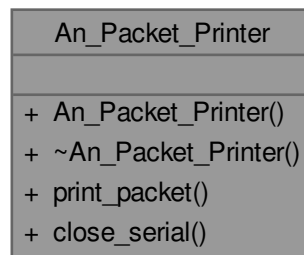
- [rtklib.h](#)

## 10.14 An\_Packet\_Printer Class Reference

class that prints PVT solutions in a serial device following a custom version of the Advanced Navigation Packet Protocol.

```
#include <an_packet_printer.h>
```

Collaboration diagram for An\_Packet\_Printer:



### Public Member Functions

- [An\\_Packet\\_Printer](#) (const std::string &an\_dump\_devname)  
*Default constructor.*
- [~An\\_Packet\\_Printer](#) ()  
*Default destructor.*
- bool [print\\_packet](#) (const [Rtklib\\_Solver](#) \*const pvt\_data, const std::map< int, [Gnss\\_Synchro](#) > &gnss\_observables\_map)  
*Print AN packet to the initialized device.*
- void [close\\_serial](#) () const  
*Close serial port. Also done in the destructor, this is only for testing.*

### 10.14.1 Detailed Description

class that prints PVT solutions in a serial device following a custom version of the Advanced Navigation Packet Protocol.

Definition at line 73 of file [an\\_packet\\_printer.h](#).

## 10.14.2 Constructor & Destructor Documentation

### 10.14.2.1 An\_Packet\_Printer()

```
An_Packet_Printer::An_Packet_Printer (
    const std::string & an_dump_devname) [explicit]
```

Default constructor.

### 10.14.2.2 ~An\_Packet\_Printer()

```
An_Packet_Printer::~~An_Packet_Printer ()
```

Default destructor.

## 10.14.3 Member Function Documentation

### 10.14.3.1 close\_serial()

```
void An_Packet_Printer::close_serial () const
```

Close serial port. Also done in the destructor, this is only for testing.

### 10.14.3.2 print\_packet()

```
bool An_Packet_Printer::print_packet (
    const Rtklib_Solver *const pvt_data,
    const std::map< int, Gnss_Synchro > & gnss_observables_map)
```

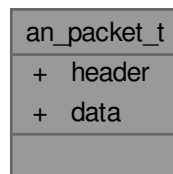
Print AN packet to the initialized device.

The documentation for this class was generated from the following file:

- [an\\_packet\\_printer.h](#)

## 10.15 an\_packet\_t Struct Reference

Collaboration diagram for an\_packet\_t:



### Public Attributes

- uint8\_t [header](#) [4]
- uint8\_t [data](#) [73]

### 10.15.1 Detailed Description

Definition at line 62 of file [an\\_packet\\_printer.h](#).

## 10.15.2 Member Data Documentation

### 10.15.2.1 data

`uint8_t an_packet_t::data[73]`

Definition at line 65 of file [an\\_packet\\_printer.h](#).

### 10.15.2.2 header

`uint8_t an_packet_t::header[4]`

Definition at line 64 of file [an\\_packet\\_printer.h](#).

The documentation for this struct was generated from the following file:

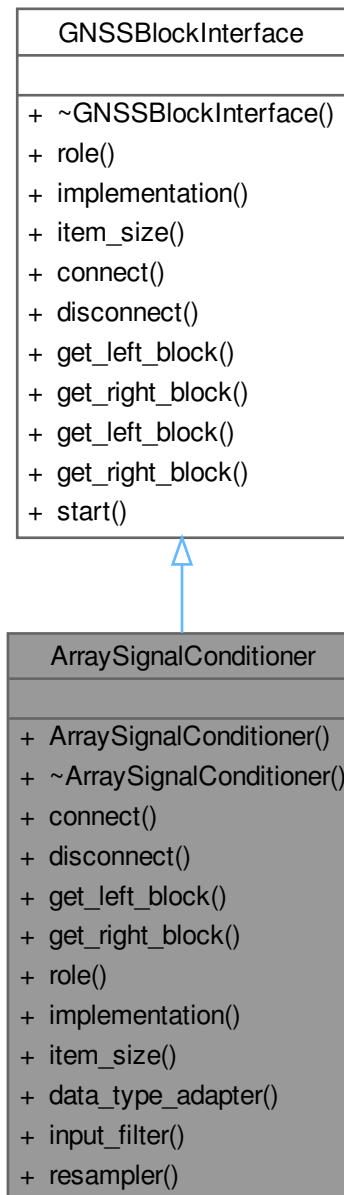
- [an\\_packet\\_printer.h](#)

## 10.16 ArraySignalConditioner Class Reference

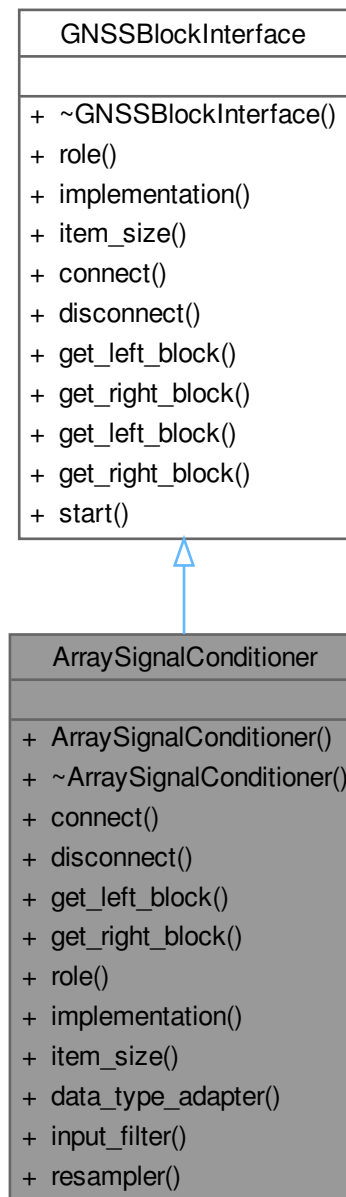
This class wraps blocks to change `data_type_adapter`, `input_filter` and `resampler` to be applied to the input flow of sampled signal.

```
#include <array_signal_conditioner.h>
```

Inheritance diagram for ArraySignalConditioner:



Collaboration diagram for ArraySignalConditioner:



### Public Member Functions

- `ArraySignalConditioner` (`std::shared_ptr< GNSSBlockInterface > data_type_adapt`, `std::shared_ptr< GNSSBlockInterface > in_filt`, `std::shared_ptr< GNSSBlockInterface > res`, `std::string role`)  
*Constructor.*
- `~ArraySignalConditioner` ()=default  
*Destructor.*
- void `connect` (`gr::top_block_sptr top_block`) override
- void `disconnect` (`gr::top_block_sptr top_block`) override
- `gr::basic_block_sptr get_left_block` () override

- `gr::basic_block_sptr` [get\\_right\\_block](#) () override
- `std::string` [role](#) () override
- `std::string` [implementation](#) () override  
*Returns "Array\_Signal\_Conditioner".*
- `size_t` [item\\_size](#) () override
- `std::shared_ptr< GNSSBlockInterface >` [data\\_type\\_adapter](#) ()
- `std::shared_ptr< GNSSBlockInterface >` [input\\_filter](#) ()
- `std::shared_ptr< GNSSBlockInterface >` [resampler](#) ()

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual `gr::basic_block_sptr` [get\\_left\\_block](#) (int RF\_channel)
- virtual `gr::basic_block_sptr` [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.16.1 Detailed Description

This class wraps blocks to change `data_type_adapter`, `input_filter` and `resampler` to be applied to the input flow of sampled signal.

Definition at line 41 of file [array\\_signal\\_conditioner.h](#).

### 10.16.2 Constructor & Destructor Documentation

#### 10.16.2.1 [ArraySignalConditioner\(\)](#)

```
ArraySignalConditioner::ArraySignalConditioner (
    std::shared_ptr< GNSSBlockInterface > data_type_adapt,
    std::shared_ptr< GNSSBlockInterface > in_filt,
    std::shared_ptr< GNSSBlockInterface > res,
    std::string role)
```

Constructor.

#### 10.16.2.2 [~ArraySignalConditioner\(\)](#)

```
ArraySignalConditioner::~~ArraySignalConditioner () [default]
```

Destructor.

### 10.16.3 Member Function Documentation

#### 10.16.3.1 [connect\(\)](#)

```
void ArraySignalConditioner::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.16.3.2 [data\\_type\\_adapter\(\)](#)

```
std::shared_ptr< GNSSBlockInterface > ArraySignalConditioner::data_type_adapter () [inline]
```

Definition at line 63 of file [array\\_signal\\_conditioner.h](#).

#### 10.16.3.3 [disconnect\(\)](#)

```
void ArraySignalConditioner::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).



**10.16.3.4 get\_left\_block()**

gr::basic\_block\_sptr ArraySignalConditioner::get\_left\_block () [override], [virtual]  
 Implements [GNSSBlockInterface](#).

**10.16.3.5 get\_right\_block()**

gr::basic\_block\_sptr ArraySignalConditioner::get\_right\_block () [override], [virtual]  
 Implements [GNSSBlockInterface](#).

**10.16.3.6 implementation()**

std::string ArraySignalConditioner::implementation () [inline], [override], [virtual]  
 Returns "Array\_Signal\_Conditioner".  
 Implements [GNSSBlockInterface](#).  
 Definition at line 60 of file [array\\_signal\\_conditioner.h](#).

**10.16.3.7 input\_filter()**

std::shared\_ptr< [GNSSBlockInterface](#) > ArraySignalConditioner::input\_filter () [inline]  
 Definition at line 64 of file [array\\_signal\\_conditioner.h](#).

**10.16.3.8 item\_size()**

size\_t ArraySignalConditioner::item\_size () [inline], [override], [virtual]  
 Implements [GNSSBlockInterface](#).  
 Definition at line 61 of file [array\\_signal\\_conditioner.h](#).

**10.16.3.9 resampler()**

std::shared\_ptr< [GNSSBlockInterface](#) > ArraySignalConditioner::resampler () [inline]  
 Definition at line 65 of file [array\\_signal\\_conditioner.h](#).

**10.16.3.10 role()**

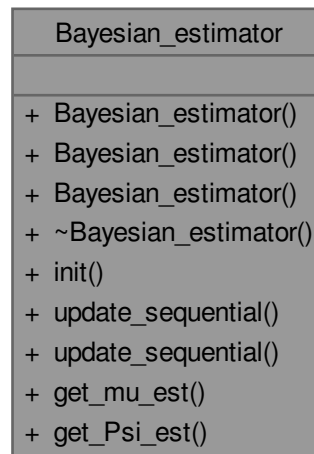
std::string ArraySignalConditioner::role () [inline], [override], [virtual]  
 Implements [GNSSBlockInterface](#).  
 Definition at line 58 of file [array\\_signal\\_conditioner.h](#).  
 The documentation for this class was generated from the following file:

- [array\\_signal\\_conditioner.h](#)

**10.17 Bayesian\_estimator Class Reference**

[Bayesian\\_estimator](#) is an estimator of noise characteristics (i.e. mean, covariance)  
 #include <bayesian\_estimation.h>

Collaboration diagram for Bayesian\_estimator:



### Public Member Functions

- **Bayesian\_estimator** (int ny)
- **Bayesian\_estimator** (const arma::vec &mu\_prior\_0, int kappa\_prior\_0, int nu\_prior\_0, const arma::mat &Psi\_prior\_0)
- void **init** (const arma::mat &mu\_prior\_0, int kappa\_prior\_0, int nu\_prior\_0, const arma::mat &Psi\_prior\_0)
- void **update\_sequential** (const arma::vec &data)
- void **update\_sequential** (const arma::vec &data, const arma::vec &mu\_prior\_0, int kappa\_prior\_0, int nu\_prior\_0, const arma::mat &Psi\_prior\_0)
- arma::mat **get\_mu\_est** () const
- arma::mat **get\_Psi\_est** () const

### 10.17.1 Detailed Description

[Bayesian\\_estimator](#) is an estimator of noise characteristics (i.e. mean, covariance)

[Bayesian\\_estimator](#) is an estimator which performs estimation of noise characteristics from a sequence of identically and independently distributed (IID) samples of a stationary stochastic process by way of Bayesian inference using conjugate priors. The posterior distribution is assumed to be Gaussian with mean  $\mathbf{\mu}$  and covariance  $\hat{\mathbf{C}}$ , which has a conjugate prior given by a normal-inverse-Wishart distribution with parameters  $\mathbf{\mu}_0$ ,  $\kappa_0$ ,  $\nu_0$ , and  $\mathbf{\Psi}$ .

[1] TODO: Ref1

Definition at line 60 of file [bayesian\\_estimation.h](#).

The documentation for this class was generated from the following file:

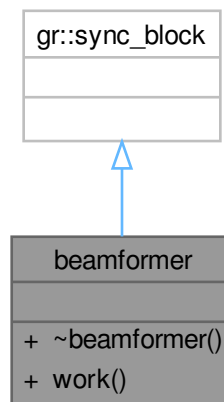
- [bayesian\\_estimation.h](#)

## 10.18 beamformer Class Reference

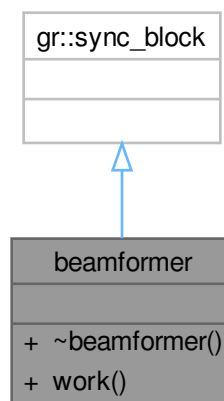
This class implements a real-time software-defined spatial filter using the CTTC GNSS experimental antenna array input and a set of dynamically reloadable weights.

```
#include <beamformer.h>
```

Inheritance diagram for beamformer:



Collaboration diagram for beamformer:



### Public Member Functions

- `int` **work** (`int` noutput\_items, `gr_vector_const_void_star` &input\_items, `gr_vector_void_star` &output\_items)

### Friends

- `beamformer_sptr` **make\_beamformer\_sptr** ()

### 10.18.1 Detailed Description

This class implements a real-time software-defined spatial filter using the CTTC GNSS experimental antenna array input and a set of dynamically reloadable weights.

Definition at line 42 of file [beamformer.h](#).

The documentation for this class was generated from the following file:

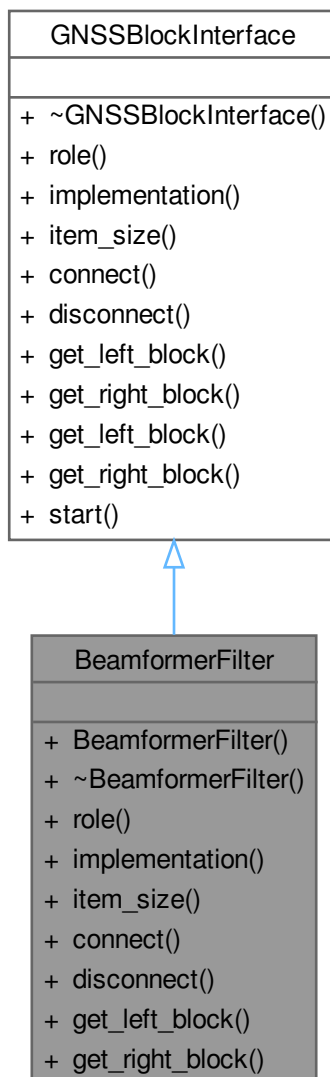
- [beamformer.h](#)

## 10.19 BeamformerFilter Class Reference

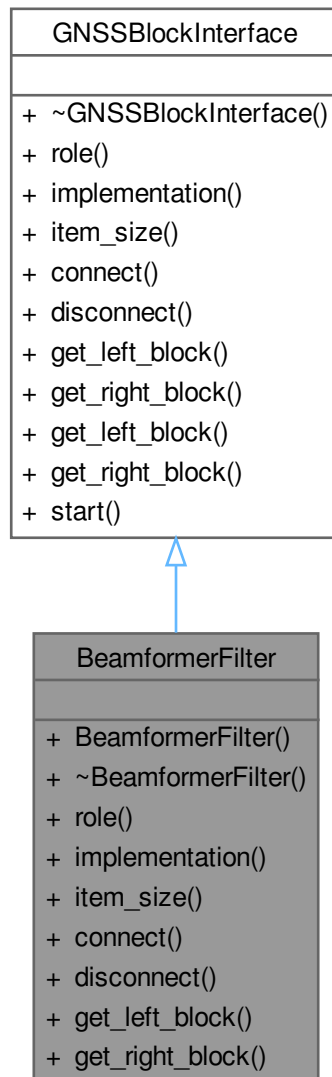
Interface of an adapter of a digital beamformer block to a [GNSSBlockInterface](#).

```
#include <beamformer_filter.h>
```

Inheritance diagram for BeamformerFilter:



Collaboration diagram for BeamformerFilter:



### Public Member Functions

- **BeamformerFilter** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_stream, unsigned int out\_stream)
- std::string [role](#) () override
- std::string [implementation](#) () override  
*returns "Beamformer\_Filte"*
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual `gr::basic_block_sptr` [get\\_left\\_block](#) (int RF\_channel)
- virtual `gr::basic_block_sptr` [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.19.1 Detailed Description

Interface of an adapter of a digital beamformer block to a [GNSSBlockInterface](#).  
Definition at line 38 of file [beamformer\\_filter.h](#).

### 10.19.2 Member Function Documentation

#### 10.19.2.1 `connect()`

```
void BeamformerFilter::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.19.2.2 `disconnect()`

```
void BeamformerFilter::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.19.2.3 `get_left_block()`

```
gr::basic_block_sptr BeamformerFilter::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.19.2.4 `get_right_block()`

```
gr::basic_block_sptr BeamformerFilter::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.19.2.5 `implementation()`

```
std::string BeamformerFilter::implementation () [inline], [override], [virtual]
```

returns "Beamformer\_Filte"

Implements [GNSSBlockInterface](#).

Definition at line 53 of file [beamformer\\_filter.h](#).

#### 10.19.2.6 `item_size()`

```
size_t BeamformerFilter::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 58 of file [beamformer\\_filter.h](#).

#### 10.19.2.7 `role()`

```
std::string BeamformerFilter::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 47 of file [beamformer\\_filter.h](#).

The documentation for this class was generated from the following file:

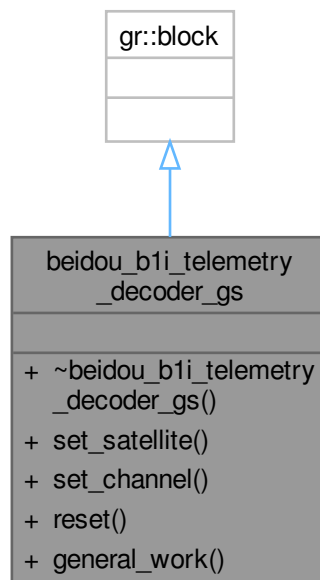
- [beamformer\\_filter.h](#)

## 10.20 beidou\_b1i\_telemetry\_decoder\_gs Class Reference

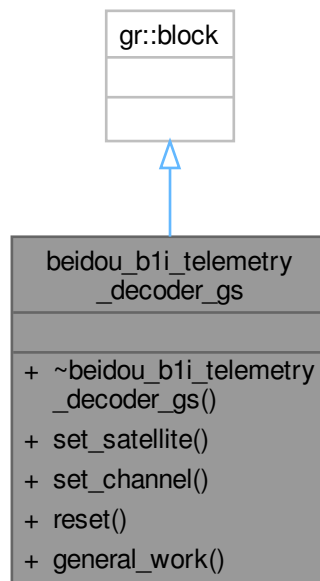
This class implements a block that decodes the BeiDou DNAV data.

```
#include <beidou_b1i_telemetry_decoder_gs.h>
```

Inheritance diagram for beidou\_b1i\_telemetry\_decoder\_gs:



Collaboration diagram for `beidou_b1i_telemetry_decoder_gs`:



### Public Member Functions

- `~beidou_b1i_telemetry_decoder_gs()` override  
*Class destructor.*
- void `set_satellite` (const `Gnss_Satellite` &satellite)  
*Set satellite PRN.*
- void `set_channel` (int channel)  
*Set receiver's channel.*
- void `reset` ()
- int `general_work` (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items) override  
*This is where all signal processing takes place.*

### Friends

- `beidou_b1i_telemetry_decoder_gs_sptr beidou_b1i_make_telemetry_decoder_gs` (const `Gnss_Satellite` &satellite, const `Tlm_Conf` &conf)

### 10.20.1 Detailed Description

This class implements a block that decodes the BeiDou DNAV data.

#### Note

Code added as part of GSoC 2018 program

Definition at line 57 of file `beidou_b1i_telemetry_decoder_gs.h`.



## 10.20.2 Constructor & Destructor Documentation

### 10.20.2.1 ~beidou\_b1i\_telemetry\_decoder\_gs()

```
beidou_b1i_telemetry_decoder_gs::~beidou_b1i_telemetry_decoder_gs () [override]
```

Class destructor.

## 10.20.3 Member Function Documentation

### 10.20.3.1 general\_work()

```
int beidou_b1i_telemetry_decoder_gs::general_work (
    int noutput_items,
    gr_vector_int & ninput_items,
    gr_vector_const_void_star & input_items,
    gr_vector_void_star & output_items) [override]
```

This is where all signal processing takes place.

### 10.20.3.2 set\_channel()

```
void beidou_b1i_telemetry_decoder_gs::set_channel (
    int channel)
```

Set receiver's channel.

### 10.20.3.3 set\_satellite()

```
void beidou_b1i_telemetry_decoder_gs::set_satellite (
    const Gnss_Satellite & satellite)
```

Set satellite PRN.

The documentation for this class was generated from the following file:

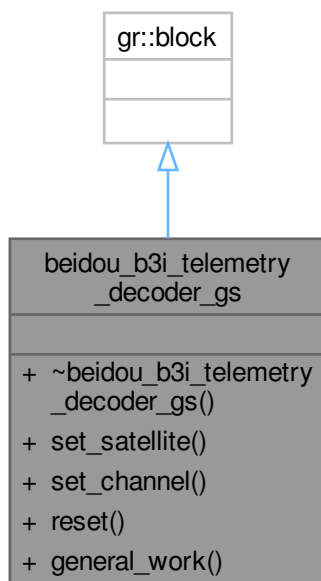
- [beidou\\_b1i\\_telemetry\\_decoder\\_gs.h](#)

## 10.21 beidou\_b3i\_telemetry\_decoder\_gs Class Reference

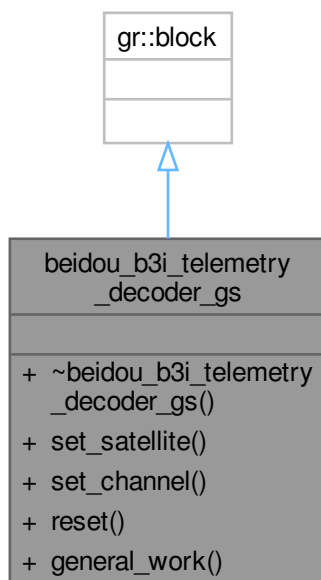
This class implements a block that decodes the BeiDou DNAV data.

```
#include <beidou_b3i_telemetry_decoder_gs.h>
```

Inheritance diagram for beidou\_b3i\_telemetry\_decoder\_gs:



Collaboration diagram for beidou\_b3i\_telemetry\_decoder\_gs:



## Public Member Functions

- [~beidou\\_b3i\\_telemetry\\_decoder\\_gs](#) () override  
*Class destructor.*
- void [set\\_satellite](#) (const [Gnss\\_Satellite](#) &satellite)  
*Set satellite PRN.*
- void [set\\_channel](#) (int channel)  
*Set receiver's channel.*
- void [reset](#) ()
- int [general\\_work](#) (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items) override  
*This is where all signal processing takes place.*

## Friends

- beidou\_b3i\_telemetry\_decoder\_gs\_sptr [beidou\\_b3i\\_make\\_telemetry\\_decoder\\_gs](#) (const [Gnss\\_Satellite](#) &satellite, const [Tlm\\_Conf](#) &conf)

### 10.21.1 Detailed Description

This class implements a block that decodes the BeiDou DNAV data.  
Definition at line 54 of file [beidou\\_b3i\\_telemetry\\_decoder\\_gs.h](#).

### 10.21.2 Constructor & Destructor Documentation

#### 10.21.2.1 ~beidou\_b3i\_telemetry\_decoder\_gs()

```
beidou_b3i_telemetry_decoder_gs::~beidou_b3i_telemetry_decoder_gs () [override]
```

Class destructor.

### 10.21.3 Member Function Documentation

#### 10.21.3.1 general\_work()

```
int beidou_b3i_telemetry_decoder_gs::general_work (
    int noutput_items,
    gr_vector_int & ninput_items,
    gr_vector_const_void_star & input_items,
    gr_vector_void_star & output_items) [override]
```

This is where all signal processing takes place.

#### 10.21.3.2 set\_channel()

```
void beidou_b3i_telemetry_decoder_gs::set_channel (
    int channel)
```

Set receiver's channel.

#### 10.21.3.3 set\_satellite()

```
void beidou_b3i_telemetry_decoder_gs::set_satellite (
    const Gnss\_Satellite & satellite)
```

Set satellite PRN.

The documentation for this class was generated from the following file:

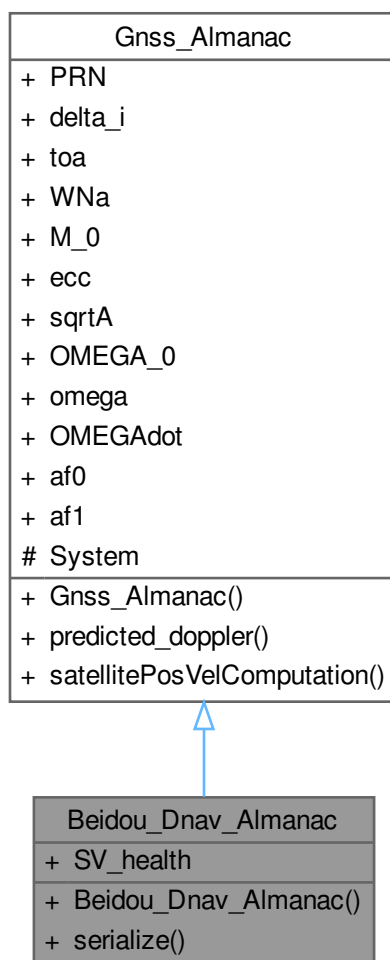
- [beidou\\_b3i\\_telemetry\\_decoder\\_gs.h](#)

## 10.22 Beidou\_Dnav\_Almanac Class Reference

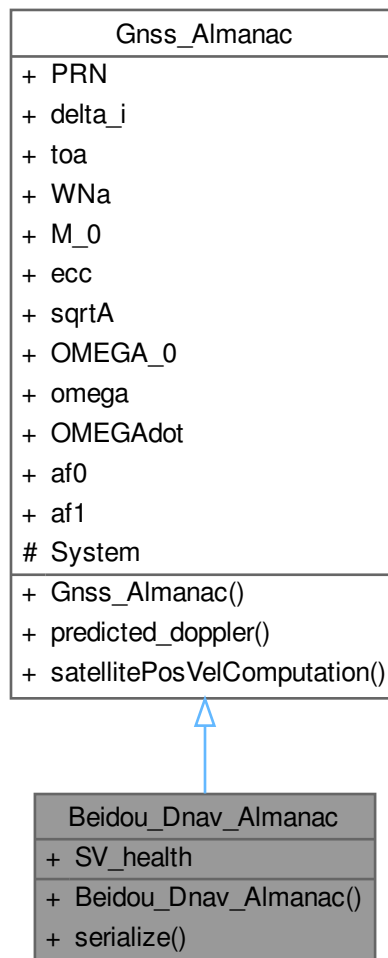
This class is a storage for the BeiDou D1 almanac.

```
#include <beidou_dnav_almanac.h>
```

Inheritance diagram for Beidou\_Dnav\_Almanac:



Collaboration diagram for Beidou\_Dnav\_Almanac:



### Public Member Functions

- [Beidou\\_Dnav\\_Almanac](#) ()
- `template<class Archive>`  
void [serialize](#) (Archive &ar, const unsigned int version)

### Public Member Functions inherited from [Gnss\\_Almanac](#)

- [Gnss\\_Almanac](#) ()=default
- double [predicted\\_doppler](#) (double rx\_time\_s, double lat, double lon, double h, double ve, double vn, double vu, int band) const  
*Computes prediction of the Doppler shift for a given time and receiver's position and velocity.*
- void [satellitePosVelComputation](#) (double transmitTime, std::array< double, 7 > &pos\_vel\_dtr) const  
*Computes satellite Position and Velocity, in ECEF, for a given time (expressed in seconds of week)*

### Public Attributes

- int [SV\\_health](#) {}

*SV Health.*

## Public Attributes inherited from [Gnss\\_Almanac](#)

- uint32\_t [PRN](#) {}  
*SV PRN NUMBER.*
- double [delta\\_i](#) {}  
*Inclination Angle at Reference Time (relative to  $i_0 = 0.30$  semi-circles)*
- int32\_t [toa](#) {}  
*Almanac data reference time of week [s].*
- int32\_t [WNa](#) {}  
*Almanac week number.*
- double [M\\_0](#) {}  
*Mean Anomaly at Reference Time [semi-circles].*
- double [ecc](#) {}  
*Eccentricity [dimensionless].*
- double [sqrtA](#) {}  
*Square Root of the Semi-Major Axis [sqrt(m)].*
- double [OMEGA\\_0](#) {}  
*Longitude of Ascending Node of Orbit Plane at Weekly Epoch [semi-circles].*
- double [omega](#) {}  
*Argument of Perigee [semi-circles].*
- double [OMEGAdot](#) {}  
*Rate of Right Ascension [semi-circles/s].*
- double [af0](#) {}  
*Coefficient 0 of code phase offset model [s].*
- double [af1](#) {}  
*Coefficient 1 of code phase offset model [s/s].*

## Additional Inherited Members

## Protected Attributes inherited from [Gnss\\_Almanac](#)

- char [System](#) {}  
*Character ID of the GNSS system. 'G': GPS. 'E': Galileo. 'B': BeiDou.*

### 10.22.1 Detailed Description

This class is a storage for the BeiDou D1 almanac.  
Definition at line 33 of file [beidou\\_dnav\\_almanac.h](#).

### 10.22.2 Constructor & Destructor Documentation

#### 10.22.2.1 Beidou\_Dnav\_Almanac()

```
Beidou_Dnav_Almanac::Beidou_Dnav_Almanac () [inline]
```

Default constructor

Definition at line 39 of file [beidou\\_dnav\\_almanac.h](#).

References [Gnss\\_Almanac::System](#).

## 10.22.3 Member Function Documentation

### 10.22.3.1 serialize()

```
template<class Archive>
void Beidou_Dnav_Almanac::serialize (
    Archive & ar,
    const unsigned int version) [inline]
```

Definition at line 48 of file [beidou\\_dnav\\_almanac.h](#).

## 10.22.4 Member Data Documentation

### 10.22.4.1 SV\_health

```
int Beidou_Dnav_Almanac::SV_health {}
```

SV Health.

Definition at line 44 of file [beidou\\_dnav\\_almanac.h](#).

The documentation for this class was generated from the following file:

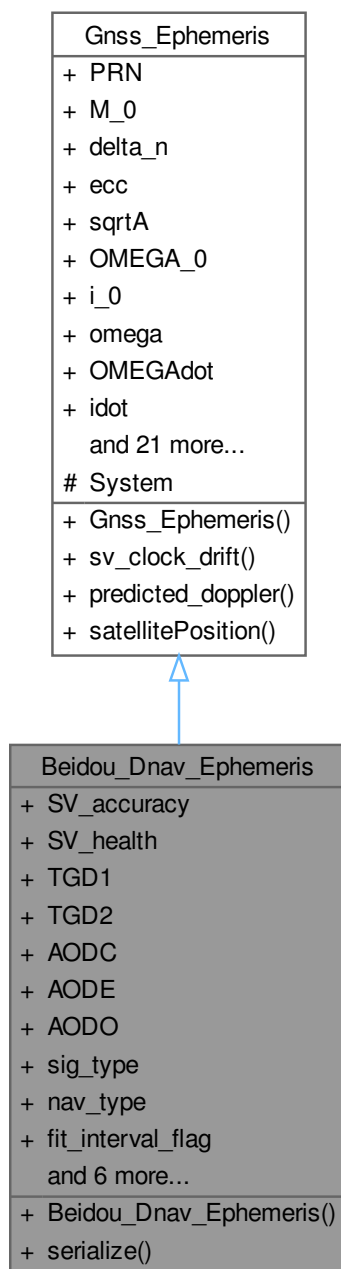
- [beidou\\_dnav\\_almanac.h](#)

## 10.23 Beidou\_Dnav\_Ephemeris Class Reference

This is a storage class for the Beidou SV ephemeris data as described in BeiDou Navigation Satellite System Signal In Space Interface Control Document Open Service Signal B1I (Version 3.0)

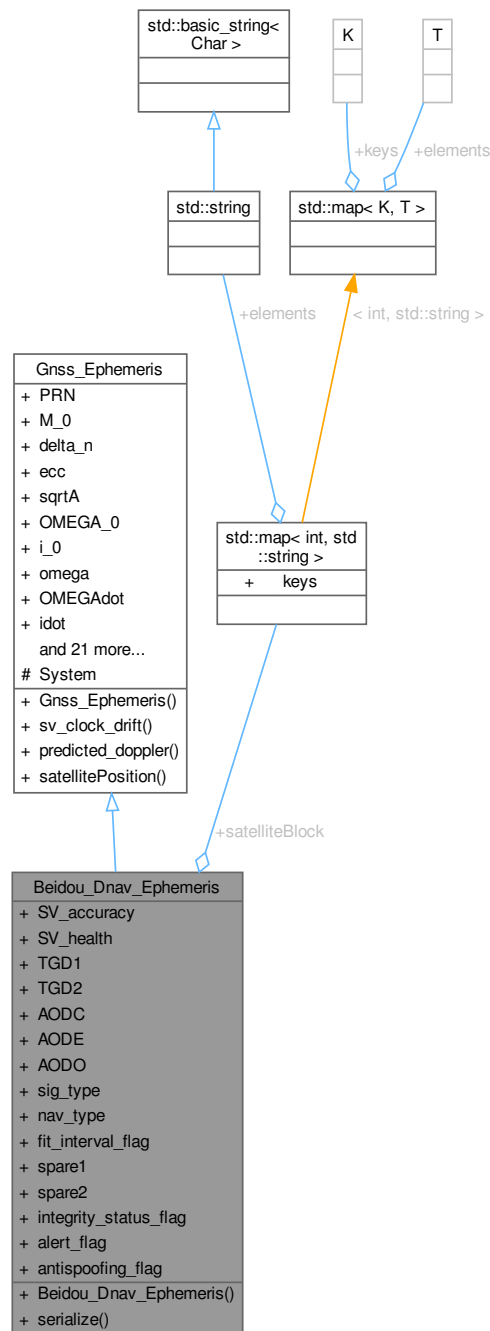
```
#include <beidou_dnav_ephemeris.h>
```

Inheritance diagram for Beidou\_Dnav\_Ephemeris:





Collaboration diagram for Beidou\_Dnav\_Ephemeris:



## Public Member Functions

- [Beidou\\_Dnav\\_Ephemeris](#) ()
- `template<class Archive>`  
void [serialize](#) (Archive &archive, const unsigned int version)

*Serialize is a boost standard method to be called by the boost XML serialization. Here is used to save the ephemeris data on disk file.*

## Public Member Functions inherited from Gnss\_Ephemeris

- double [sv\\_clock\\_drift](#) (double transmitTime)  
*Sets (satClkDrift) and (dtr), and returns the clock drift in seconds according to the User Algorithm for SV Clock Correction (IS-GPS-200M, 20.3.3.3.3.1, and Galileo OS SIS ICD, 5.1.4).*
- double [predicted\\_doppler](#) (double rx\_time\_s, double lat, double lon, double h, double ve, double vn, double vu, int band) const  
*Computes prediction of the Doppler shift for a given time and receiver's position and velocity.*
- void [satellitePosition](#) (double transmitTime)  
*Computes the ECEF SV coordinates and ECEF velocity.*

## Public Attributes

- int [SV\\_accuracy](#) {}  
*User Range Accuracy (URA) index of the SV (reference paragraph 5.2.4.5) for the standard positioning service user.*
- int [SV\\_health](#) {}
- double [TGD1](#) {}  
*Estimated Group Delay Differential on B1I [s].*
- double [TGD2](#) {}  
*Estimated Group Delay Differential on B2I [s].*
- double [AODC](#) {}  
*Age of Data, Clock.*
- double [AODE](#) {}  
*Age of Data, Ephemeris.*
- int [AODO](#) {}  
*Age of Data Offset (AODO) term for the navigation message correction table (NMCT) contained in subframe 4 (reference paragraph 20.3.3.5.1.9) [s].*
- int [sig\\_type](#) {}  
*BDS: data source (0:unknown,1:B1I,2:B1Q,3:B2I,4:B2Q,5:B3I,6:B3Q) \*/.*
- int [nav\\_type](#) {}  
*BDS: nav type (0:unknown,1:IGSO/MEO,2:GEO) \*/.*
- bool [fit\\_interval\\_flag](#) {}  
*Curve-fit interval used by the CS (Block II/IIA/IIR/IIR-M/IIF) and SS (Block IIIA) in determining the ephemeris parameters, as follows: 0 = 4 hours, 1 = greater than 4 hours.*
- double [spare1](#) {}
- double [spare2](#) {}
- bool [integrity\\_status\\_flag](#) {}  
*If true, enhanced level of integrity assurance.*
- bool [alert\\_flag](#) {}  
*If true, indicates that the SV URA may be worse than indicated in d\_SV\_accuracy, use that SV at our own risk.*
- bool [antispoofing\\_flag](#) {}  
*If true, the AntiSpoofing mode is ON in that SV.*
- std::map< int, std::string > [satelliteBlock](#)  
*Map that stores to which block the PRN belongs.*

## Public Attributes inherited from Gnss\_Ephemeris

- uint32\_t [PRN](#) {}  
*SV ID.*
- double [M\\_0](#) {}  
*Mean anomaly at reference time [rad].*
- double [delta\\_n](#) {}  
*Mean motion difference from computed value [rad/sec].*

- double `ecc` {}  
*Eccentricity.*
- double `sqrta` {}  
*Square root of the semi-major axis [meters<sup>1/2</sup>].*
- double `OMEGA_0` {}  
*Longitude of ascending node of orbital plane at weekly epoch [rad].*
- double `i_0` {}  
*Inclination angle at reference time [rad].*
- double `omega` {}  
*Argument of perigee [rad].*
- double `OMEGAdot` {}  
*Rate of right ascension [rad/sec].*
- double `idot` {}  
*Rate of inclination angle [rad/sec].*
- double `Cuc` {}  
*Amplitude of the cosine harmonic correction term to the argument of latitude [rad].*
- double `Cus` {}  
*Amplitude of the sine harmonic correction term to the argument of latitude [rad].*
- double `Crc` {}  
*Amplitude of the cosine harmonic correction term to the orbit radius [meters].*
- double `Crs` {}  
*Amplitude of the sine harmonic correction term to the orbit radius [meters].*
- double `Cic` {}  
*Amplitude of the cosine harmonic correction term to the angle of inclination [rad].*
- double `Cis` {}  
*Amplitude of the sine harmonic correction term to the angle of inclination [rad].*
- int32\_t `toe` {}  
*Ephemeris reference time [s].*
- int32\_t `toc` {}  
*Clock correction data reference Time of Week [sec].*
- double `af0` {}  
*SV clock bias correction coefficient [s].*
- double `af1` {}  
*SV clock drift correction coefficient [s/s].*
- double `af2` {}  
*SV clock drift rate correction coefficient [s/s<sup>2</sup>].*
- double `satClkDrift` {}  
*SV clock drift.*
- double `dtr` {}  
*Relativistic clock correction term.*
- int32\_t `WN` {}  
*Week number.*
- int32\_t `tow` {}  
*Time of Week.*
- double `satpos_X` {}  
*Earth-fixed coordinate x of the satellite [m]. Intersection of the IERS Reference Meridian (IRM) and the plane passing through the origin and normal to the Z-axis.*
- double `satpos_Y` {}  
*Earth-fixed coordinate y of the satellite [m]. Completes a right-handed, Earth-Centered, Earth-Fixed orthogonal coordinate system.*
- double `satpos_Z` {}

*Earth-fixed coordinate z of the satellite [m]. The direction of the IERS (International Earth Rotation and Reference Systems Service) Reference Pole (IRP).*

- double [satvel\\_X](#) {}

*Earth-fixed velocity coordinate x of the satellite [m].*

- double [satvel\\_Y](#) {}

*Earth-fixed velocity coordinate y of the satellite [m].*

- double [satvel\\_Z](#) {}

*Earth-fixed velocity coordinate z of the satellite [m].*

## Additional Inherited Members

## Protected Attributes inherited from [Gnss\\_Ephemeris](#)

- char [System](#) {}

*Character ID of the GNSS system. 'G': GPS. 'E': Galileo. 'B': BeiDou.*

## 10.23.1 Detailed Description

This is a storage class for the Beidou SV ephemeris data as described in BeiDou Navigation Satellite System Signal In Space Interface Control Document Open Service Signal B1I (Version 3.0)

See <http://en.beidou.gov.cn/SYSTEMS/Officialdocument/201902/P020190227601370045731.pdf>

Definition at line 39 of file [beidou\\_dnav\\_ephemeris.h](#).

## 10.23.2 Constructor & Destructor Documentation

### 10.23.2.1 Beidou\_Dnav\_Ephemeris()

```
Beidou_Dnav_Ephemeris::Beidou_Dnav_Ephemeris ()
```

Default constructor

## 10.23.3 Member Function Documentation

### 10.23.3.1 serialize()

```
template<class Archive>
void Beidou_Dnav_Ephemeris::serialize (
    Archive & archive,
    const unsigned int version) [inline]
```

Serialize is a boost standard method to be called by the boost XML serialization. Here is used to save the ephemeris data on disk file.

Definition at line 87 of file [beidou\\_dnav\\_ephemeris.h](#).

References [Gnss\\_Ephemeris::af0](#), [Gnss\\_Ephemeris::af1](#), [Gnss\\_Ephemeris::af2](#), [alert\\_flag](#), [antispoofing\\_flag](#), [AODC](#), [AODE](#), [AODO](#), [Gnss\\_Ephemeris::Cic](#), [Gnss\\_Ephemeris::Cis](#), [Gnss\\_Ephemeris::Crc](#), [Gnss\\_Ephemeris::Crs](#), [Gnss\\_Ephemeris::Cuc](#), [Gnss\\_Ephemeris::Cus](#), [Gnss\\_Ephemeris::delta\\_n](#), [Gnss\\_Ephemeris::dtr](#), [Gnss\\_Ephemeris::ecc](#), [fit\\_interval\\_flag](#), [Gnss\\_Ephemeris::i\\_0](#), [Gnss\\_Ephemeris::idot](#), [integrity\\_status\\_flag](#), [Gnss\\_Ephemeris::M\\_0](#), [nav\\_type](#), [Gnss\\_Ephemeris::omega](#), [Gnss\\_Ephemeris::OMEGA\\_0](#), [Gnss\\_Ephemeris::OMEGAdot](#), [Gnss\\_Ephemeris::PRN](#), [Gnss\\_Ephemeris::satClkDrift](#), [sig\\_type](#), [Gnss\\_Ephemeris::sqrtA](#), [SV\\_accuracy](#), [TGD1](#), [TGD2](#), [Gnss\\_Ephemeris::toc](#), [Gnss\\_Ephemeris::toe](#), [Gnss\\_Ephemeris::tow](#), and [Gnss\\_Ephemeris::WN](#).

## 10.23.4 Member Data Documentation

### 10.23.4.1 alert\_flag

```
bool Beidou_Dnav_Ephemeris::alert_flag {}
```

If true, indicates that the SV URA may be worse than indicated in [d\\_SV\\_accuracy](#), use that SV at our own risk.

Definition at line 76 of file [beidou\\_dnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.23.4.2 antispoofing\_flag

```
bool Beidou_Dnav_Ephemeris::antispoofing_flag {}
```

If true, the AntiSpoofing mode is ON in that SV.

Definition at line 77 of file [beidou\\_dnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.23.4.3 AODC

```
double Beidou_Dnav_Ephemeris::AODC {}
```

Age of Data, Clock.

Definition at line 51 of file [beidou\\_dnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.23.4.4 AODE

```
double Beidou_Dnav_Ephemeris::AODE {}
```

Age of Data, Ephemeris.

Definition at line 52 of file [beidou\\_dnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.23.4.5 AODO

```
int Beidou_Dnav_Ephemeris::AODO {}
```

Age of Data Offset (AODO) term for the navigation message correction table (NMCT) contained in subframe 4 (reference paragraph 20.3.3.5.1.9) [s].

Definition at line 53 of file [beidou\\_dnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.23.4.6 fit\_interval\_flag

```
bool Beidou_Dnav_Ephemeris::fit_interval_flag {}
```

Curve-fit interval used by the CS (Block II/IIA/IIR/IIR-M/IIF) and SS (Block IIIA) in determining the ephemeris parameters, as follows: 0 = 4 hours, 1 = greater than 4 hours.

Definition at line 58 of file [beidou\\_dnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.23.4.7 integrity\_status\_flag

```
bool Beidou_Dnav_Ephemeris::integrity_status_flag {}
```

If true, enhanced level of integrity assurance.

If false, indicates that the conveying signal is provided with the legacy level of integrity assurance. That is, the probability that the instantaneous URE of the conveying signal exceeds 4.42 times the upper bound value of the current broadcast URA index, for more than 5.2 seconds, without an accompanying alert, is less than 1E-5 per hour.

If true, indicates that the conveying signal is provided with an enhanced level of integrity assurance. That is, the probability that the instantaneous URE of the conveying signal exceeds 5.73 times the upper bound value of the current broadcast URA index, for more than 5.2 seconds, without an accompanying alert, is less than 1E-8 per hour.

Definition at line 75 of file [beidou\\_dnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.23.4.8 nav\_type

```
int Beidou_Dnav_Ephemeris::nav_type {}
```

BDS: nav type (0:unknown,1:IGSO/MEO,2:GEO) \*/.

Definition at line 56 of file [beidou\\_dnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.23.4.9 satelliteBlock

```
std::map<int, std::string> Beidou_Dnav_Ephemeris::satelliteBlock
```

Map that stores to which block the PRN belongs.

Definition at line 79 of file [beidou\\_dnav\\_ephemeris.h](#).

#### 10.23.4.10 sig\_type

```
int Beidou_Dnav_Ephemeris::sig_type {}
```

BDS: data source (0:unknown,1:B1I,2:B1Q,3:B2I,4:B2Q,5:B3I,6:B3Q) \*/.

Definition at line 55 of file [beidou\\_dnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.23.4.11 spare1

```
double Beidou_Dnav_Ephemeris::spare1 {}
```

Definition at line 59 of file [beidou\\_dnav\\_ephemeris.h](#).

#### 10.23.4.12 spare2

```
double Beidou_Dnav_Ephemeris::spare2 {}
```

Definition at line 60 of file [beidou\\_dnav\\_ephemeris.h](#).

#### 10.23.4.13 SV\_accuracy

```
int Beidou_Dnav_Ephemeris::SV_accuracy {}
```

User Range Accuracy (URA) index of the SV (reference paragraph 5.2.4.5) for the standard positioning service user.

Definition at line 47 of file [beidou\\_dnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.23.4.14 SV\_health

```
int Beidou_Dnav_Ephemeris::SV_health {}
```

Definition at line 48 of file [beidou\\_dnav\\_ephemeris.h](#).

#### 10.23.4.15 TGD1

```
double Beidou_Dnav_Ephemeris::TGD1 {}
```

Estimated Group Delay Differential on B1I [s].

Definition at line 49 of file [beidou\\_dnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.23.4.16 TGD2

```
double Beidou_Dnav_Ephemeris::TGD2 {}
```

Estimated Group Delay Differential on B2I [s].

Definition at line 50 of file [beidou\\_dnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

The documentation for this class was generated from the following file:

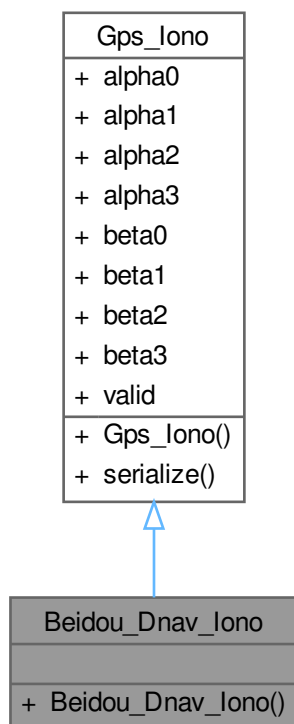
- [beidou\\_dnav\\_ephemeris.h](#)

## 10.24 Beidou\_Dnav\_Iono Class Reference

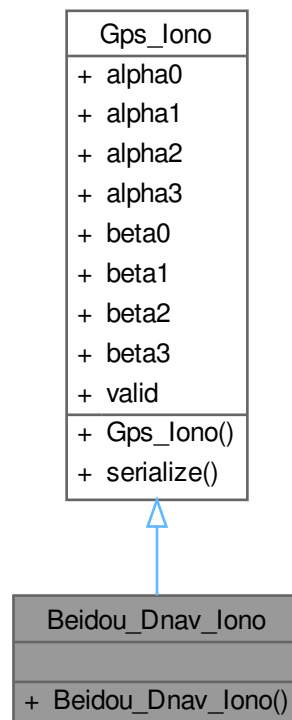
This class is a storage for the BEIDOU IONOSPHERIC data as described in ICD v2.1.

```
#include <beidou_dnav_iono.h>
```

Inheritance diagram for Beidou\_Dnav\_Iono:



Collaboration diagram for Beidou\_Dnav\_Iono:



### Public Member Functions

- [Beidou\\_Dnav\\_Iono](#) ()=default  
*Default constructor.*

### Public Member Functions inherited from [Gps\\_Iono](#)

- [Gps\\_Iono](#) ()=default  
*Default constructor.*
- `template<class Archive>`  
void [serialize](#) (Archive &archive, const unsigned int version)  
*Serialize is a boost standard method to be called by the boost XML serialization. Here is used to save the ephemeris data on disk file.*

### Additional Inherited Members

### Public Attributes inherited from [Gps\\_Iono](#)

- double [alpha0](#) {}  
*Coefficient 0 of a cubic equation representing the amplitude of the vertical delay [s].*
- double [alpha1](#) {}  
*Coefficient 1 of a cubic equation representing the amplitude of the vertical delay [s/semi-circle].*
- double [alpha2](#) {}  
*Coefficient 2 of a cubic equation representing the amplitude of the vertical delay [s(semi-circle)<sup>2</sup>].*



- double [alpha3](#) {}  
*Coefficient 3 of a cubic equation representing the amplitude of the vertical delay  $[s(\text{semi-circle})^3]$ .*
- double [beta0](#) {}  
*Coefficient 0 of a cubic equation representing the period of the model [s].*
- double [beta1](#) {}  
*Coefficient 1 of a cubic equation representing the period of the model  $[s/\text{semi-circle}]$ .*
- double [beta2](#) {}  
*Coefficient 2 of a cubic equation representing the period of the model  $[s(\text{semi-circle})^2]$ .*
- double [beta3](#) {}  
*Coefficient 3 of a cubic equation representing the period of the model  $[s(\text{semi-circle})^3]$ .*
- bool [valid](#) {}  
*Valid flag.*

### 10.24.1 Detailed Description

This class is a storage for the BEIDOU IONOSPHERIC data as described in ICD v2.1.  
Definition at line 33 of file [beidou\\_dnav\\_iono.h](#).

### 10.24.2 Constructor & Destructor Documentation

#### 10.24.2.1 Beidou\_Dnav\_Iono()

```
Beidou_Dnav_Iono::Beidou_Dnav_Iono () [default]
```

Default constructor.

The documentation for this class was generated from the following file:

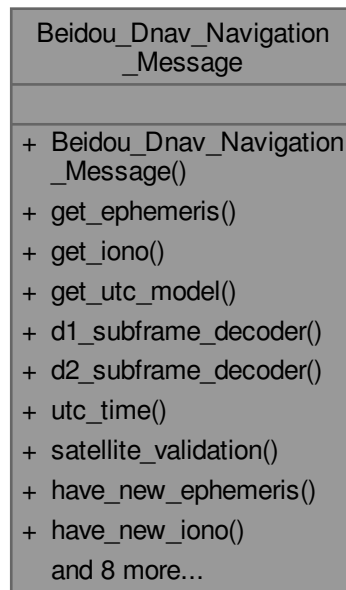
- [beidou\\_dnav\\_iono.h](#)

## 10.25 Beidou\_Dnav\_Navigation\_Message Class Reference

This class decodes a BeiDou D1 NAV Data message.

```
#include <beidou_dnav_navigation_message.h>
```

Collaboration diagram for Beidou\_Dnav\_Navigation\_Message:



## Public Member Functions

- [Beidou\\_Dnav\\_Navigation\\_Message](#) ()
- [Beidou\\_Dnav\\_Ephemeris](#) [get\\_ephemeris](#) () const  
*Obtain a BDS SV Ephemeris class filled with current SV data.*
- [Beidou\\_Dnav\\_Iono](#) [get\\_iono](#) ()  
*Obtain a BDS ionospheric correction parameters class filled with current SV data.*
- [Beidou\\_Dnav\\_Utc\\_Model](#) [get\\_utc\\_model](#) ()  
*Obtain a BDS UTC model parameters class filled with current SV data.*
- [int32\\_t](#) [d1\\_subframe\\_decoder](#) (std::string const &subframe)  
*Decodes the BDS D1 NAV message.*
- [int32\\_t](#) [d2\\_subframe\\_decoder](#) (std::string const &subframe)  
*Decodes the BDS D2 NAV message.*
- [double](#) [utc\\_time](#) (double beidoutime\_corrected) const  
*Computes the Coordinated Universal Time (UTC) and returns it in [s].*
- [bool](#) [satellite\\_validation](#) ()
- [bool](#) [have\\_new\\_ephemeris](#) ()  
*Returns true if new Ephemeris has arrived. The flag is set to false when the function is executed.*
- [bool](#) [have\\_new\\_iono](#) () const  
*Returns true if new Iono model has arrived. The flag is set to false when the function is executed.*
- [bool](#) [have\\_new\\_utc\\_model](#) ()  
*Returns true if new UTC model has arrived. The flag is set to false when the function is executed.*
- [bool](#) [have\\_new\\_almanac](#) ()  
*Returns true if new UTC model has arrived. The flag is set to false when the function is executed.*
- [void](#) [set\\_satellite\\_PRN](#) (uint32\_t prn)  
*Sets satellite PRN number.*

- void [set\\_signal\\_type](#) (int32\_t signal\_type)
- bool [get\\_flag\\_CRC\\_test](#) () const
- bool [get\\_flag\\_new\\_SOW\\_available](#) () const
- void [set\\_flag\\_new\\_SOW\\_available](#) (bool new\_SOW\_available)
- double [get\\_SOW](#) () const

### 10.25.1 Detailed Description

This class decodes a BeiDou D1 NAV Data message.

Definition at line 46 of file [beidou\\_dnav\\_navigation\\_message.h](#).

### 10.25.2 Constructor & Destructor Documentation

#### 10.25.2.1 Beidou\_Dnav\_Navigation\_Message()

```
Beidou_Dnav_Navigation_Message::Beidou_Dnav_Navigation_Message ()
```

Default constructor

### 10.25.3 Member Function Documentation

#### 10.25.3.1 d1\_subframe\_decoder()

```
int32_t Beidou_Dnav_Navigation_Message::d1_subframe_decoder (
    std::string const & subframe)
```

Decodes the BDS D1 NAV message.

#### 10.25.3.2 d2\_subframe\_decoder()

```
int32_t Beidou_Dnav_Navigation_Message::d2_subframe_decoder (
    std::string const & subframe)
```

Decodes the BDS D2 NAV message.

#### 10.25.3.3 get\_ephemeris()

```
Beidou_Dnav_Ephemeris Beidou_Dnav_Navigation_Message::get_ephemeris () const
```

Obtain a BDS SV Ephemeris class filled with current SV data.

#### 10.25.3.4 get\_flag\_CRC\_test()

```
bool Beidou_Dnav_Navigation_Message::get_flag_CRC_test () const [inline]
```

Definition at line 120 of file [beidou\\_dnav\\_navigation\\_message.h](#).

#### 10.25.3.5 get\_flag\_new\_SOW\_available()

```
bool Beidou_Dnav_Navigation_Message::get_flag_new_SOW_available () const [inline]
```

Definition at line 125 of file [beidou\\_dnav\\_navigation\\_message.h](#).

#### 10.25.3.6 get\_iono()

```
Beidou_Dnav_Iono Beidou_Dnav_Navigation_Message::get_iono ()
```

Obtain a BDS ionospheric correction parameters class filled with current SV data.

#### 10.25.3.7 get\_SOW()

```
double Beidou_Dnav_Navigation_Message::get_SOW () const [inline]
```

Definition at line 135 of file [beidou\\_dnav\\_navigation\\_message.h](#).

**10.25.3.8 get\_utc\_model()**

[Beidou\\_Dnav\\_Utc\\_Model](#) `Beidou_Dnav_Navigation_Message::get_utc_model ()`

Obtain a BDS UTC model parameters class filled with current SV data.

**10.25.3.9 have\_new\_almanac()**

`bool Beidou_Dnav_Navigation_Message::have_new_almanac ()`

Returns true if new UTC model has arrived. The flag is set to false when the function is executed.

**10.25.3.10 have\_new\_ephemeris()**

`bool Beidou_Dnav_Navigation_Message::have_new_ephemeris ()`

Returns true if new Ephemeris has arrived. The flag is set to false when the function is executed.

**10.25.3.11 have\_new\_iono()**

`bool Beidou_Dnav_Navigation_Message::have_new_iono () const`

Returns true if new Iono model has arrived. The flag is set to false when the function is executed.

**10.25.3.12 have\_new\_utc\_model()**

`bool Beidou_Dnav_Navigation_Message::have_new_utc_model ()`

Returns true if new UTC model has arrived. The flag is set to false when the function is executed.

**10.25.3.13 set\_flag\_new\_SOW\_available()**

`void Beidou_Dnav_Navigation_Message::set_flag_new_SOW_available (`  
     `bool new_SOW_available) [inline]`

Definition at line 130 of file [beidou\\_dnav\\_navigation\\_message.h](#).

**10.25.3.14 set\_satellite\_PRN()**

`void Beidou_Dnav_Navigation_Message::set_satellite_PRN (`  
     `uint32_t prn) [inline]`

Sets satellite PRN number.

Definition at line 110 of file [beidou\\_dnav\\_navigation\\_message.h](#).

**10.25.3.15 set\_signal\_type()**

`void Beidou_Dnav_Navigation_Message::set_signal_type (`  
     `int32_t signal_type) [inline]`

Definition at line 115 of file [beidou\\_dnav\\_navigation\\_message.h](#).

**10.25.3.16 utc\_time()**

`double Beidou_Dnav_Navigation_Message::utc_time (`  
     `double beidou_time_corrected) const`

Computes the Coordinated Universal Time (UTC) and returns it in [s].

The documentation for this class was generated from the following file:

- [beidou\\_dnav\\_navigation\\_message.h](#)

**10.26 Beidou\_Dnav\_Utc\_Model Class Reference**

This class is a storage for the BeiDou DNAV UTC Model.

`#include <beidou_dnav_utc_model.h>`

Collaboration diagram for Beidou\_Dnav\_Utc\_Model:

Beidou_Dnav_Utc_Model
+ A0_UTC
+ A1_UTC
+ DeltaT_LS
+ WN_LSF
+ DN
+ DeltaT_LSF
+ A0_GPS
+ A1_GPS
+ A0_GAL
+ A1_GAL
+ A0_GLO
+ A1_GLO
+ valid
+ Beidou_Dnav_Utc_Model()
+ serialize()

### Public Member Functions

- template<class Archive>  
void [serialize](#) (Archive &archive, const unsigned int version)

### Public Attributes

- double [A0\\_UTC](#) {}  
*BDT clock bias relative to UTC [s].*
- double [A1\\_UTC](#) {}  
*BDT clock rate relative to UTC [s/s].*
- int [DeltaT\\_LS](#) {}  
*Delta time due to leap seconds before the new leap second effective.*
- int [WN\\_LSF](#) {}  
*Week number of the new leap second.*
- int [DN](#) {}  
*Day number of week of the new leap second.*
- double [DeltaT\\_LSF](#) {}  
*Delta time due to leap seconds after the new leap second effective [s].*
- double [A0\\_GPS](#) {}  
*BDT clock bias relative to GPS time [s].*
- double [A1\\_GPS](#) {}  
*BDT clock rate relative to GPS time [s/s].*
- double [A0\\_GAL](#) {}  
*BDT clock bias relative to GAL time [s].*

- double [A1\\_GAL](#) {}  
*BDT clock rate relative to GAL time [s/s].*
- double [A0\\_GLO](#) {}  
*BDT clock bias relative to GLO time [s].*
- double [A1\\_GLO](#) {}  
*BDT clock rate relative to GLO time [s/s].*
- bool [valid](#) {}

### 10.26.1 Detailed Description

This class is a storage for the BeiDou DNAV UTC Model.  
Implementation follows the interface described in the Open Service Signal (Version 2.1)  
Definition at line 35 of file [beidou\\_dnav\\_utc\\_model.h](#).

### 10.26.2 Member Function Documentation

#### 10.26.2.1 `serialize()`

```
template<class Archive>
void Beidou_Dnav_Utc_Model::serialize (
    Archive & archive,
    const unsigned int version) [inline]
```

Definition at line 67 of file [beidou\\_dnav\\_utc\\_model.h](#).

### 10.26.3 Member Data Documentation

#### 10.26.3.1 `A0_GAL`

```
double Beidou_Dnav_Utc_Model::A0_GAL {}
```

BDT clock bias relative to GAL time [s].  
Definition at line 53 of file [beidou\\_dnav\\_utc\\_model.h](#).

#### 10.26.3.2 `A0_GLO`

```
double Beidou_Dnav_Utc_Model::A0_GLO {}
```

BDT clock bias relative to GLO time [s].  
Definition at line 57 of file [beidou\\_dnav\\_utc\\_model.h](#).

#### 10.26.3.3 `A0_GPS`

```
double Beidou_Dnav_Utc_Model::A0_GPS {}
```

BDT clock bias relative to GPS time [s].  
Definition at line 49 of file [beidou\\_dnav\\_utc\\_model.h](#).

#### 10.26.3.4 `A0_UTC`

```
double Beidou_Dnav_Utc_Model::A0_UTC {}
```

BDT clock bias relative to UTC [s].  
Definition at line 41 of file [beidou\\_dnav\\_utc\\_model.h](#).

#### 10.26.3.5 `A1_GAL`

```
double Beidou_Dnav_Utc_Model::A1_GAL {}
```

BDT clock rate relative to GAL time [s/s].  
Definition at line 54 of file [beidou\\_dnav\\_utc\\_model.h](#).

### 10.26.3.6 A1\_GLO

```
double Beidou_Dnav_Utc_Model::A1_GLO {}
```

BDT clock rate relative to GLO time [s/s].  
Definition at line 58 of file [beidou\\_dnav\\_utc\\_model.h](#).

### 10.26.3.7 A1\_GPS

```
double Beidou_Dnav_Utc_Model::A1_GPS {}
```

BDT clock rate relative to GPS time [s/s].  
Definition at line 50 of file [beidou\\_dnav\\_utc\\_model.h](#).

### 10.26.3.8 A1\_UTC

```
double Beidou_Dnav_Utc_Model::A1_UTC {}
```

BDT clock rate relative to UTC [s/s].  
Definition at line 42 of file [beidou\\_dnav\\_utc\\_model.h](#).

### 10.26.3.9 DeltaT\_LS

```
int Beidou_Dnav_Utc_Model::DeltaT_LS {}
```

Delta time due to leap seconds before the new leap second effective.  
Definition at line 43 of file [beidou\\_dnav\\_utc\\_model.h](#).

### 10.26.3.10 DeltaT\_LSF

```
double Beidou_Dnav_Utc_Model::DeltaT_LSF {}
```

Delta time due to leap seconds after the new leap second effective [s].  
Definition at line 46 of file [beidou\\_dnav\\_utc\\_model.h](#).

### 10.26.3.11 DN

```
int Beidou_Dnav_Utc_Model::DN {}
```

Day number of week of the new leap second.  
Definition at line 45 of file [beidou\\_dnav\\_utc\\_model.h](#).

### 10.26.3.12 valid

```
bool Beidou_Dnav_Utc_Model::valid {}
```

Definition at line 60 of file [beidou\\_dnav\\_utc\\_model.h](#).

### 10.26.3.13 WN\_LSF

```
int Beidou_Dnav_Utc_Model::WN_LSF {}
```

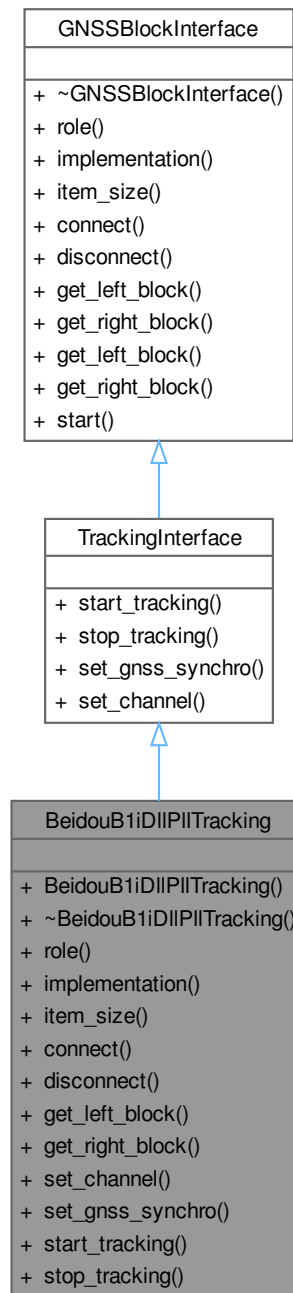
Week number of the new leap second.  
Definition at line 44 of file [beidou\\_dnav\\_utc\\_model.h](#).  
The documentation for this class was generated from the following file:

- [beidou\\_dnav\\_utc\\_model.h](#)

## 10.27 BeidouB1iDIPIITracking Class Reference

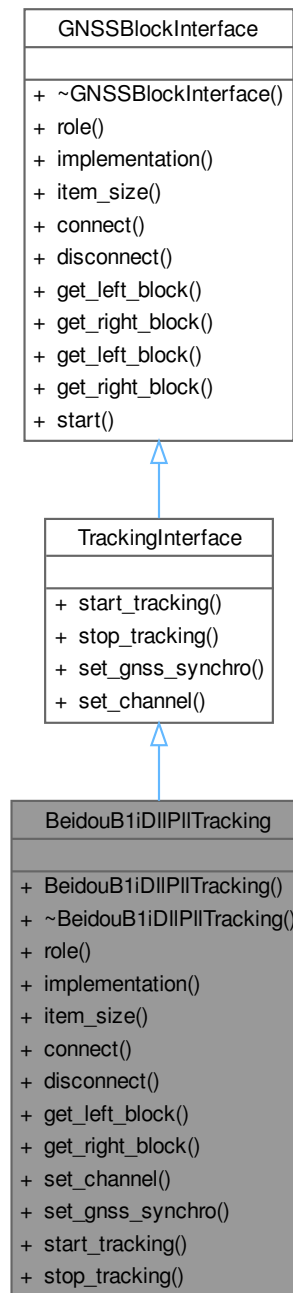
This class implements a code DLL + carrier PLL tracking loop.  
`#include <beidou_bli_dll_pll_tracking.h>`

Inheritance diagram for BeidouB1iDIIPITracking:





Collaboration diagram for BeidouB1iDIIPIITracking:



### Public Member Functions

- **BeidouB1iDIIPIITracking** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override

- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_channel](#) (unsigned int channel) override
 

*Set tracking channel unique ID.*
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override
 

*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [start\\_tracking](#) () override
- void [stop\\_tracking](#) () override
 

*Stop running tracking.*

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()
 

*Start the flow of samples if needed.*

### 10.27.1 Detailed Description

This class implements a code DLL + carrier PLL tracking loop.  
 Definition at line 41 of file [beidou\\_b1i\\_dll\\_pll\\_tracking.h](#).

### 10.27.2 Member Function Documentation

#### 10.27.2.1 connect()

```
void BeidouBliDllPllTracking::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.27.2.2 disconnect()

```
void BeidouBliDllPllTracking::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.27.2.3 get\_left\_block()

```
gr::basic_block_sptr BeidouBliDllPllTracking::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.27.2.4 get\_right\_block()

```
gr::basic_block_sptr BeidouBliDllPllTracking::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.27.2.5 implementation()

```
std::string BeidouBliDllPllTracking::implementation () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).  
 Definition at line 57 of file [beidou\\_b1i\\_dll\\_pll\\_tracking.h](#).

#### 10.27.2.6 item\_size()

```
size_t BeidouBliDllPllTracking::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).  
 Definition at line 62 of file [beidou\\_b1i\\_dll\\_pll\\_tracking.h](#).

**10.27.2.7 role()**

```
std::string BeidouBliDllPllTracking::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 52 of file [beidou\\_b1i\\_dll\\_pll\\_tracking.h](#).

**10.27.2.8 set\_channel()**

```
void BeidouBliDllPllTracking::set_channel (
    unsigned int channel) [override], [virtual]
```

Set tracking channel unique ID.

Implements [TrackingInterface](#).

**10.27.2.9 set\_gnss\_synchro()**

```
void BeidouBliDllPllTracking::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [TrackingInterface](#).

**10.27.2.10 start\_tracking()**

```
void BeidouBliDllPllTracking::start_tracking () [override], [virtual]
```

Implements [TrackingInterface](#).

**10.27.2.11 stop\_tracking()**

```
void BeidouBliDllPllTracking::stop_tracking () [override], [virtual]
```

Stop running tracking.

Implements [TrackingInterface](#).

The documentation for this class was generated from the following file:

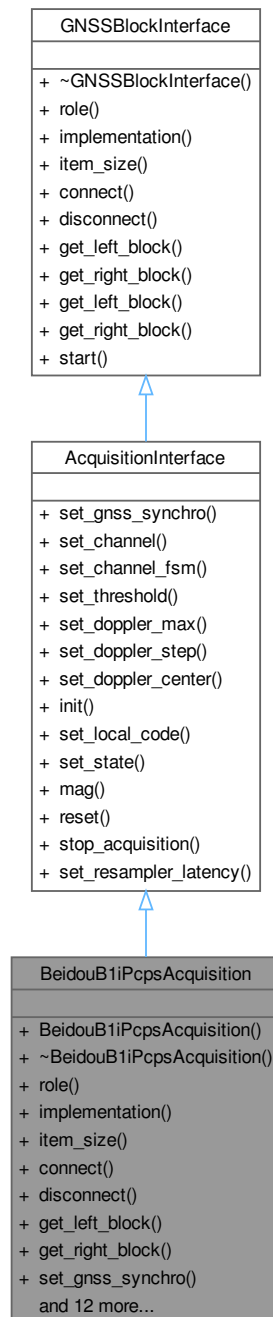
- [beidou\\_b1i\\_dll\\_pll\\_tracking.h](#)

**10.28 BeidouB1iPcpsAcquisition Class Reference**

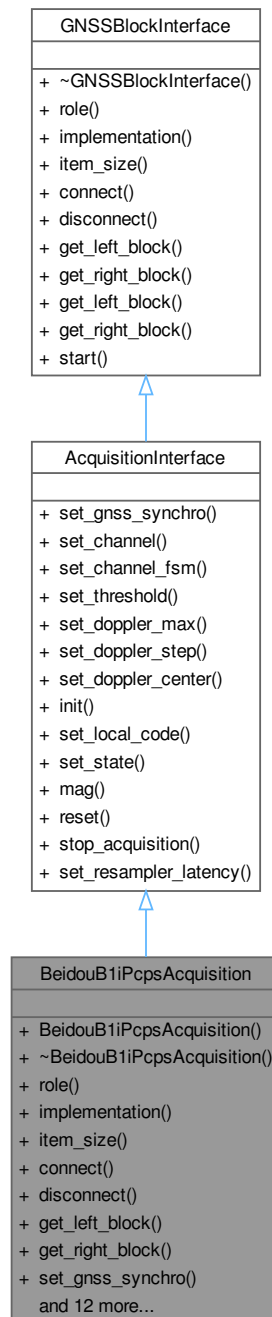
This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.

```
#include <beidou_bli_pcps_acquisition.h>
```

Inheritance diagram for BeidouB1iPcpsAcquisition:



Collaboration diagram for BeidouB1iPcpsAcquisition:



## Public Member Functions

- **BeidouB1iPcpsAcquisition** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override
  - Returns "BEIDOU\_B1I\_PCPS\_Acquisition".
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [set\\_channel](#) (unsigned int channel) override  
*Set acquisition channel unique ID.*
- void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm) override  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold) override  
*Set statistics threshold of PCPS algorithm.*
- void [set\\_doppler\\_max](#) (uint32\_t doppler\_max) override  
*Set maximum Doppler off grid search.*
- void [set\\_doppler\\_step](#) (uint32\_t doppler\_step) override  
*Set Doppler steps for the grid search.*
- void [init](#) () override  
*Initializes acquisition algorithm.*
- void [set\\_local\\_code](#) () override  
*Sets local code for GPS L1/CA PCPS acquisition algorithm.*
- signed int [mag](#) () override  
*Returns the maximum peak of grid search.*
- void [reset](#) () override  
*Restart acquisition algorithm.*
- void [set\\_state](#) (int state) override  
*If state = 1, it forces the block to start acquiring from the first sample.*
- void [stop\\_acquisition](#) () override  
*Stop running acquisition.*
- void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples) override  
*Sets the resampler latency to account it in the acquisition code delay estimation.*

### Public Member Functions inherited from [AcquisitionInterface](#)

- virtual void [set\\_doppler\\_max](#) (unsigned int doppler\_max)=0
- virtual void [set\\_doppler\\_step](#) (unsigned int doppler\_step)=0
- virtual void [set\\_doppler\\_center](#) (int doppler\_center \_\_attribute\_\_((unused)))

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

## 10.28.1 Detailed Description

This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.  
 Definition at line 47 of file [beidou\\_b1i\\_pcps\\_acquisition.h](#).

## 10.28.2 Member Function Documentation

### 10.28.2.1 connect()

```
void BeidouBliPcpsAcquisition::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.28.2.2 disconnect()

```
void BeidouBliPcpsAcquisition::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.28.2.3 get\_left\_block()

```
gr::basic_block_sptr BeidouBliPcpsAcquisition::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.28.2.4 get\_right\_block()

```
gr::basic_block_sptr BeidouBliPcpsAcquisition::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.28.2.5 implementation()

```
std::string BeidouBliPcpsAcquisition::implementation () [inline], [override], [virtual]
```

Returns "BEIDOU\_B1I\_PCPS\_Acquisition".  
Implements [GNSSBlockInterface](#).  
Definition at line 64 of file [beidou\\_b1i\\_pcps\\_acquisition.h](#).

### 10.28.2.6 init()

```
void BeidouBliPcpsAcquisition::init () [override], [virtual]
```

Initializes acquisition algorithm.  
Implements [AcquisitionInterface](#).

### 10.28.2.7 item\_size()

```
size_t BeidouBliPcpsAcquisition::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).  
Definition at line 69 of file [beidou\\_b1i\\_pcps\\_acquisition.h](#).

### 10.28.2.8 mag()

```
signed int BeidouBliPcpsAcquisition::mag () [override], [virtual]
```

Returns the maximum peak of grid search.  
Implements [AcquisitionInterface](#).

### 10.28.2.9 reset()

```
void BeidouBliPcpsAcquisition::reset () [override], [virtual]
```

Restart acquisition algorithm.  
Implements [AcquisitionInterface](#).

### 10.28.2.10 role()

```
std::string BeidouBliPcpsAcquisition::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).  
Definition at line 56 of file [beidou\\_b1i\\_pcps\\_acquisition.h](#).

**10.28.2.11 set\_channel()**

```
void BeidouBliPcpsAcquisition::set_channel (
    unsigned int channel) [inline], [override], [virtual]
```

Set acquisition channel unique ID.

Implements [AcquisitionInterface](#).

Definition at line 89 of file [beidou\\_b1i\\_pcps\\_acquisition.h](#).

**10.28.2.12 set\_channel\_fsm()**

```
void BeidouBliPcpsAcquisition::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline], [override], [virtual]
```

Set channel fsm associated to this acquisition instance.

Implements [AcquisitionInterface](#).

Definition at line 98 of file [beidou\\_b1i\\_pcps\\_acquisition.h](#).

**10.28.2.13 set\_doppler\_max()**

```
void BeidouBliPcpsAcquisition::set_doppler_max (
    uint32_t doppler_max) [override]
```

Set maximum Doppler off grid search.

**10.28.2.14 set\_doppler\_step()**

```
void BeidouBliPcpsAcquisition::set_doppler_step (
    uint32_t doppler_step) [override]
```

Set Doppler steps for the grid search.

**10.28.2.15 set\_gnss\_synchro()**

```
void BeidouBliPcpsAcquisition::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [AcquisitionInterface](#).

**10.28.2.16 set\_local\_code()**

```
void BeidouBliPcpsAcquisition::set_local_code () [override], [virtual]
```

Sets local code for GPS L1/CA PCPS acquisition algorithm.

Implements [AcquisitionInterface](#).

**10.28.2.17 set\_resampler\_latency()**

```
void BeidouBliPcpsAcquisition::set_resampler_latency (
    uint32_t latency_samples) [override], [virtual]
```

Sets the resampler latency to account it in the acquisition code delay estimation.

Implements [AcquisitionInterface](#).

**10.28.2.18 set\_state()**

```
void BeidouBliPcpsAcquisition::set_state (
    int state) [override], [virtual]
```

If state = 1, it forces the block to start acquiring from the first sample.

Implements [AcquisitionInterface](#).

**10.28.2.19 set\_threshold()**

```
void BeidouBliPcpsAcquisition::set_threshold (
    float threshold) [override], [virtual]
```



Set statistics threshold of PCPS algorithm.  
Implements [AcquisitionInterface](#).

#### 10.28.2.20 stop\_acquisition()

```
void BeidouB1iPcpsAcquisition::stop_acquisition () [override], [virtual]
```

Stop running acquisition.

Implements [AcquisitionInterface](#).

The documentation for this class was generated from the following file:

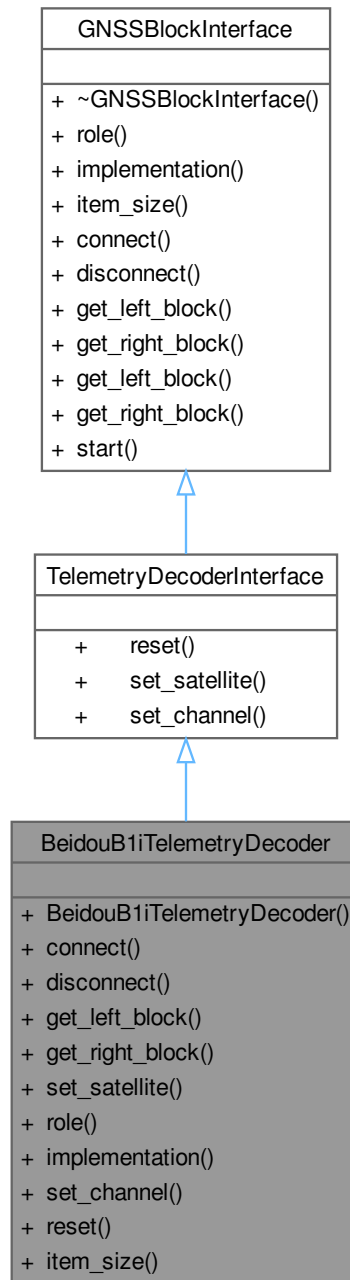
- [beidou\\_b1i\\_pcps\\_acquisition.h](#)

## 10.29 BeidouB1iTelemetryDecoder Class Reference

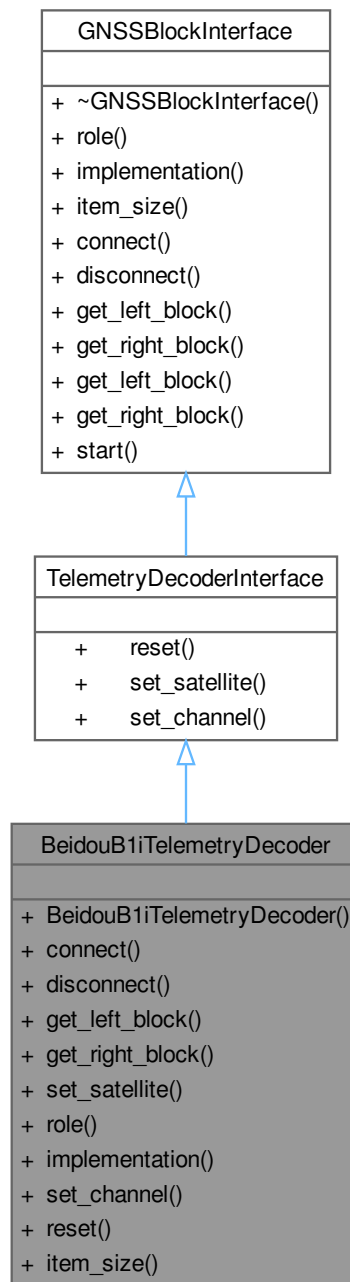
This class implements a NAV data decoder for BEIDOU B1I.

```
#include <beidou_b1i_telemetry_decoder.h>
```

Inheritance diagram for BeidouB1iTelemetryDecoder:



Collaboration diagram for BeidouB1iTelemetryDecoder:



### Public Member Functions

- **BeidouB1iTelemetryDecoder** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override

- void [set\\_satellite](#) (const [Gnss\\_Satellite](#) &satellite) override
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "[BEIDOU\\_B1I\\_Telemetry\\_Decoder](#)".
- void [set\\_channel](#) (int channel) override
- void [reset](#) () override
- size\_t [item\\_size](#) () override

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

## 10.29.1 Detailed Description

This class implements a NAV data decoder for BEIDOU B1I.  
Definition at line 43 of file [beidou\\_b1i\\_telemetry\\_decoder.h](#).

## 10.29.2 Member Function Documentation

### 10.29.2.1 connect()

```
void BeidouBliTelemetryDecoder::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.29.2.2 disconnect()

```
void BeidouBliTelemetryDecoder::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.29.2.3 get\_left\_block()

```
gr::basic_block_sptr BeidouBliTelemetryDecoder::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.29.2.4 get\_right\_block()

```
gr::basic_block_sptr BeidouBliTelemetryDecoder::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.29.2.5 implementation()

```
std::string BeidouBliTelemetryDecoder::implementation () [inline], [override], [virtual]
```

Returns "[BEIDOU\\_B1I\\_Telemetry\\_Decoder](#)".

Implements [GNSSBlockInterface](#).

Definition at line 65 of file [beidou\\_b1i\\_telemetry\\_decoder.h](#).

### 10.29.2.6 item\_size()

```
size_t BeidouBliTelemetryDecoder::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 77 of file [beidou\\_b1i\\_telemetry\\_decoder.h](#).

### 10.29.2.7 reset()

```
void BeidouBliTelemetryDecoder::reset () [inline], [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

Definition at line 72 of file [beidou\\_b1i\\_telemetry\\_decoder.h](#).

### 10.29.2.8 role()

```
std::string BeidouBliTelemetryDecoder::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 59 of file [beidou\\_b1i\\_telemetry\\_decoder.h](#).

### 10.29.2.9 set\_channel()

```
void BeidouBliTelemetryDecoder::set_channel (  
    int channel) [inline], [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

Definition at line 70 of file [beidou\\_b1i\\_telemetry\\_decoder.h](#).

### 10.29.2.10 set\_satellite()

```
void BeidouBliTelemetryDecoder::set_satellite (  
    const Gnss\_Satellite & satellite) [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

The documentation for this class was generated from the following file:

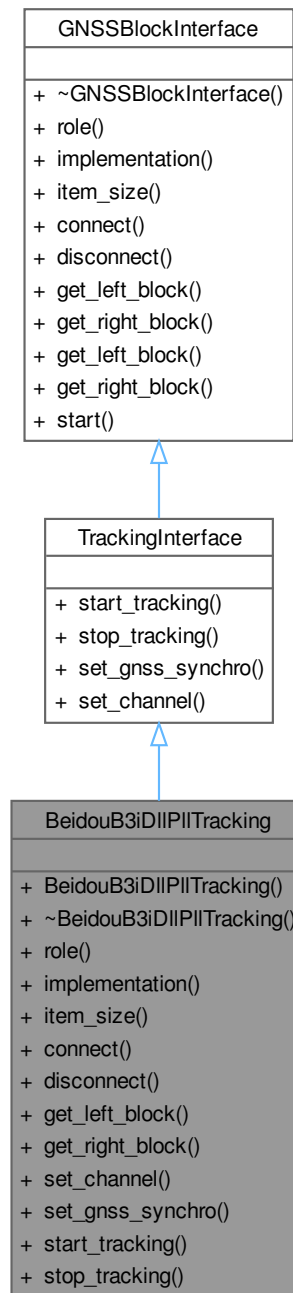
- [beidou\\_b1i\\_telemetry\\_decoder.h](#)

## 10.30 BeidouB3iDIIPITracking Class Reference

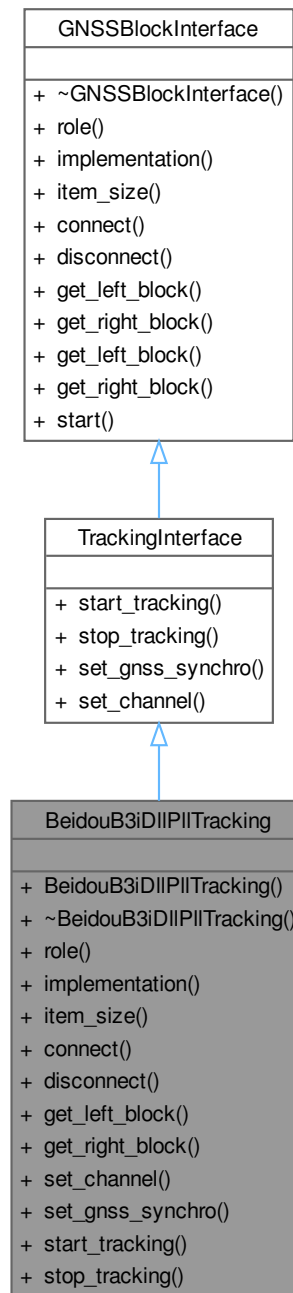
This class implements a code DLL + carrier PLL tracking loop.

```
#include <beidou_b3i_dll_pll_tracking.h>
```

Inheritance diagram for BeidouB3iDIIPIITracking:



Collaboration diagram for BeidouB3iDIIPIITracking:



### Public Member Functions

- **BeidouB3iDIIPIITracking** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override

- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_channel](#) (unsigned int channel) override
 

*Set tracking channel unique ID.*
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override
 

*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [start\\_tracking](#) () override
- void [stop\\_tracking](#) () override
 

*Stop running tracking.*

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()
 

*Start the flow of samples if needed.*

### 10.30.1 Detailed Description

This class implements a code DLL + carrier PLL tracking loop.  
Definition at line 41 of file [beidou\\_b3i\\_dll\\_pll\\_tracking.h](#).

### 10.30.2 Member Function Documentation

#### 10.30.2.1 connect()

```
void BeidouB3iDllPllTracking::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.30.2.2 disconnect()

```
void BeidouB3iDllPllTracking::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.30.2.3 get\_left\_block()

```
gr::basic_block_sptr BeidouB3iDllPllTracking::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.30.2.4 get\_right\_block()

```
gr::basic_block_sptr BeidouB3iDllPllTracking::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.30.2.5 implementation()

```
std::string BeidouB3iDllPllTracking::implementation () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).  
Definition at line 57 of file [beidou\\_b3i\\_dll\\_pll\\_tracking.h](#).

#### 10.30.2.6 item\_size()

```
size_t BeidouB3iDllPllTracking::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).  
Definition at line 62 of file [beidou\\_b3i\\_dll\\_pll\\_tracking.h](#).



**10.30.2.7 role()**

```
std::string BeidouB3iDllPllTracking::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 52 of file [beidou\\_b3i\\_dll\\_pll\\_tracking.h](#).

**10.30.2.8 set\_channel()**

```
void BeidouB3iDllPllTracking::set_channel (
    unsigned int channel) [override], [virtual]
```

Set tracking channel unique ID.

Implements [TrackingInterface](#).

**10.30.2.9 set\_gnss\_synchro()**

```
void BeidouB3iDllPllTracking::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [TrackingInterface](#).

**10.30.2.10 start\_tracking()**

```
void BeidouB3iDllPllTracking::start_tracking () [override], [virtual]
```

Implements [TrackingInterface](#).

**10.30.2.11 stop\_tracking()**

```
void BeidouB3iDllPllTracking::stop_tracking () [override], [virtual]
```

Stop running tracking.

Implements [TrackingInterface](#).

The documentation for this class was generated from the following file:

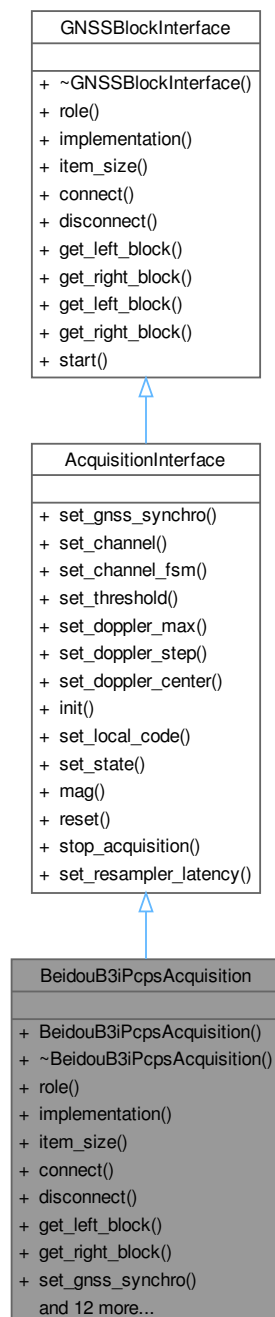
- [beidou\\_b3i\\_dll\\_pll\\_tracking.h](#)

**10.31 BeidouB3iPcpsAcquisition Class Reference**

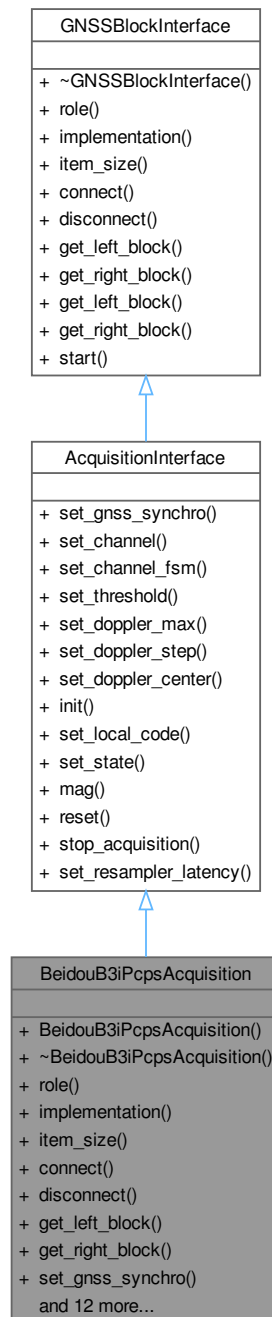
This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for BeiDou B3I signals.

```
#include <beidou_b3i_pcps_acquisition.h>
```

Inheritance diagram for BeidouB3iPcpsAcquisition:



Collaboration diagram for BeidouB3iPcpsAcquisition:



## Public Member Functions

- **BeidouB3iPcpsAcquisition** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override
  - Returns "BEIDOU\_B1I\_PCPS\_Acquisition".
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [set\\_channel](#) (unsigned int channel) override  
*Set acquisition channel unique ID.*
- void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm) override  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold) override  
*Set statistics threshold of PCPS algorithm.*
- void [set\\_doppler\\_max](#) (unsigned int doppler\_max) override  
*Set maximum Doppler off grid search.*
- void [set\\_doppler\\_step](#) (unsigned int doppler\_step) override  
*Set Doppler steps for the grid search.*
- void [init](#) () override  
*Initializes acquisition algorithm.*
- void [set\\_local\\_code](#) () override  
*Sets local code for GPS L1/CA PCPS acquisition algorithm.*
- signed int [mag](#) () override  
*Returns the maximum peak of grid search.*
- void [reset](#) () override  
*Restart acquisition algorithm.*
- void [set\\_state](#) (int state) override  
*If state = 1, it forces the block to start acquiring from the first sample.*
- void [stop\\_acquisition](#) () override  
*Stop running acquisition.*
- void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples) override  
*Sets the resampler latency to account it in the acquisition code delay estimation.*

## Public Member Functions inherited from [AcquisitionInterface](#)

- virtual void [set\\_doppler\\_center](#) (int doppler\_center \_\_attribute\_\_((unused)))

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.31.1 Detailed Description

This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for BeiDou B3I signals.  
 Definition at line 46 of file [beidou\\_b3i\\_pcps\\_acquisition.h](#).

### 10.31.2 Member Function Documentation

#### 10.31.2.1 [connect](#)()

```
void BeidouB3iPcpsAcquisition::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.31.2.2 disconnect()

```
void BeidouB3iPcpsAcquisition::disconnect (  
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.31.2.3 get\_left\_block()

```
gr::basic_block_sptr BeidouB3iPcpsAcquisition::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.31.2.4 get\_right\_block()

```
gr::basic_block_sptr BeidouB3iPcpsAcquisition::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.31.2.5 implementation()

```
std::string BeidouB3iPcpsAcquisition::implementation () [inline], [override], [virtual]
```

Returns "BEIDOU\_B1I\_PCPS\_Acquisition".

Implements [GNSSBlockInterface](#).

Definition at line 63 of file [beidou\\_b3i\\_pcps\\_acquisition.h](#).

### 10.31.2.6 init()

```
void BeidouB3iPcpsAcquisition::init () [override], [virtual]
```

Initializes acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.31.2.7 item\_size()

```
size_t BeidouB3iPcpsAcquisition::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 68 of file [beidou\\_b3i\\_pcps\\_acquisition.h](#).

### 10.31.2.8 mag()

```
signed int BeidouB3iPcpsAcquisition::mag () [override], [virtual]
```

Returns the maximum peak of grid search.

Implements [AcquisitionInterface](#).

### 10.31.2.9 reset()

```
void BeidouB3iPcpsAcquisition::reset () [override], [virtual]
```

Restart acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.31.2.10 role()

```
std::string BeidouB3iPcpsAcquisition::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 55 of file [beidou\\_b3i\\_pcps\\_acquisition.h](#).

### 10.31.2.11 set\_channel()

```
void BeidouB3iPcpsAcquisition::set_channel (  
    unsigned int channel) [inline], [override], [virtual]
```

Set acquisition channel unique ID.

Implements [AcquisitionInterface](#).

Definition at line 88 of file [beidou\\_b3i\\_pcps\\_acquisition.h](#).

**10.31.2.12 set\_channel\_fsm()**

```
void BeidouB3iPcpsAcquisition::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline], [override], [virtual]
```

Set channel fsm associated to this acquisition instance.

Implements [AcquisitionInterface](#).

Definition at line 97 of file [beidou\\_b3i\\_pcps\\_acquisition.h](#).

**10.31.2.13 set\_doppler\_max()**

```
void BeidouB3iPcpsAcquisition::set_doppler_max (
    unsigned int doppler_max) [override], [virtual]
```

Set maximum Doppler off grid search.

Implements [AcquisitionInterface](#).

**10.31.2.14 set\_doppler\_step()**

```
void BeidouB3iPcpsAcquisition::set_doppler_step (
    unsigned int doppler_step) [override], [virtual]
```

Set Doppler steps for the grid search.

Implements [AcquisitionInterface](#).

**10.31.2.15 set\_gnss\_synchro()**

```
void BeidouB3iPcpsAcquisition::set_gnss_synchro (
    Gnss\_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [AcquisitionInterface](#).

**10.31.2.16 set\_local\_code()**

```
void BeidouB3iPcpsAcquisition::set_local_code () [override], [virtual]
```

Sets local code for GPS L1/CA PCPS acquisition algorithm.

Implements [AcquisitionInterface](#).

**10.31.2.17 set\_resampler\_latency()**

```
void BeidouB3iPcpsAcquisition::set_resampler_latency (
    uint32_t latency_samples) [override], [virtual]
```

Sets the resampler latency to account it in the acquisition code delay estimation.

Implements [AcquisitionInterface](#).

**10.31.2.18 set\_state()**

```
void BeidouB3iPcpsAcquisition::set_state (
    int state) [override], [virtual]
```

If state = 1, it forces the block to start acquiring from the first sample.

Implements [AcquisitionInterface](#).

**10.31.2.19 set\_threshold()**

```
void BeidouB3iPcpsAcquisition::set_threshold (
    float threshold) [override], [virtual]
```

Set statistics threshold of PCPS algorithm.

Implements [AcquisitionInterface](#).

#### 10.31.2.20 stop\_acquisition()

```
void BeidouB3iPcpsAcquisition::stop_acquisition () [override], [virtual]
```

Stop running acquisition.

Implements [AcquisitionInterface](#).

The documentation for this class was generated from the following file:

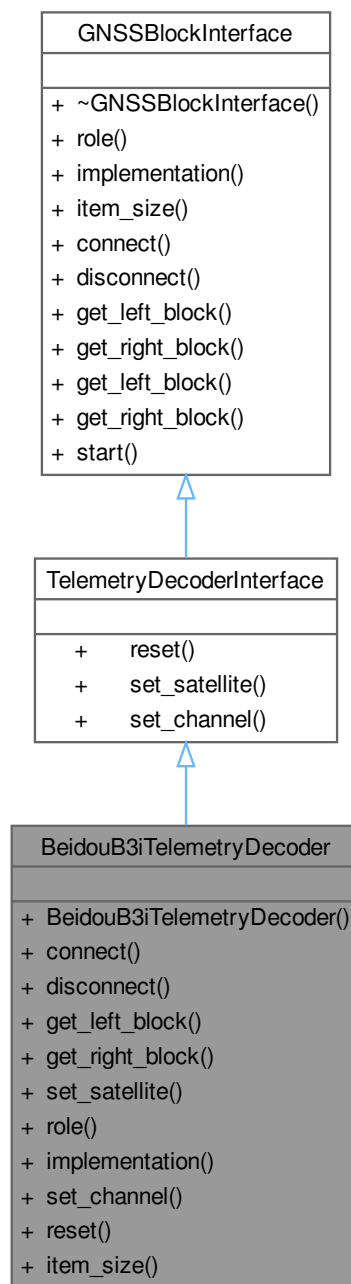
- [beidou\\_b3i\\_pcps\\_acquisition.h](#)

## 10.32 BeidouB3iTelemetryDecoder Class Reference

This class implements a NAV data decoder for BEIDOU B1I.

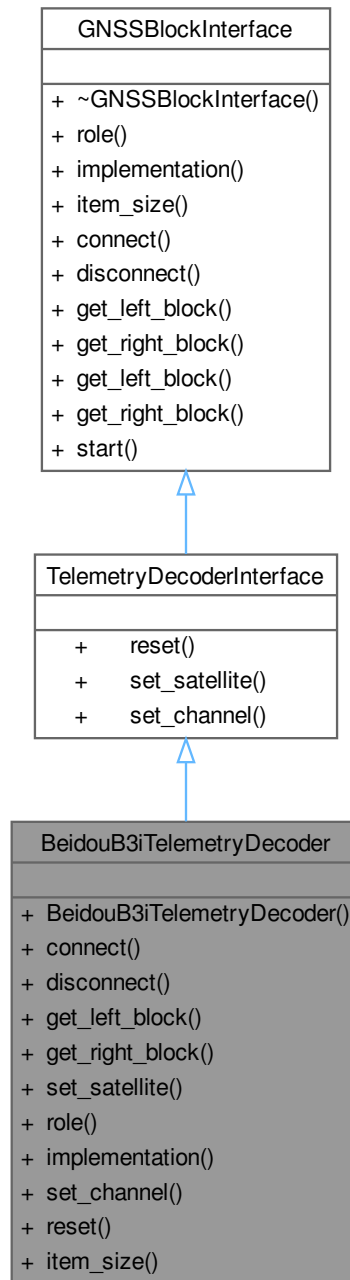
```
#include <beidou_b3i_telemetry_decoder.h>
```

Inheritance diagram for BeidouB3iTelemetryDecoder:





Collaboration diagram for BeidouB3iTelemetryDecoder:



### Public Member Functions

- **BeidouB3iTelemetryDecoder** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override

- void [set\\_satellite](#) (const [Gnss\\_Satellite](#) &satellite) override
- std::string [role](#) () override
- std::string [implementation](#) () override
- Returns "BEIDOU\_B3I\_Telemetry\_Decoder".*
- void [set\\_channel](#) (int channel) override
- void [reset](#) () override
- size\_t [item\\_size](#) () override

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.32.1 Detailed Description

This class implements a NAV data decoder for BEIDOU B1I.

Definition at line 42 of file [beidou\\_b3i\\_telemetry\\_decoder.h](#).

### 10.32.2 Member Function Documentation

#### 10.32.2.1 [connect\(\)](#)

```
void BeidouB3iTelemetryDecoder::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.32.2.2 [disconnect\(\)](#)

```
void BeidouB3iTelemetryDecoder::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.32.2.3 [get\\_left\\_block\(\)](#)

```
gr::basic_block_sptr BeidouB3iTelemetryDecoder::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.32.2.4 [get\\_right\\_block\(\)](#)

```
gr::basic_block_sptr BeidouB3iTelemetryDecoder::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.32.2.5 [implementation\(\)](#)

```
std::string BeidouB3iTelemetryDecoder::implementation () [inline], [override], [virtual]
```

Returns "BEIDOU\_B3I\_Telemetry\_Decoder".

Implements [GNSSBlockInterface](#).

Definition at line 60 of file [beidou\\_b3i\\_telemetry\\_decoder.h](#).

#### 10.32.2.6 [item\\_size\(\)](#)

```
size_t BeidouB3iTelemetryDecoder::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 75 of file [beidou\\_b3i\\_telemetry\\_decoder.h](#).

**10.32.2.7 reset()**

```
void BeidouB3iTelemetryDecoder::reset () [inline], [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

Definition at line 70 of file [beidou\\_b3i\\_telemetry\\_decoder.h](#).

**10.32.2.8 role()**

```
std::string BeidouB3iTelemetryDecoder::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 57 of file [beidou\\_b3i\\_telemetry\\_decoder.h](#).

**10.32.2.9 set\_channel()**

```
void BeidouB3iTelemetryDecoder::set_channel (
    int channel) [inline], [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

Definition at line 65 of file [beidou\\_b3i\\_telemetry\\_decoder.h](#).

**10.32.2.10 set\_satellite()**

```
void BeidouB3iTelemetryDecoder::set_satellite (
    const Gnss_Satellite & satellite) [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

The documentation for this class was generated from the following file:

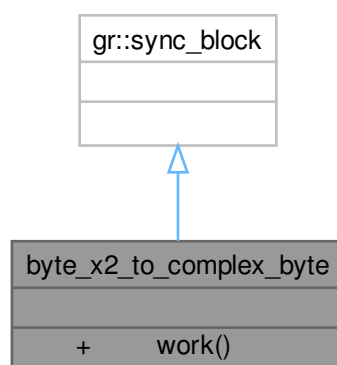
- [beidou\\_b3i\\_telemetry\\_decoder.h](#)

**10.33 byte\_x2\_to\_complex\_byte Class Reference**

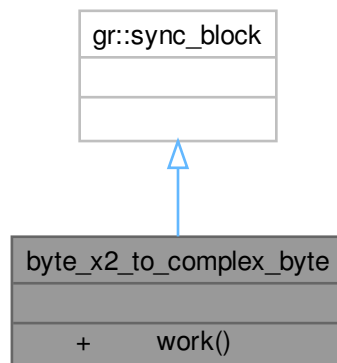
This class adapts two signed char streams into a `std::complex<signed char>` stream.

```
#include <byte_x2_to_complex_byte.h>
```

Inheritance diagram for `byte_x2_to_complex_byte`:



Collaboration diagram for `byte_x2_to_complex_byte`:



### Public Member Functions

- `int` **work** (`int` noutput\_items, `gr_vector_const_void_star` &input\_items, `gr_vector_void_star` &output\_items)

### Friends

- `byte_x2_to_complex_byte_sptr` **make\_byte\_x2\_to\_complex\_byte** ()

### 10.33.1 Detailed Description

This class adapts two signed char streams into a `std::complex<signed char>` stream.  
 Definition at line 41 of file [byte\\_x2\\_to\\_complex\\_byte.h](#).

The documentation for this class was generated from the following file:

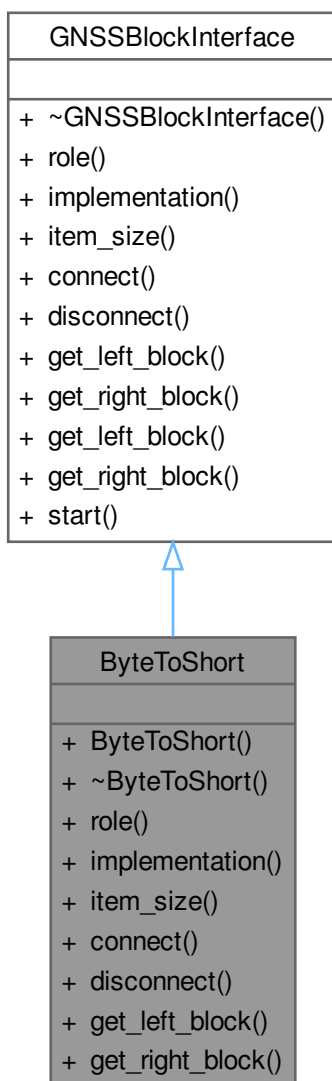
- [byte\\_x2\\_to\\_complex\\_byte.h](#)

## 10.34 ByteToShort Class Reference

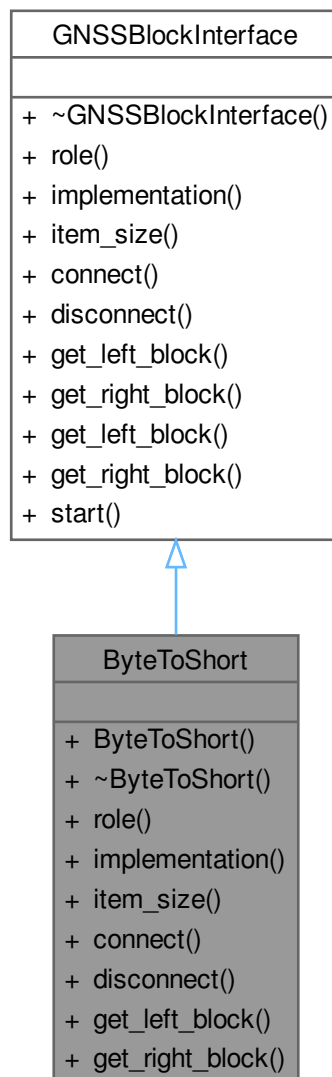
Adapts an 8-bits sample stream (IF) to a short int stream (IF)

```
#include <byte_to_short.h>
```

Inheritance diagram for ByteToShort:



Collaboration diagram for ByteToShort:



### Public Member Functions

- **ByteToShort** (const [ConfigurationInterface](#) \*configuration, std::string role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "Byte\_To\_Short".
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override

## Public Member Functions inherited from GNSSBlockInterface

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.34.1 Detailed Description

Adapts an 8-bits sample stream (IF) to a short int stream (IF)

Definition at line 40 of file [byte\\_to\\_short.h](#).

### 10.34.2 Member Function Documentation

#### 10.34.2.1 connect()

```
void ByteToShort::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.34.2.2 disconnect()

```
void ByteToShort::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.34.2.3 get\_left\_block()

```
gr::basic_block_sptr ByteToShort::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.34.2.4 get\_right\_block()

```
gr::basic_block_sptr ByteToShort::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.34.2.5 implementation()

```
std::string ByteToShort::implementation () [inline], [override], [virtual]
```

Returns "Byte\_To\_Short".

Implements [GNSSBlockInterface](#).

Definition at line 55 of file [byte\\_to\\_short.h](#).

#### 10.34.2.6 item\_size()

```
size_t ByteToShort::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 60 of file [byte\\_to\\_short.h](#).

#### 10.34.2.7 role()

```
std::string ByteToShort::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 49 of file [byte\\_to\\_short.h](#).

The documentation for this class was generated from the following file:

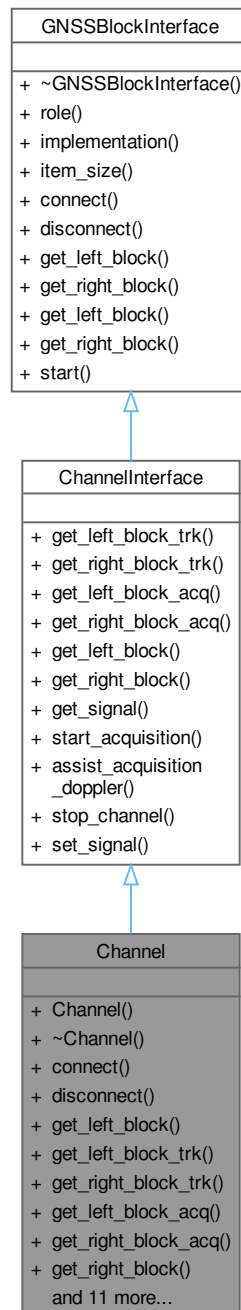
- [byte\\_to\\_short.h](#)

## 10.35 Channel Class Reference

This class represents a GNSS channel. It wraps an [AcquisitionInterface](#), a [TrackingInterface](#) and a [TelemetryDecoderInterface](#), and handles their interaction through a Finite State Machine.

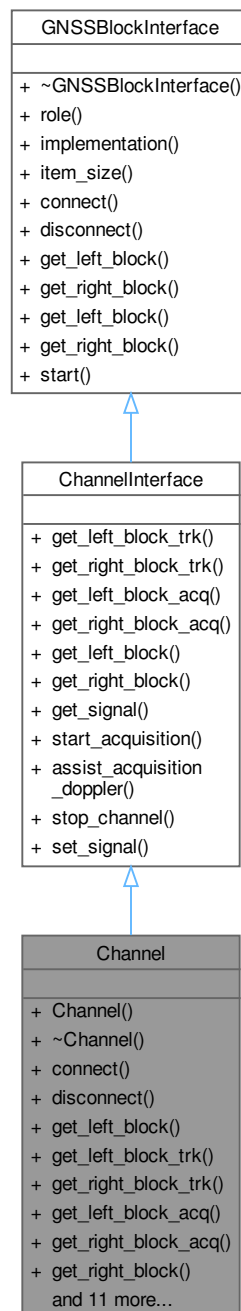
```
#include <channel.h>
```

Inheritance diagram for Channel:





Collaboration diagram for Channel:



## Public Member Functions

- [Channel](#) (const [ConfigurationInterface](#) \*configuration, uint32\_t channel, std::shared\_ptr< [AcquisitionInterface](#) > acq, std::shared\_ptr< [TrackingInterface](#) > trk, std::shared\_ptr< [TelemetryDecoderInterface](#) > nav, const std::string &role, const std::string &signal\_str, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)

*Constructor.*

- [~Channel](#) ()=default

*Destructor.*

- void [connect](#) (gr::top\_block\_sptr top\_block) override  
*Connects the tracking block to the top\_block and to the telemetry.*
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_left\\_block\\_trk](#) () override  
*Gets the GNU Radio tracking block input pointer.*
- gr::basic\_block\_sptr [get\\_right\\_block\\_trk](#) () override  
*Gets the GNU Radio tracking block output pointer.*
- gr::basic\_block\_sptr [get\\_left\\_block\\_acq](#) () override  
*Gets the GNU Radio acquisition block input pointer.*
- gr::basic\_block\_sptr [get\\_right\\_block\\_acq](#) () override  
*Gets the GNU Radio acquisition block output pointer.*
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override  
*Gets the GNU Radio channel block output pointer.*
- std::string [role](#) () override
- std::string [implementation](#) () override  
*Returns "Channel".*
- size\_t [item\\_size](#) () override
- [Gnss\\_Signal](#) [get\\_signal](#) () override
- void [start\\_acquisition](#) () override  
*Start the State Machine.*
- void [stop\\_channel](#) () override  
*Stop the State Machine.*
- void [set\\_signal](#) (const [Gnss\\_Signal](#) &gnss\_signal\_) override  
*Sets the channel GNSS signal.*
- void [assist\\_acquisition\\_doppler](#) (double Carrier\_Doppler\_hz) override
- std::shared\_ptr< [AcquisitionInterface](#) > [acquisition](#) () const
- std::shared\_ptr< [TrackingInterface](#) > [tracking](#) () const
- std::shared\_ptr< [TelemetryDecoderInterface](#) > [telemetry](#) () const

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()  
*Start the flow of samples if needed.*

### 10.35.1 Detailed Description

This class represents a GNSS channel. It wraps an [AcquisitionInterface](#), a [TrackingInterface](#) and a [TelemetryDecoderInterface](#), and handles their interaction through a Finite State Machine. Definition at line 60 of file [channel.h](#).

### 10.35.2 Constructor & Destructor Documentation

#### 10.35.2.1 Channel()

```
Channel::Channel (
    const ConfigurationInterface * configuration,
    uint32_t channel,
    std::shared_ptr< AcquisitionInterface > acq,
    std::shared_ptr< TrackingInterface > trk,
    std::shared_ptr< TelemetryDecoderInterface > nav,
    const std::string & role,
    const std::string & signal_str,
    Concurrent\_Queue< pmt::pmt_t > * queue)
```

Constructor.

### 10.35.2.2 ~Channel()

Channel::~Channel () [default]  
Destructor.

## 10.35.3 Member Function Documentation

### 10.35.3.1 acquisition()

std::shared\_ptr< [AcquisitionInterface](#) > Channel::acquisition () const [inline]  
Definition at line 94 of file [channel.h](#).

### 10.35.3.2 assist\_acquisition\_doppler()

void Channel::assist\_acquisition\_doppler (  
double *Carrier\_Doppler\_hz*) [override], [virtual]  
Implements [ChannelInterface](#).

### 10.35.3.3 connect()

void Channel::connect (  
gr::top\_block\_sptr *top\_block*) [override], [virtual]  
Connects the tracking block to the top\_block and to the telemetry.  
Implements [GNSSBlockInterface](#).

### 10.35.3.4 disconnect()

void Channel::disconnect (  
gr::top\_block\_sptr *top\_block*) [override], [virtual]  
Implements [GNSSBlockInterface](#).

### 10.35.3.5 get\_left\_block()

gr::basic\_block\_sptr Channel::get\_left\_block () [override], [virtual]  
Implements [ChannelInterface](#).

### 10.35.3.6 get\_left\_block\_acq()

gr::basic\_block\_sptr Channel::get\_left\_block\_acq () [override], [virtual]  
Gets the GNU Radio acquisition block input pointer.  
Implements [ChannelInterface](#).

### 10.35.3.7 get\_left\_block\_trk()

gr::basic\_block\_sptr Channel::get\_left\_block\_trk () [override], [virtual]  
Gets the GNU Radio tracking block input pointer.  
Implements [ChannelInterface](#).

### 10.35.3.8 get\_right\_block()

gr::basic\_block\_sptr Channel::get\_right\_block () [override], [virtual]  
Gets the GNU Radio channel block output pointer.  
Implements [ChannelInterface](#).

### 10.35.3.9 get\_right\_block\_acq()

gr::basic\_block\_sptr Channel::get\_right\_block\_acq () [override], [virtual]  
Gets the GNU Radio acquisition block output pointer.  
Implements [ChannelInterface](#).

**10.35.3.10 get\_right\_block\_trk()**

```
gr::basic_block_sptr Channel::get_right_block_trk () [override], [virtual]
```

Gets the GNU Radio tracking block output pointer.

Implements [ChannelInterface](#).

**10.35.3.11 get\_signal()**

```
Gnss_Signal Channel::get_signal () [override], [virtual]
```

Implements [ChannelInterface](#).

**10.35.3.12 implementation()**

```
std::string Channel::implementation () [inline], [override], [virtual]
```

Returns "Channel".

Implements [GNSSBlockInterface](#).

Definition at line 85 of file [channel.h](#).

**10.35.3.13 item\_size()**

```
size_t Channel::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 86 of file [channel.h](#).

**10.35.3.14 role()**

```
std::string Channel::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 84 of file [channel.h](#).

**10.35.3.15 set\_signal()**

```
void Channel::set_signal (
    const Gnss_Signal & gnss_signal_) [override], [virtual]
```

Sets the channel GNSS signal.

Implements [ChannelInterface](#).

**10.35.3.16 start\_acquisition()**

```
void Channel::start_acquisition () [override], [virtual]
```

Start the State Machine.

Implements [ChannelInterface](#).

**10.35.3.17 stop\_channel()**

```
void Channel::stop_channel () [override], [virtual]
```

Stop the State Machine.

Implements [ChannelInterface](#).

**10.35.3.18 telemetry()**

```
std::shared_ptr< TelemetryDecoderInterface > Channel::telemetry () const [inline]
```

Definition at line 96 of file [channel.h](#).

**10.35.3.19 tracking()**

```
std::shared_ptr< TrackingInterface > Channel::tracking () const [inline]
```

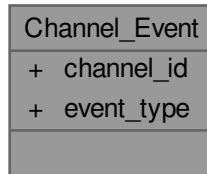
Definition at line 95 of file [channel.h](#).

The documentation for this class was generated from the following file:

- [channel.h](#)

## 10.36 Channel\_Event Class Reference

Collaboration diagram for Channel\_Event:



### Public Attributes

- int [channel\\_id](#)
- int [event\\_type](#)

### Friends

- `channel_event_sptr channel_event_make (int channel_id, int event_type)`

### 10.36.1 Detailed Description

Definition at line 34 of file [channel\\_event.h](#).

### 10.36.2 Member Data Documentation

#### 10.36.2.1 channel\_id

```
int Channel_Event::channel_id
```

Definition at line 37 of file [channel\\_event.h](#).

#### 10.36.2.2 event\_type

```
int Channel_Event::event_type
```

Definition at line 38 of file [channel\\_event.h](#).

The documentation for this class was generated from the following file:

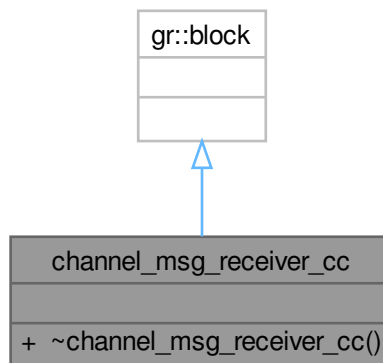
- [channel\\_event.h](#)

## 10.37 channel\_msg\_receiver\_cc Class Reference

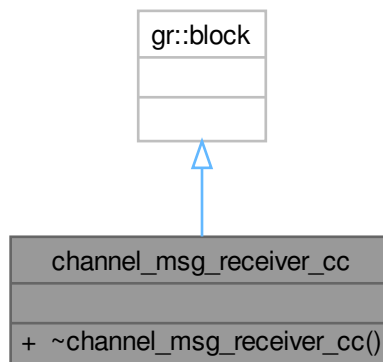
GNU Radio block that receives asynchronous channel messages from acquisition and tracking blocks.

```
#include <channel_msg_receiver_cc.h>
```

Inheritance diagram for `channel_msg_receiver_cc`:



Collaboration diagram for `channel_msg_receiver_cc`:



### Public Member Functions

- `~channel_msg_receiver_cc()`=default  
*Default destructor.*

### Friends

- `channel_msg_receiver_cc_sptr channel_msg_receiver_make_cc (std::shared_ptr< ChannelFsm > channel_fsm, bool repeat)`

## 10.37.1 Detailed Description

GNU Radio block that receives asynchronous channel messages from acquisition and tracking blocks.  
Definition at line 40 of file [channel\\_msg\\_receiver\\_cc.h](#).

## 10.37.2 Constructor & Destructor Documentation

### 10.37.2.1 ~channel\_msg\_receiver\_cc()

```
channel_msg_receiver_cc::~channel_msg_receiver_cc () [default]
```

Default destructor.

The documentation for this class was generated from the following file:

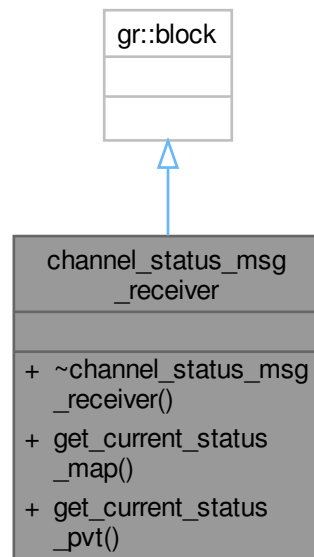
- [channel\\_msg\\_receiver\\_cc.h](#)

## 10.38 channel\_status\_msg\_receiver Class Reference

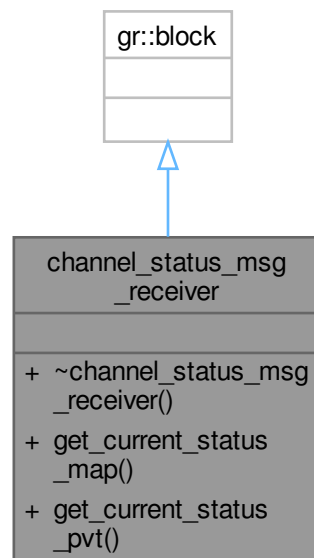
GNU Radio block that receives asynchronous channel messages from tlm blocks.

```
#include <channel_status_msg_receiver.h>
```

Inheritance diagram for channel\_status\_msg\_receiver:



Collaboration diagram for `channel_status_msg_receiver`:



### Public Member Functions

- `~channel_status_msg_receiver()`=default  
*Default destructor.*
- `std::map< int, std::shared_ptr< Gnss_Synchro > > get_current_status_map()`  
*return the current status map of all channels with valid telemetry*
- `Monitor_Pvt get_current_status_pvt()`  
*return the current receiver PVT*

### Friends

- `channel_status_msg_receiver_sptr channel_status_msg_receiver_make()`

## 10.38.1 Detailed Description

GNU Radio block that receives asynchronous channel messages from tlm blocks.  
Definition at line 44 of file [channel\\_status\\_msg\\_receiver.h](#).

## 10.38.2 Constructor & Destructor Documentation

### 10.38.2.1 ~channel\_status\_msg\_receiver()

`channel_status_msg_receiver::~channel_status_msg_receiver()` [default]  
Default destructor.

## 10.38.3 Member Function Documentation

### 10.38.3.1 get\_current\_status\_map()

`std::map< int, std::shared_ptr< Gnss_Synchro > > channel_status_msg_receiver::get_current_status_map()`



return the current status map of all channels with valid telemetry

### 10.38.3.2 get\_current\_status\_pvt()

`Monitor_Pvt channel_status_msg_receiver::get_current_status_pvt ()`

return the current receiver PVT

The documentation for this class was generated from the following file:

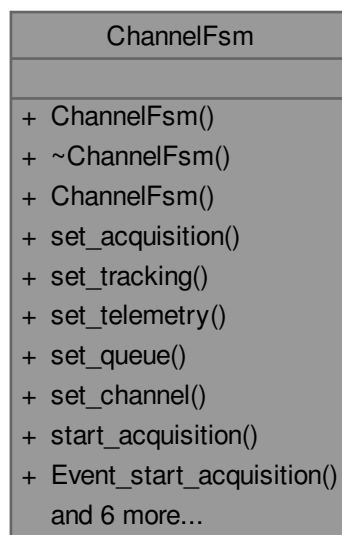
- [channel\\_status\\_msg\\_receiver.h](#)

## 10.39 ChannelFsm Class Reference

This class implements a State Machine for channel.

```
#include <channel_fsm.h>
```

Collaboration diagram for ChannelFsm:



### Public Member Functions

- **ChannelFsm** (std::shared\_ptr< [AcquisitionInterface](#) > acquisition)
- void **set\_acquisition** (std::shared\_ptr< [AcquisitionInterface](#) > acquisition)
- void **set\_tracking** (std::shared\_ptr< [TrackingInterface](#) > tracking)
- void **set\_telemetry** (std::shared\_ptr< [TelemetryDecoderInterface](#) > telemetry)
- void **set\_queue** ([Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)
- void **set\_channel** (uint32\_t channel)
- void **start\_acquisition** ()
- bool **Event\_start\_acquisition** ()
- bool **Event\_start\_acquisition\_fpga** ()
- bool **Event\_stop\_channel** ()
- bool **Event\_failed\_tracking\_standby** ()
- virtual bool **Event\_valid\_acquisition** ()
- virtual bool **Event\_failed\_acquisition\_repeat** ()
- virtual bool **Event\_failed\_acquisition\_no\_repeat** ()

### 10.39.1 Detailed Description

This class implements a State Machine for channel.

Definition at line 41 of file [channel\\_fsm.h](#).

The documentation for this class was generated from the following file:

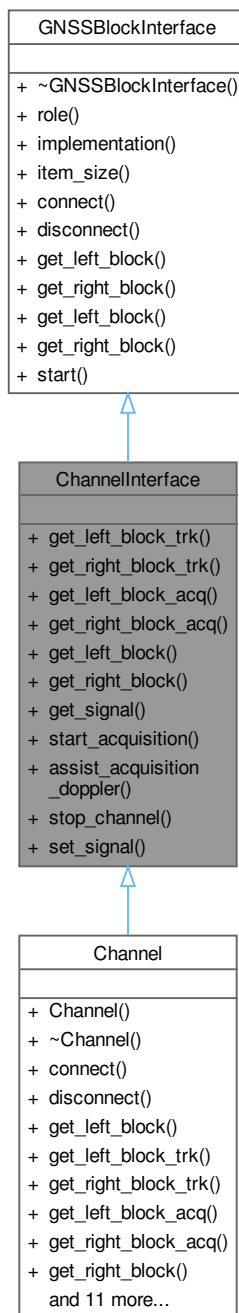
- [channel\\_fsm.h](#)

## 10.40 ChannelInterface Class Reference

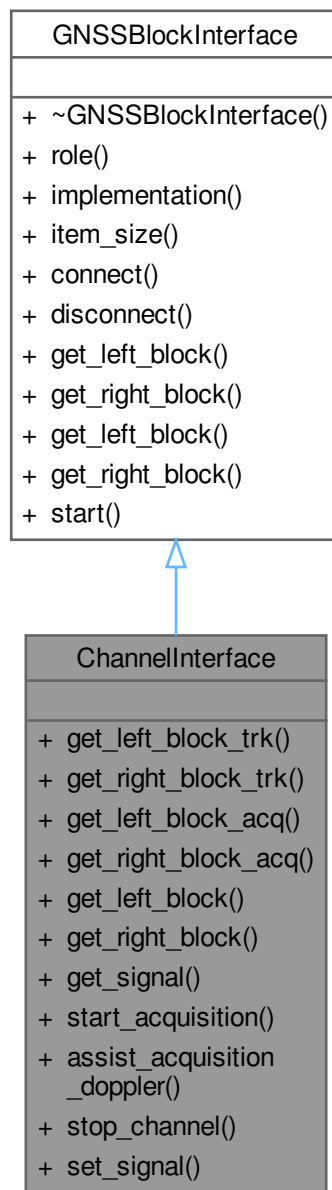
This abstract class represents an interface to a channel GNSS block.

```
#include <channel_interface.h>
```

Inheritance diagram for ChannelInterface:



Collaboration diagram for ChannelInterface:



### Public Member Functions

- virtual gr::basic\_block\_sptr [get\\_left\\_block\\_trk](#) ()=0
- virtual gr::basic\_block\_sptr [get\\_right\\_block\\_trk](#) ()=0
- virtual gr::basic\_block\_sptr [get\\_left\\_block\\_acq](#) ()=0
- virtual gr::basic\_block\_sptr [get\\_right\\_block\\_acq](#) ()=0
- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) ()=0
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) ()=0
- virtual [Gnss\\_Signal](#) [get\\_signal](#) ()=0
- virtual void [start\\_acquisition](#) ()=0

- virtual void **assist\_acquisition\_doppler** (double Carrier\_Doppler\_hz)=0
- virtual void **stop\_channel** ()=0
- virtual void **set\_signal** (const Gnss\_Signal &)=0

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual std::string **role** ()=0
- virtual std::string **implementation** ()=0
- virtual size\_t **item\_size** ()=0
- virtual void **connect** (gr::top\_block\_sptr top\_block)=0
- virtual void **disconnect** (gr::top\_block\_sptr top\_block)=0
- virtual gr::basic\_block\_sptr **get\_left\_block** (int RF\_channel)
- virtual gr::basic\_block\_sptr **get\_right\_block** (int RF\_channel)
- virtual void **start** ()

*Start the flow of samples if needed.*

### 10.40.1 Detailed Description

This abstract class represents an interface to a channel GNSS block.

Abstract class for channel blocks. Since all its methods are pure virtual, this class cannot be instantiated directly, and a subclass can only be instantiated directly if all inherited pure virtual methods have been implemented by that class or a parent class.

Definition at line 43 of file [channel\\_interface.h](#).

### 10.40.2 Member Function Documentation

#### 10.40.2.1 **get\_left\_block()**

```
virtual gr::basic_block_sptr ChannelInterface::get_left_block () [pure virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.40.2.2 **get\_left\_block\_acq()**

```
virtual gr::basic_block_sptr ChannelInterface::get_left_block_acq () [pure virtual]
```

Implemented in [Channel](#).

#### 10.40.2.3 **get\_left\_block\_trk()**

```
virtual gr::basic_block_sptr ChannelInterface::get_left_block_trk () [pure virtual]
```

Implemented in [Channel](#).

#### 10.40.2.4 **get\_right\_block()**

```
virtual gr::basic_block_sptr ChannelInterface::get_right_block () [pure virtual]
```

Implements [GNSSBlockInterface](#).

Implemented in [Channel](#).

#### 10.40.2.5 **get\_right\_block\_acq()**

```
virtual gr::basic_block_sptr ChannelInterface::get_right_block_acq () [pure virtual]
```

Implemented in [Channel](#).

#### 10.40.2.6 **get\_right\_block\_trk()**

```
virtual gr::basic_block_sptr ChannelInterface::get_right_block_trk () [pure virtual]
```

Implemented in [Channel](#).

**10.40.2.7 set\_signal()**

```
virtual void ChannelInterface::set_signal (
    const Gnss_Signal & ) [pure virtual]
```

Implemented in [Channel](#).

**10.40.2.8 start\_acquisition()**

```
virtual void ChannelInterface::start_acquisition () [pure virtual]
```

Implemented in [Channel](#).

**10.40.2.9 stop\_channel()**

```
virtual void ChannelInterface::stop_channel () [pure virtual]
```

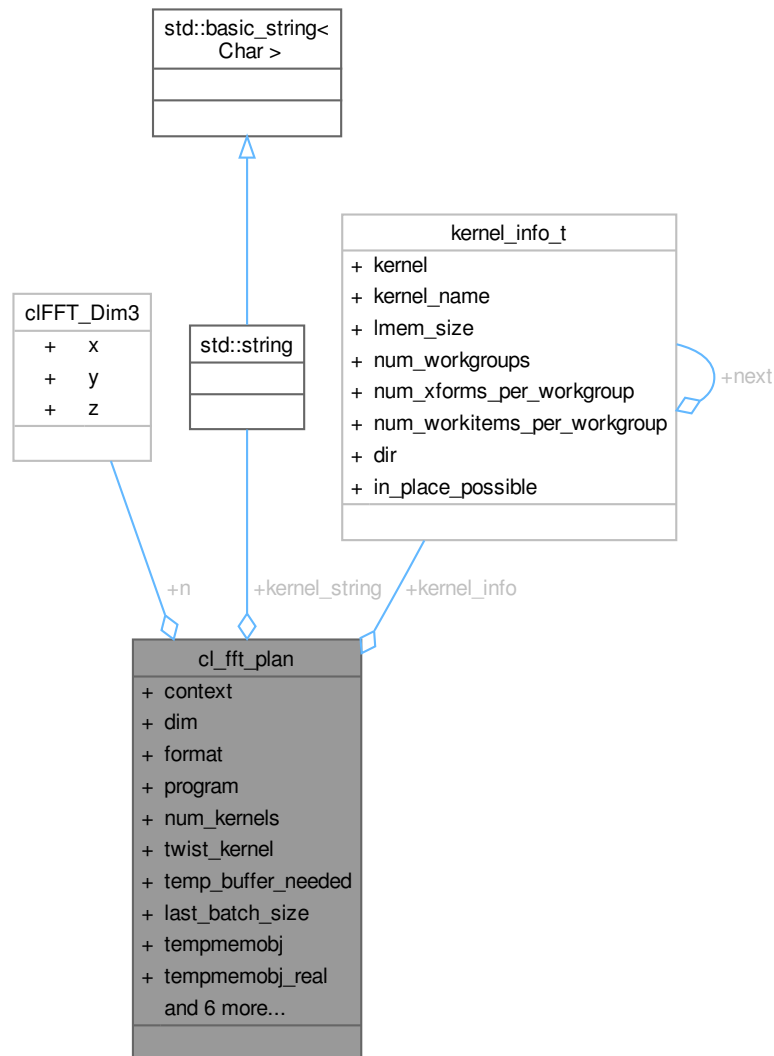
Implemented in [Channel](#).

The documentation for this class was generated from the following file:

- [channel\\_interface.h](#)

## 10.41 cl\_fft\_plan Struct Reference

Collaboration diagram for cl\_fft\_plan:



### Public Attributes

- cl\_context [context](#)
- clFFT\_Dim3 [n](#)
- clFFT\_Dimension [dim](#)
- clFFT\_DataFormat [format](#)
- string \* [kernel\\_string](#)
- cl\_program [program](#)
- cl\_fft\_kernel\_info \* [kernel\\_info](#)
- int [num\\_kernels](#)
- cl\_kernel [twist\\_kernel](#)
- cl\_int [temp\\_buffer\\_needed](#)
- unsigned [last\\_batch\\_size](#)
- cl\_mem [tempmemobj](#)

- `cl_mem` [tempmemobj\\_real](#)
- `cl_mem` [tempmemobj\\_imag](#)
- `unsigned` [max\\_localmem\\_fft\\_size](#)
- `unsigned` [max\\_work\\_item\\_per\\_workgroup](#)
- `unsigned` [max\\_radix](#)
- `unsigned` [min\\_mem\\_coalesce\\_width](#)
- `unsigned` [num\\_local\\_mem\\_banks](#)

### 10.41.1 Detailed Description

Definition at line 45 of file [fft\\_internal.h](#).

### 10.41.2 Member Data Documentation

#### 10.41.2.1 context

`cl_context` `cl_fft_plan::context`

Definition at line 48 of file [fft\\_internal.h](#).

#### 10.41.2.2 dim

`clFFT_Dimension` `cl_fft_plan::dim`

Definition at line 54 of file [fft\\_internal.h](#).

#### 10.41.2.3 format

`clFFT_DataFormat` `cl_fft_plan::format`

Definition at line 57 of file [fft\\_internal.h](#).

#### 10.41.2.4 kernel\_info

`cl_fft_kernel_info*` `cl_fft_plan::kernel_info`

Definition at line 68 of file [fft\\_internal.h](#).

#### 10.41.2.5 kernel\_string

`string*` `cl_fft_plan::kernel_string`

Definition at line 61 of file [fft\\_internal.h](#).

#### 10.41.2.6 last\_batch\_size

`unsigned` `cl_fft_plan::last_batch_size`

Definition at line 93 of file [fft\\_internal.h](#).

#### 10.41.2.7 max\_localmem\_fft\_size

`unsigned` `cl_fft_plan::max_localmem_fft_size`

Definition at line 106 of file [fft\\_internal.h](#).

#### 10.41.2.8 max\_radix

`unsigned` `cl_fft_plan::max_radix`

Definition at line 114 of file [fft\\_internal.h](#).

#### 10.41.2.9 max\_work\_item\_per\_workgroup

`unsigned` `cl_fft_plan::max_work_item_per_workgroup`

Definition at line 110 of file [fft\\_internal.h](#).



#### 10.41.2.10 min\_mem\_coalesce\_width

unsigned cl\_fft\_plan::min\_mem\_coalesce\_width  
Definition at line 119 of file [fft\\_internal.h](#).

#### 10.41.2.11 n

clFFT\_Dim3 cl\_fft\_plan::n  
Definition at line 51 of file [fft\\_internal.h](#).

#### 10.41.2.12 num\_kernels

int cl\_fft\_plan::num\_kernels  
Definition at line 71 of file [fft\\_internal.h](#).

#### 10.41.2.13 num\_local\_mem\_banks

unsigned cl\_fft\_plan::num\_local\_mem\_banks  
Definition at line 124 of file [fft\\_internal.h](#).

#### 10.41.2.14 program

cl\_program cl\_fft\_plan::program  
Definition at line 65 of file [fft\\_internal.h](#).

#### 10.41.2.15 temp\_buffer\_needed

cl\_int cl\_fft\_plan::temp\_buffer\_needed  
Definition at line 84 of file [fft\\_internal.h](#).

#### 10.41.2.16 tempmemobj

cl\_mem cl\_fft\_plan::tempmemobj  
Definition at line 96 of file [fft\\_internal.h](#).

#### 10.41.2.17 tempmemobj\_imag

cl\_mem cl\_fft\_plan::tempmemobj\_imag  
Definition at line 101 of file [fft\\_internal.h](#).

#### 10.41.2.18 tempmemobj\_real

cl\_mem cl\_fft\_plan::tempmemobj\_real  
Definition at line 101 of file [fft\\_internal.h](#).

#### 10.41.2.19 twist\_kernel

cl\_kernel cl\_fft\_plan::twist\_kernel  
Definition at line 75 of file [fft\\_internal.h](#).

The documentation for this struct was generated from the following file:

- [fft\\_internal.h](#)

## 10.42 cIFFT\_Complex Struct Reference

Collaboration diagram for cIFFT\_Complex:

cIFFT_Complex	
+	real
+	imag

### Public Attributes

- float [real](#)
- float [imag](#)

### 10.42.1 Detailed Description

Definition at line 65 of file [cIFFT.h](#).

### 10.42.2 Member Data Documentation

#### 10.42.2.1 imag

`float cIFFT_Complex::imag`

Definition at line 68 of file [cIFFT.h](#).

#### 10.42.2.2 real

`float cIFFT_Complex::real`

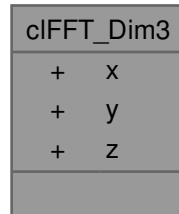
Definition at line 67 of file [cIFFT.h](#).

The documentation for this struct was generated from the following file:

- [cIFFT.h](#)

## 10.43 cIFFT\_Dim3 Struct Reference

Collaboration diagram for cIFFT\_Dim3:



### Public Attributes

- unsigned int [x](#)
- unsigned int [y](#)
- unsigned int [z](#)

### 10.43.1 Detailed Description

Definition at line [52](#) of file [cIFFT.h](#).

### 10.43.2 Member Data Documentation

#### 10.43.2.1 [x](#)

`unsigned int cIFFT_Dim3::x`

Definition at line [54](#) of file [cIFFT.h](#).

#### 10.43.2.2 [y](#)

`unsigned int cIFFT_Dim3::y`

Definition at line [55](#) of file [cIFFT.h](#).

#### 10.43.2.3 [z](#)

`unsigned int cIFFT_Dim3::z`

Definition at line [56](#) of file [cIFFT.h](#).

The documentation for this struct was generated from the following file:

- [cIFFT.h](#)

## 10.44 cIFFT\_SplitComplex Struct Reference

Collaboration diagram for cIFFT\_SplitComplex:

cIFFT_SplitComplex	
+	real
+	imag

### Public Attributes

- float \* [real](#)
- float \* [imag](#)

### 10.44.1 Detailed Description

Definition at line 59 of file [cIFFT.h](#).

### 10.44.2 Member Data Documentation

#### 10.44.2.1 imag

float\* cIFFT\_SplitComplex::imag

Definition at line 62 of file [cIFFT.h](#).

#### 10.44.2.2 real

float\* cIFFT\_SplitComplex::real

Definition at line 61 of file [cIFFT.h](#).

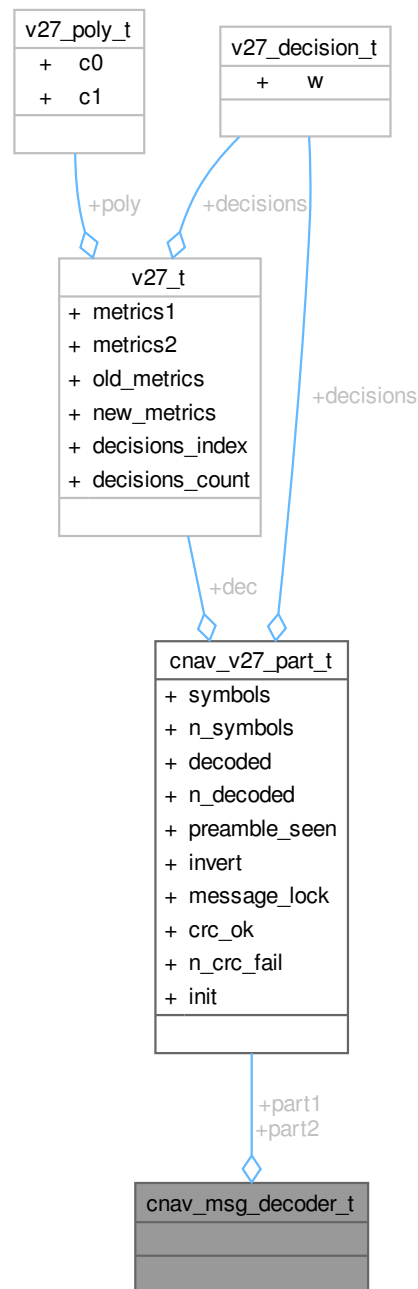
The documentation for this struct was generated from the following file:

- [cIFFT.h](#)

## 10.45 cnav\_msg\_decoder\_t Struct Reference

```
#include <cnav_msg.h>
```

Collaboration diagram for cnav\_msg\_decoder\_t:



### Public Attributes

- [cnav\\_v27\\_part\\_t part1](#)
- [cnav\\_v27\\_part\\_t part2](#)

### 10.45.1 Detailed Description

GPS CNAV message lock and decoder object.

Decoder uses two Viterbi decoder objects to ensure the lock is acquired when the input symbol phase is not known.

Definition at line 95 of file [cnav\\_msg.h](#).

## 10.45.2 Member Data Documentation

### 10.45.2.1 part1

[cnav\\_v27\\_part\\_t](#) [cnav\\_msg\\_decoder\\_t::part1](#)

Decoder for odd symbol pairs

Definition at line 97 of file [cnav\\_msg.h](#).

### 10.45.2.2 part2

[cnav\\_v27\\_part\\_t](#) [cnav\\_msg\\_decoder\\_t::part2](#)

Decoder for even symbol pairs

Definition at line 98 of file [cnav\\_msg.h](#).

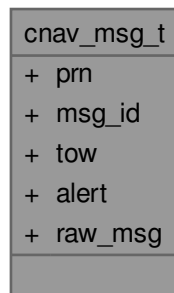
The documentation for this struct was generated from the following file:

- [cnav\\_msg.h](#)

## 10.46 cnav\_msg\_t Struct Reference

```
#include <cnav_msg.h>
```

Collaboration diagram for [cnav\\_msg\\_t](#):



### Public Attributes

- [uint8\\_t](#) [prn](#)
- [uint8\\_t](#) [msg\\_id](#)
- [uint32\\_t](#) [tow](#)
- [bool](#) [alert](#)
- [uint8\\_t](#) [raw\\_msg](#) [[GPS\\_L2C\\_V27\\_DECODE\\_BITS](#)+[GPS\\_L2C\\_V27\\_DELAY\\_BITS](#)]

### 10.46.1 Detailed Description

GPS CNAV message container.

See also

[cnav\\_msg\\_decoder\\_add\\_symbol](#)

Definition at line 52 of file [cnav\\_msg.h](#).

## 10.46.2 Member Data Documentation

### 10.46.2.1 alert

`bool cnav_msg_t::alert`  
CNAV message alert flag  
Definition at line 57 of file [cnav\\_msg.h](#).

### 10.46.2.2 msg\_id

`uint8_t cnav_msg_t::msg_id`  
Message id. 0..31  
Definition at line 55 of file [cnav\\_msg.h](#).

### 10.46.2.3 prn

`uint8_t cnav_msg_t::prn`  
SV PRN. 0..31  
Definition at line 54 of file [cnav\\_msg.h](#).

### 10.46.2.4 raw\_msg

`uint8_t cnav_msg_t::raw_msg[GPS_L2C_V27_DECODE_BITS+GPS_L2C_V27_DELAY_BITS]`  
RAW MSG for GNSS-SDR  
Definition at line 58 of file [cnav\\_msg.h](#).

### 10.46.2.5 tow

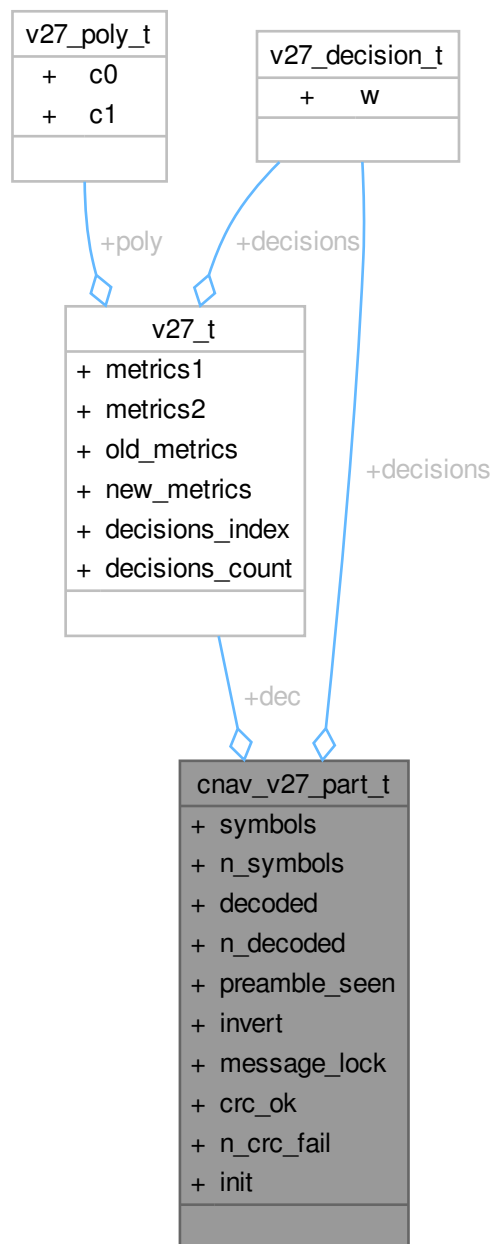
`uint32_t cnav_msg_t::tow`  
GPS ToW in 6-second units. Multiply to 6 to get seconds.  
Definition at line 56 of file [cnav\\_msg.h](#).  
The documentation for this struct was generated from the following file:

- [cnav\\_msg.h](#)

## 10.47 cnav\_v27\_part\_t Struct Reference

```
#include <cnav_msg.h>
```

Collaboration diagram for `cnav_v27_part_t`:



## Public Attributes

- [v27\\_t dec](#)
- [v27\\_decision\\_t decisions](#) [[GPS\\_L2\\_V27\\_HISTORY\\_LENGTH\\_BITS](#)]
- unsigned char [symbols](#) [([GPS\\_L2C\\_V27\\_INIT\\_BITS](#)+[GPS\\_L2C\\_V27\\_DECODE\\_BITS](#)) \*2]
- size\_t [n\\_symbols](#)
- unsigned char [decoded](#) [[GPS\\_L2C\\_V27\\_DECODE\\_BITS](#)+[GPS\\_L2C\\_V27\\_DELAY\\_BITS](#)]
- size\_t [n\\_decoded](#)
- bool [preamble\\_seen](#)



- bool [invert](#)
- bool [message\\_lock](#)
- bool [crc\\_ok](#)
- size\_t [n\\_crc\\_fail](#)
- bool [init](#)

### 10.47.1 Detailed Description

GPS CNAV decoder component. This component controls symbol decoding string.

See also

[cnav\\_msg\\_decoder\\_t](#)

Definition at line 67 of file [cnav\\_msg.h](#).

### 10.47.2 Member Data Documentation

#### 10.47.2.1 [crc\\_ok](#)

bool [cnav\\_v27\\_part\\_t::crc\\_ok](#)

Flag that the last message had good CRC

Definition at line 83 of file [cnav\\_msg.h](#).

#### 10.47.2.2 [dec](#)

[v27\\_t](#) [cnav\\_v27\\_part\\_t::dec](#)

Viterbi block decoder object

Definition at line 69 of file [cnav\\_msg.h](#).

#### 10.47.2.3 [decisions](#)

[v27\\_decision\\_t](#) [cnav\\_v27\\_part\\_t::decisions](#)[[GPS\\_L2\\_V27\\_HISTORY\\_LENGTH\\_BITS](#)]

Decision graph

Definition at line 70 of file [cnav\\_msg.h](#).

#### 10.47.2.4 [decoded](#)

unsigned char [cnav\\_v27\\_part\\_t::decoded](#)[[GPS\\_L2C\\_V27\\_DECODE\\_BITS](#)+[GPS\\_L2C\\_V27\\_DELAY\\_BITS](#)]

Decode buffer

Definition at line 75 of file [cnav\\_msg.h](#).

#### 10.47.2.5 [init](#)

bool [cnav\\_v27\\_part\\_t::init](#)

Initial state flag. When true, initial bits do not produce output.

Definition at line 85 of file [cnav\\_msg.h](#).

#### 10.47.2.6 [invert](#)

bool [cnav\\_v27\\_part\\_t::invert](#)

When true, indicates the bits are inverted

Definition at line 80 of file [cnav\\_msg.h](#).

#### 10.47.2.7 [message\\_lock](#)

bool [cnav\\_v27\\_part\\_t::message\\_lock](#)

When true, indicates the message boundary is found.

Definition at line 81 of file [cnav\\_msg.h](#).

**10.47.2.8 n\_crc\_fail**

```
size_t cnav_v27_part_t::n_crc_fail
```

Counter for CRC failures

Definition at line 84 of file [cnav\\_msg.h](#).

**10.47.2.9 n\_decoded**

```
size_t cnav_v27_part_t::n_decoded
```

Number of bits in the decode buffer

Definition at line 77 of file [cnav\\_msg.h](#).

**10.47.2.10 n\_symbols**

```
size_t cnav_v27_part_t::n_symbols
```

Count of symbols in the symbol buffer

Definition at line 74 of file [cnav\\_msg.h](#).

**10.47.2.11 preamble\_seen**

```
bool cnav_v27_part_t::preamble_seen
```

When true, the decode buffer is aligned on preamble.

Definition at line 78 of file [cnav\\_msg.h](#).

**10.47.2.12 symbols**

```
unsigned char cnav_v27_part_t::symbols[(GPS_L2C_V27_INIT_BITS+GPS_L2C_V27_DECODE_BITS) *2]
```

Symbol buffer

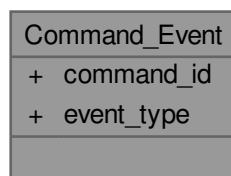
Definition at line 72 of file [cnav\\_msg.h](#).

The documentation for this struct was generated from the following file:

- [cnav\\_msg.h](#)

**10.48 Command\_Event Class Reference**

Collaboration diagram for Command\_Event:

**Public Attributes**

- int [command\\_id](#)
- int [event\\_type](#)

**Friends**

- `command_event_sptr` **command\_event\_make** (int command\_id, int event\_type)

### 10.48.1 Detailed Description

Definition at line 34 of file [command\\_event.h](#).

### 10.48.2 Member Data Documentation

#### 10.48.2.1 command\_id

```
int Command_Event::command_id
```

Definition at line 37 of file [command\\_event.h](#).

#### 10.48.2.2 event\_type

```
int Command_Event::event_type
```

Definition at line 38 of file [command\\_event.h](#).

The documentation for this class was generated from the following file:

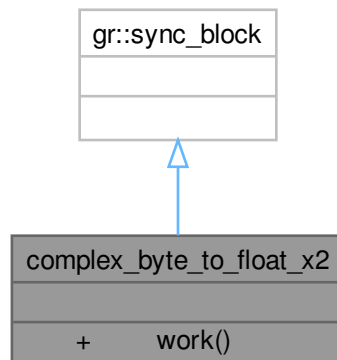
- [command\\_event.h](#)

## 10.49 complex\_byte\_to\_float\_x2 Class Reference

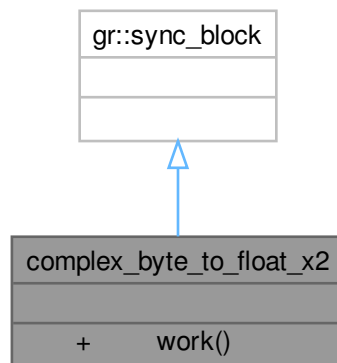
This class adapts a `std::complex<signed char>` stream into two 16-bits (short) streams.

```
#include <complex_byte_to_float_x2.h>
```

Inheritance diagram for `complex_byte_to_float_x2`:



Collaboration diagram for `complex_byte_to_float_x2`:



#### Public Member Functions

- `int` **work** (`int` noutput\_items, `gr_vector_const_void_star` &input\_items, `gr_vector_void_star` &output\_items)

#### Friends

- `complex_byte_to_float_x2_sptr` **make\_complex\_byte\_to\_float\_x2** ()

#### 10.49.1 Detailed Description

This class adapts a `std::complex<signed char>` stream into two 16-bits (short) streams. Definition at line 41 of file [complex\\_byte\\_to\\_float\\_x2.h](#).

The documentation for this class was generated from the following file:

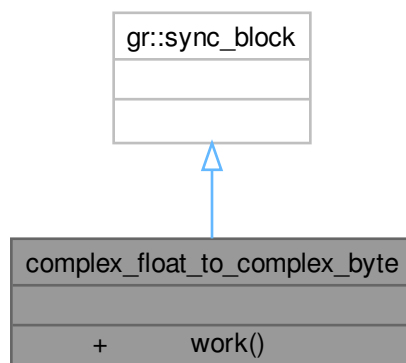
- [complex\\_byte\\_to\\_float\\_x2.h](#)

### 10.50 `complex_float_to_complex_byte` Class Reference

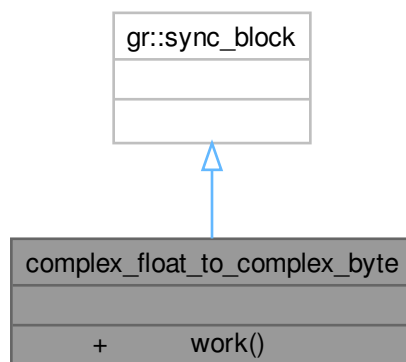
This class adapts a `gr_complex` stream into a `std::complex<signed char>` stream.

```
#include <complex_float_to_complex_byte.h>
```

Inheritance diagram for complex\_float\_to\_complex\_byte:



Collaboration diagram for complex\_float\_to\_complex\_byte:



### Public Member Functions

- `int` **work** (`int` noutput\_items, `gr_vector_const_void_star` &input\_items, `gr_vector_void_star` &output\_items)

### Friends

- `complex_float_to_complex_byte_sptr` **make\_complex\_float\_to\_complex\_byte** ()

### 10.50.1 Detailed Description

This class adapts a `gr_complex` stream into a `std::complex<signed char>` stream.

Definition at line 40 of file [complex\\_float\\_to\\_complex\\_byte.h](#).

The documentation for this class was generated from the following file:

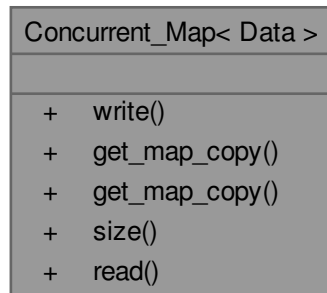
- [complex\\_float\\_to\\_complex\\_byte.h](#)

## 10.51 Concurrent\_Map< Data > Class Template Reference

This class implements a thread-safe `std::map`.

```
#include <concurrent_map.h>
```

Collaboration diagram for `Concurrent_Map< Data >`:



### Public Member Functions

- void [write](#) (int key, Data const &data)
- `std::map< int, Data >` [get\\_map\\_copy](#) () const &
- `std::map< int, Data >` [get\\_map\\_copy](#) () &&
- `size_t` [size](#) () const
- bool [read](#) (int key, Data &p\_data) const

### 10.51.1 Detailed Description

```
template<typename Data>
class Concurrent_Map< Data >
```

This class implements a thread-safe `std::map`.

Definition at line 37 of file [concurrent\\_map.h](#).

### 10.51.2 Member Function Documentation

#### 10.51.2.1 [get\\_map\\_copy\(\)](#) [1/2]

```
template<typename Data>
std::map< int, Data > Concurrent_Map< Data >::get_map_copy () && [inline]
Definition at line 60 of file concurrent\_map.h.
```

#### 10.51.2.2 [get\\_map\\_copy\(\)](#) [2/2]

```
template<typename Data>
std::map< int, Data > Concurrent_Map< Data >::get_map_copy () const & [inline]
Definition at line 54 of file concurrent\_map.h.
```

#### 10.51.2.3 [read\(\)](#)

```
template<typename Data>
bool Concurrent_Map< Data >::read (
    int key,
    Data & p_data) const [inline]
```

Definition at line 72 of file [concurrent\\_map.h](#).

#### 10.51.2.4 size()

```
template<typename Data>
size_t Concurrent_Map< Data >::size () const [inline]
Definition at line 66 of file concurrent\_map.h.
```

#### 10.51.2.5 write()

```
template<typename Data>
void Concurrent_Map< Data >::write (
    int key,
    Data const & data) [inline]
Definition at line 40 of file concurrent\_map.h.
```

The documentation for this class was generated from the following file:

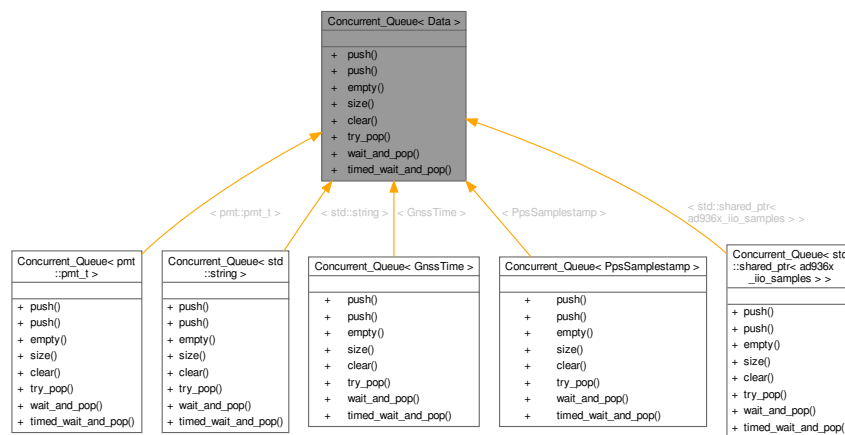
- [concurrent\\_map.h](#)

## 10.52 Concurrent\_Queue< Data > Class Template Reference

This class implements a thread-safe `std::queue`.

```
#include <concurrent_queue.h>
```

Inheritance diagram for `Concurrent_Queue< Data >`:



Collaboration diagram for `Concurrent_Queue< Data >`:

Concurrent_Queue< Data >
<ul style="list-style-type: none"> <li>+ <code>push()</code></li> <li>+ <code>push()</code></li> <li>+ <code>empty()</code></li> <li>+ <code>size()</code></li> <li>+ <code>clear()</code></li> <li>+ <code>try_pop()</code></li> <li>+ <code>wait_and_pop()</code></li> <li>+ <code>timed_wait_and_pop()</code></li> </ul>

### Public Member Functions

- void `push` (const Data &data)
- void `push` (Data &&data)
- bool `empty` () const noexcept
- size\_t `size` () const noexcept
- void `clear` ()
- bool `try_pop` (Data &popped\_value)
- void `wait_and_pop` (Data &popped\_value)
- bool `timed_wait_and_pop` (Data &popped\_value, int wait\_ms)

## 10.52.1 Detailed Description

`template<typename Data>`  
**class** `Concurrent_Queue< Data >`

This class implements a thread-safe `std::queue`.  
 Definition at line 38 of file [concurrent\\_queue.h](#).

## 10.52.2 Member Function Documentation

### 10.52.2.1 `clear()`

```
template<typename Data>
void Concurrent_Queue< Data >::clear () [inline]
Definition at line 70 of file concurrent\_queue.h.
```

### 10.52.2.2 `empty()`

```
template<typename Data>
bool Concurrent_Queue< Data >::empty () const [inline], [noexcept]
Definition at line 59 of file concurrent\_queue.h.
```



**10.52.2.3 push() [1/2]**

```
template<typename Data>
void Concurrent_Queue< Data >::push (
    const Data & data) [inline]
```

Definition at line 41 of file [concurrent\\_queue.h](#).

**10.52.2.4 push() [2/2]**

```
template<typename Data>
void Concurrent_Queue< Data >::push (
    Data && data) [inline]
```

Definition at line 50 of file [concurrent\\_queue.h](#).

**10.52.2.5 size()**

```
template<typename Data>
size_t Concurrent_Queue< Data >::size () const [inline], [noexcept]
```

Definition at line 64 of file [concurrent\\_queue.h](#).

**10.52.2.6 timed\_wait\_and\_pop()**

```
template<typename Data>
bool Concurrent_Queue< Data >::timed_wait_and_pop (
    Data & popped_value,
    int wait_ms) [inline]
```

Definition at line 96 of file [concurrent\\_queue.h](#).

**10.52.2.7 try\_pop()**

```
template<typename Data>
bool Concurrent_Queue< Data >::try_pop (
    Data & popped_value) [inline]
```

Definition at line 76 of file [concurrent\\_queue.h](#).

**10.52.2.8 wait\_and\_pop()**

```
template<typename Data>
void Concurrent_Queue< Data >::wait_and_pop (
    Data & popped_value) [inline]
```

Definition at line 88 of file [concurrent\\_queue.h](#).

The documentation for this class was generated from the following files:

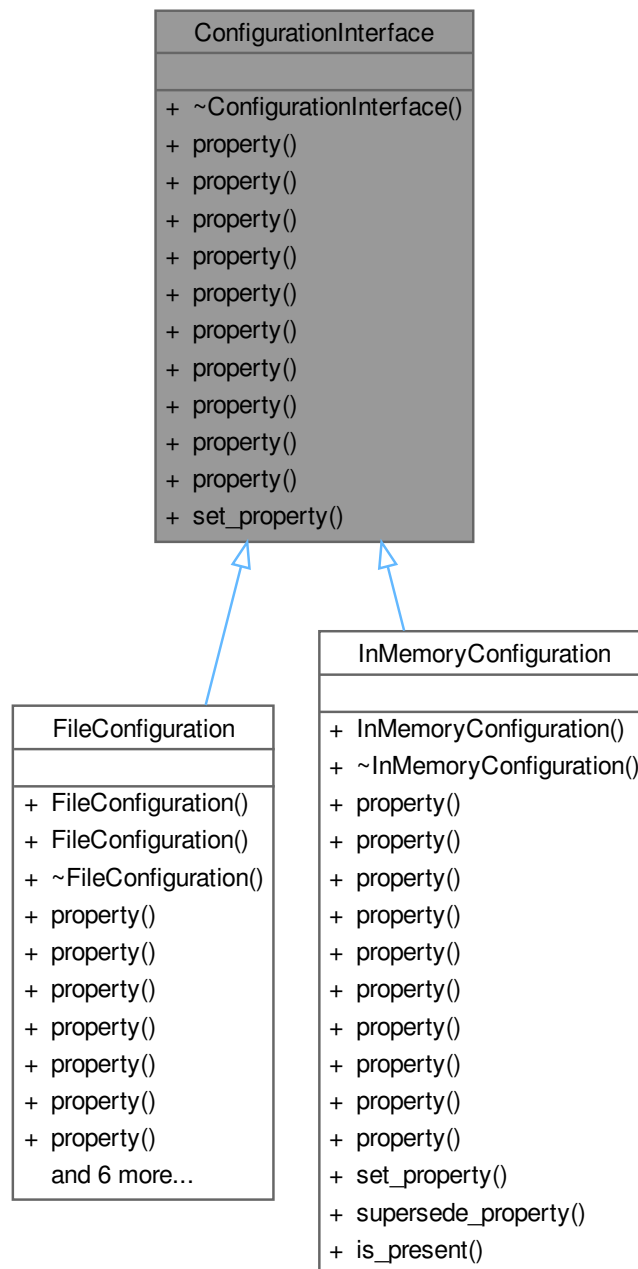
- [acquisition\\_interface.h](#)
- [concurrent\\_queue.h](#)

**10.53 ConfigurationInterface Class Reference**

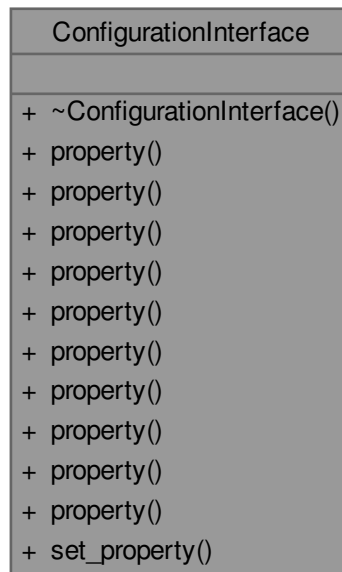
This abstract class represents an interface to configuration parameters.

```
#include <configuration_interface.h>
```

Inheritance diagram for ConfigurationInterface:



Collaboration diagram for ConfigurationInterface:



### Public Member Functions

- virtual std::string **property** (std::string property\_name, std::string default\_value) const =0
- virtual bool **property** (std::string property\_name, bool default\_value) const =0
- virtual int64\_t **property** (std::string property\_name, int64\_t default\_value) const =0
- virtual uint64\_t **property** (std::string property\_name, uint64\_t default\_value) const =0
- virtual int32\_t **property** (std::string property\_name, int32\_t default\_value) const =0
- virtual uint32\_t **property** (std::string property\_name, uint32\_t default\_value) const =0
- virtual int16\_t **property** (std::string property\_name, int16\_t default\_value) const =0
- virtual uint16\_t **property** (std::string property\_name, uint16\_t default\_value) const =0
- virtual float **property** (std::string property\_name, float default\_value) const =0
- virtual double **property** (std::string property\_name, double default\_value) const =0
- virtual void **set\_property** (std::string property\_name, std::string value)=0

### 10.53.1 Detailed Description

This abstract class represents an interface to configuration parameters.

The interface defines an accessor method that gets a parameter name as input and returns the value of this parameter, a string, as output. Property names are defined here. This is an abstract class for interfaces. Since all its methods are virtual, this class cannot be instantiated directly, and a subclass can only be instantiated directly if all inherited pure virtual methods have been implemented by that class or a parent class.

Definition at line 44 of file [configuration\\_interface.h](#).

The documentation for this class was generated from the following file:

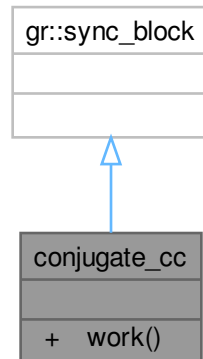
- [configuration\\_interface.h](#)

## 10.54 conjugate\_cc Class Reference

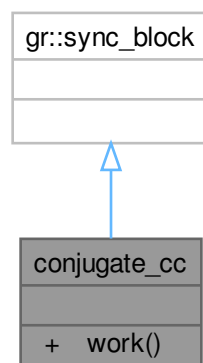
This class adapts a std::complex<short> stream into two 32-bits (float) streams.

```
#include <conjugate_cc.h>
```

Inheritance diagram for conjugate\_cc:



Collaboration diagram for conjugate\_cc:



### Public Member Functions

- `int` **work** (int noutput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)

### Friends

- `conjugate_cc_sptr` **make\_conjugate\_cc** ()

### 10.54.1 Detailed Description

This class adapts a `std::complex<short>` stream into two 32-bits (float) streams.

Definition at line 41 of file [conjugate\\_cc.h](#).

The documentation for this class was generated from the following file:

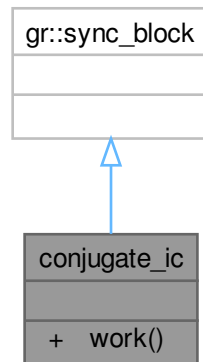
- [conjugate\\_cc.h](#)

## 10.55 conjugate\_ic Class Reference

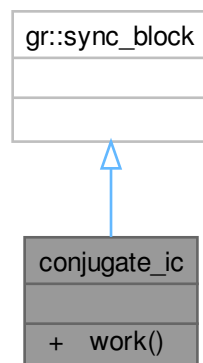
This class adapts a `std::complex<short>` stream into two 32-bits (float) streams.

```
#include <conjugate_ic.h>
```

Inheritance diagram for `conjugate_ic`:



Collaboration diagram for `conjugate_ic`:



### Public Member Functions

- `int` **work** (`int` noutput\_items, `gr_vector_const_void_star` &input\_items, `gr_vector_void_star` &output\_items)

### Friends

- `conjugate_ic_sptr` **make\_conjugate\_ic** ()

### 10.55.1 Detailed Description

This class adapts a `std::complex<short>` stream into two 32-bits (float) streams.

Definition at line 41 of file [conjugate\\_ic.h](#).

The documentation for this class was generated from the following file:

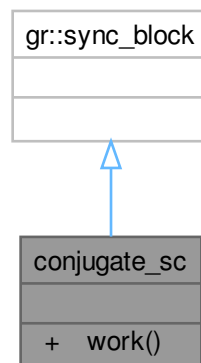
- [conjugate\\_ic.h](#)

## 10.56 conjugate\_sc Class Reference

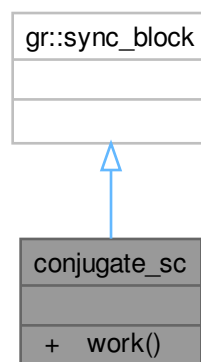
This class adapts a `std::complex<short>` stream into two 32-bits (float) streams.

```
#include <conjugate_sc.h>
```

Inheritance diagram for `conjugate_sc`:



Collaboration diagram for `conjugate_sc`:



### Public Member Functions

- `int` **work** (`int` noutput\_items, `gr_vector_const_void_star` &input\_items, `gr_vector_void_star` &output\_items)

**Friends**

- `conjugate_sc_sptr make_conjugate_sc ()`

**10.56.1 Detailed Description**

This class adapts a `std::complex<short>` stream into two 32-bits (float) streams.

Definition at line 41 of file [conjugate\\_sc.h](#).

The documentation for this class was generated from the following file:

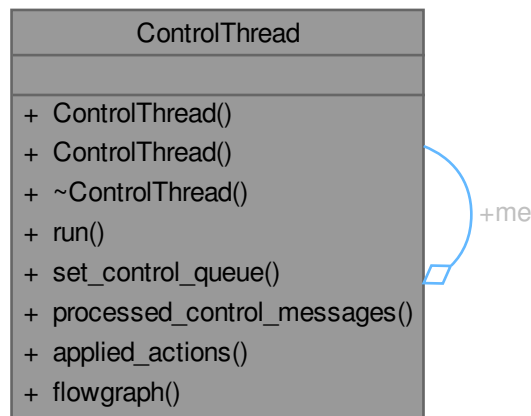
- [conjugate\\_sc.h](#)

**10.57 ControlThread Class Reference**

This class represents the main thread of the application, so the name is [ControlThread](#). This is the GNSS Receiver Control Plane: it connects the flowgraph, starts running it, and while it does not stop, reads the control messages generated by the blocks, processes them, and applies the corresponding actions.

```
#include <control_thread.h>
```

Collaboration diagram for ControlThread:

**Public Member Functions**

- [ControlThread](#) ()  
*Default constructor.*
- [ControlThread](#) (std::shared\_ptr< [ConfigurationInterface](#) > configuration)  
*Constructor that initializes the class with parameters.*
- [~ControlThread](#) ()  
*Destructor.*
- int [run](#) ()  
*Runs the control thread.*
- void [set\\_control\\_queue](#) (std::shared\_ptr< [Concurrent\\_Queue](#)< pmt::pmt\_t > > control\_queue)  
*Sets the control\_queue.*
- unsigned int [processed\\_control\\_messages](#) () const
- unsigned int [applied\\_actions](#) () const
- std::shared\_ptr< [GNSSFlowgraph](#) > [flowgraph](#) ()  
*Instantiates a flowgraph.*

## Static Public Attributes

- static [ControlThread](#) \* [me](#)

### 10.57.1 Detailed Description

This class represents the main thread of the application, so the name is [ControlThread](#). This is the GNSS Receiver Control Plane: it connects the flowgraph, starts running it, and while it does not stop, reads the control messages generated by the blocks, processes them, and applies the corresponding actions.

Definition at line 64 of file [control\\_thread.h](#).

### 10.57.2 Constructor & Destructor Documentation

#### 10.57.2.1 [ControlThread\(\)](#) [1/2]

```
ControlThread::ControlThread ()
```

Default constructor.

#### 10.57.2.2 [ControlThread\(\)](#) [2/2]

```
ControlThread::ControlThread (
    std::shared_ptr< ConfigurationInterface > configuration) [explicit]
```

Constructor that initializes the class with parameters.

#### Parameters

in	<i>configuration</i>	Pointer to a <a href="#">ConfigurationInterface</a>
----	----------------------	---

#### 10.57.2.3 [~ControlThread\(\)](#)

```
ControlThread::~~ControlThread ()
```

Destructor.

### 10.57.3 Member Function Documentation

#### 10.57.3.1 [applied\\_actions\(\)](#)

```
unsigned int ControlThread::applied_actions () const [inline]
```

Definition at line 111 of file [control\\_thread.h](#).

#### 10.57.3.2 [flowgraph\(\)](#)

```
std::shared_ptr< GNSSFlowgraph > ControlThread::flowgraph () [inline]
```

Instantiates a flowgraph.

#### Returns

Returns a smart pointer to a flowgraph object

Definition at line 121 of file [control\\_thread.h](#).

#### 10.57.3.3 [processed\\_control\\_messages\(\)](#)

```
unsigned int ControlThread::processed_control_messages () const [inline]
```

Definition at line 106 of file [control\\_thread.h](#).



### 10.57.3.4 run()

```
int ControlThread::run ()
```

Runs the control thread.

This is the main loop that reads and process the control messages:

- Connect the GNSS receiver flowgraph;
- Start the GNSS receiver flowgraph;

```
while (flowgraph_->running() && !stop_){
```

- Read control messages and process them; }

### 10.57.3.5 set\_control\_queue()

```
void ControlThread::set_control_queue (
    std::shared_ptr< Concurrent_Queue< pmt::pmt_t > > control_queue)
```

Sets the control\_queue.

Parameters

in	<code>std::shared_ptr&lt;Concurrent_Queue&lt;pmt::pmt_t&gt;&gt;</code>	control_queue
----	--	---------------

## 10.57.4 Member Data Documentation

### 10.57.4.1 me

```
ControlThread* ControlThread::me [static]
```

Definition at line 67 of file [control\\_thread.h](#).

The documentation for this class was generated from the following file:

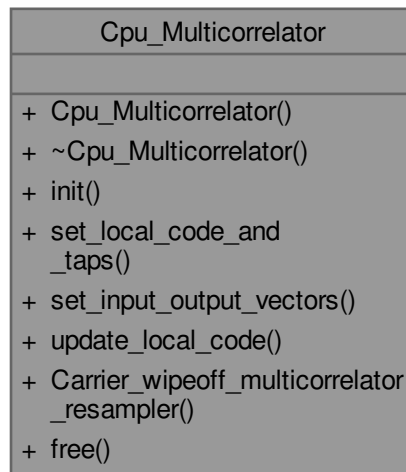
- [control\\_thread.h](#)

## 10.58 Cpu\_Multicorrelator Class Reference

Class that implements carrier wipe-off and correlators.

```
#include <cpu_multicorrelator.h>
```

Collaboration diagram for Cpu\_Multicorrelator:



### Public Member Functions

- bool **init** (int max\_signal\_length\_samples, int n\_correlators)
- bool **set\_local\_code\_and\_taps** (int code\_length\_chips, const std::complex< float > \*local\_code\_in, float \*shifts\_chips)
- bool **set\_input\_output\_vectors** (std::complex< float > \*corr\_out, const std::complex< float > \*sig\_in)
- void **update\_local\_code** (int correlator\_length\_samples, float rem\_code\_phase\_chips, float code\_phase\_↔ step\_chips)
- bool **Carrier\_wipeoff\_multicorrelator\_resampler** (float rem\_carrier\_phase\_in\_rad, float phase\_step\_rad, float rem\_code\_phase\_chips, float code\_phase\_step\_chips, int signal\_length\_samples)
- bool **free** ()

### 10.58.1 Detailed Description

Class that implements carrier wipe-off and correlators.

Definition at line 37 of file [cpu\\_multicorrelator.h](#).

The documentation for this class was generated from the following file:

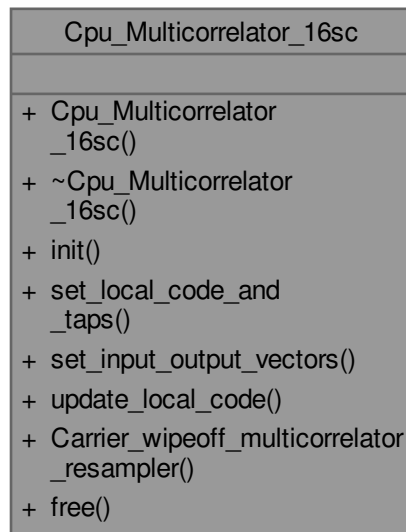
- [cpu\\_multicorrelator.h](#)

## 10.59 Cpu\_Multicorrelator\_16sc Class Reference

Class that implements carrier wipe-off and correlators.

#include <cpu\_multicorrelator\_16sc.h>

Collaboration diagram for Cpu\_Multicorrelator\_16sc:



### Public Member Functions

- bool **init** (int max\_signal\_length\_samples, int n\_correlators)
- bool **set\_local\_code\_and\_taps** (int code\_length\_chips, const lv\_16sc\_t \*local\_code\_in, float \*shifts\_chips)
- bool **set\_input\_output\_vectors** (lv\_16sc\_t \*corr\_out, const lv\_16sc\_t \*sig\_in)
- void **update\_local\_code** (int correlator\_length\_samples, float rem\_code\_phase\_chips, float code\_phase\_↔ step\_chips)
- bool **Carrier\_wipeoff\_multicorrelator\_resampler** (float rem\_carrier\_phase\_in\_rad, float phase\_step\_rad, float rem\_code\_phase\_chips, float code\_phase\_step\_chips, int signal\_length\_samples)
- bool **free** ()

### 10.59.1 Detailed Description

Class that implements carrier wipe-off and correlators.

Definition at line 35 of file [cpu\\_multicorrelator\\_16sc.h](#).

The documentation for this class was generated from the following file:

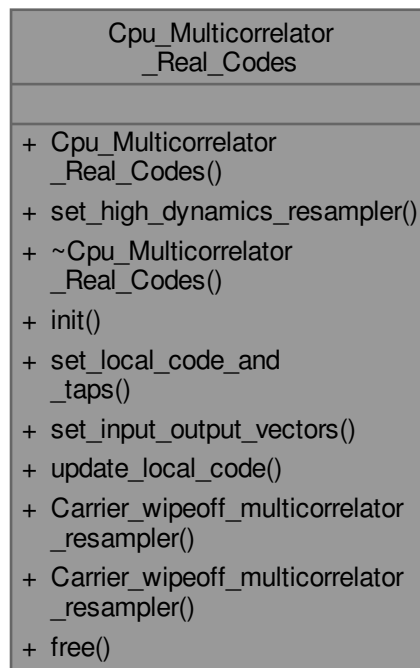
- [cpu\\_multicorrelator\\_16sc.h](#)

## 10.60 Cpu\_Multicorrelator\_Real\_Codes Class Reference

Class that implements carrier wipe-off and correlators.

#include <cpu\_multicorrelator\_real\_codes.h>

Collaboration diagram for Cpu\_Multicorrelator\_Real\_Codes:



### Public Member Functions

- void **set\_high\_dynamics\_resampler** (bool use\_high\_dynamics\_resampler)
- bool **init** (int max\_signal\_length\_samples, int n\_correlators)
- bool **set\_local\_code\_and\_taps** (int code\_length\_chips, const float \*local\_code\_in, float \*shifts\_chips)
- bool **set\_input\_output\_vectors** (std::complex< float > \*corr\_out, const std::complex< float > \*sig\_in)
- void **update\_local\_code** (int correlator\_length\_samples, float rem\_code\_phase\_chips, float code\_phase\_step\_chips, float code\_phase\_rate\_step\_chips=0.0)
- bool **Carrier\_wipeoff\_multicorrelator\_resampler** (float rem\_carrier\_phase\_in\_rad, float phase\_step\_rad, float phase\_rate\_step\_rad, float rem\_code\_phase\_chips, float code\_phase\_step\_chips, float code\_phase\_rate\_step\_chips, int signal\_length\_samples)
- bool **Carrier\_wipeoff\_multicorrelator\_resampler** (float rem\_carrier\_phase\_in\_rad, float phase\_step\_rad, float rem\_code\_phase\_chips, float code\_phase\_step\_chips, float code\_phase\_rate\_step\_chips, int signal\_length\_samples)
- bool **free** ()

#### 10.60.1 Detailed Description

Class that implements carrier wipe-off and correlators.

Definition at line 37 of file [cpu\\_multicorrelator\\_real\\_codes.h](#).

The documentation for this class was generated from the following file:

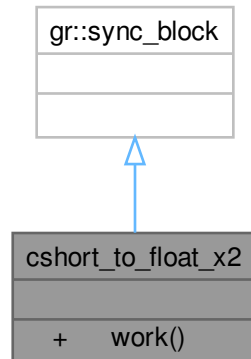
- [cpu\\_multicorrelator\\_real\\_codes.h](#)

## 10.61 cshort\_to\_float\_x2 Class Reference

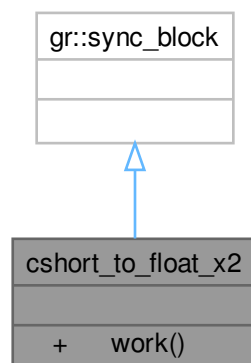
This class adapts a std::complex<short> stream into two 32-bits (float) streams.

```
#include <csort_to_float_x2.h>
```

Inheritance diagram for cshort\_to\_float\_x2:



Collaboration diagram for cshort\_to\_float\_x2:



### Public Member Functions

- `int` **work** (`int` noutput\_items, `gr_vector_const_void_star` &input\_items, `gr_vector_void_star` &output\_items)

### Friends

- `csort_to_float_x2_sptr` **make\_csort\_to\_float\_x2** ()

### 10.61.1 Detailed Description

This class adapts a `std::complex<short>` stream into two 32-bits (float) streams.

Definition at line 41 of file [csort\\_to\\_float\\_x2.h](#).

The documentation for this class was generated from the following file:

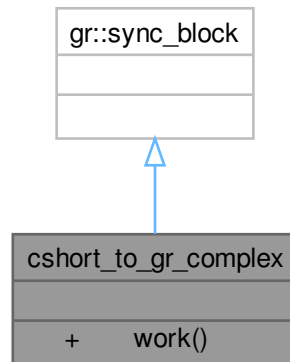
- [csort\\_to\\_float\\_x2.h](#)

## 10.62 cshort\_to\_gr\_complex Class Reference

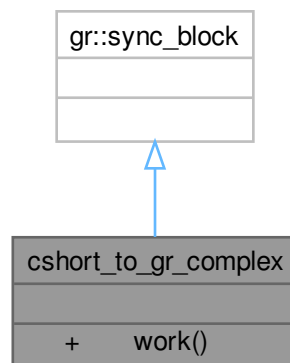
This class adapts a short (16-bits) interleaved sample stream into a `std::complex<float>` stream.

```
#include <csort_to_gr_complex.h>
```

Inheritance diagram for `csort_to_gr_complex`:



Collaboration diagram for `csort_to_gr_complex`:



### Public Member Functions

- `int` **work** (`int` noutput\_items, `gr_vector_const_void_star` &input\_items, `gr_vector_void_star` &output\_items)

### Friends

- `csort_to_gr_complex_sptr` **make\_csort\_to\_gr\_complex** ()

### 10.62.1 Detailed Description

This class adapts a short (16-bits) interleaved sample stream into a `std::complex<float>` stream.

Definition at line 40 of file [cshort\\_to\\_gr\\_complex.h](#).

The documentation for this class was generated from the following file:

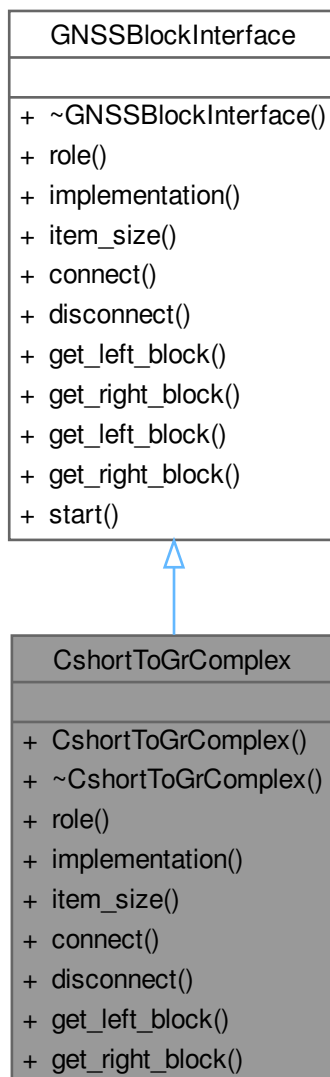
- [cshort\\_to\\_gr\\_complex.h](#)

## 10.63 CshortToGrComplex Class Reference

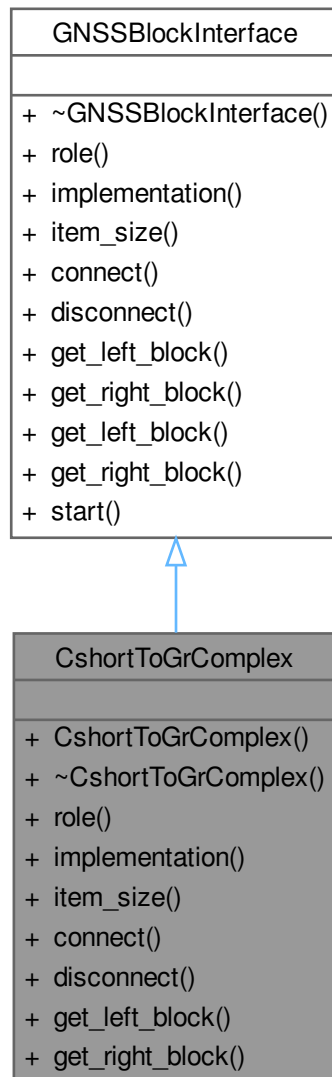
Adapts an 16-bits complex sample stream to a float complex stream.

```
#include <cshort_to_grcomplex.h>
```

Inheritance diagram for CshortToGrComplex:



Collaboration diagram for CshortToGrComplex:



### Public Member Functions

- **CshortToGrComplex** (const [ConfigurationInterface](#) \*configuration, std::string role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "Cshort\_To\_Gr\_Complex".
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override



## Public Member Functions inherited from GNSSBlockInterface

- virtual `gr::basic_block_sptr` [get\\_left\\_block](#) (int RF\_channel)
- virtual `gr::basic_block_sptr` [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.63.1 Detailed Description

Adapts an 16-bits complex sample stream to a float complex stream.

Definition at line 40 of file [cshort\\_to\\_grcomplex.h](#).

### 10.63.2 Member Function Documentation

#### 10.63.2.1 connect()

```
void CshortToGrComplex::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.63.2.2 disconnect()

```
void CshortToGrComplex::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.63.2.3 get\_left\_block()

```
gr::basic_block_sptr CshortToGrComplex::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.63.2.4 get\_right\_block()

```
gr::basic_block_sptr CshortToGrComplex::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.63.2.5 implementation()

```
std::string CshortToGrComplex::implementation () [inline], [override], [virtual]
```

Returns "Cshort\_To\_Gr\_Complex".

Implements [GNSSBlockInterface](#).

Definition at line 55 of file [cshort\\_to\\_grcomplex.h](#).

#### 10.63.2.6 item\_size()

```
size_t CshortToGrComplex::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 60 of file [cshort\\_to\\_grcomplex.h](#).

#### 10.63.2.7 role()

```
std::string CshortToGrComplex::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 49 of file [cshort\\_to\\_grcomplex.h](#).

The documentation for this class was generated from the following file:

- [cshort\\_to\\_grcomplex.h](#)

## 10.64 CubatureFilter Class Reference

Collaboration diagram for CubatureFilter:

CubatureFilter
<ul style="list-style-type: none"> <li>+ CubatureFilter()</li> <li>+ CubatureFilter()</li> <li>+ CubatureFilter()</li> <li>+ ~CubatureFilter()</li> <li>+ initialize()</li> <li>+ predict_sequential()</li> <li>+ update_sequential()</li> <li>+ get_x_pred()</li> <li>+ get_P_x_pred()</li> <li>+ get_x_est()</li> <li>+ get_P_x_est()</li> </ul>

### Public Member Functions

- **CubatureFilter** (int nx)
- **CubatureFilter** (const arma::vec &x\_pred\_0, const arma::mat &P\_x\_pred\_0)
- void **initialize** (const arma::mat &x\_pred\_0, const arma::mat &P\_x\_pred\_0)
- void **predict\_sequential** (const arma::vec &x\_post, const arma::mat &P\_x\_post, [ModelFunction](#) \*transition\_fcn, const arma::mat &noise\_covariance)
- void **update\_sequential** (const arma::vec &z\_upd, const arma::vec &x\_pred, const arma::mat &P\_x\_pred, [ModelFunction](#) \*measurement\_fcn, const arma::mat &noise\_covariance)
- arma::mat **get\_x\_pred** () const
- arma::mat **get\_P\_x\_pred** () const
- arma::mat **get\_x\_est** () const
- arma::mat **get\_P\_x\_est** () const

### 10.64.1 Detailed Description

Definition at line 54 of file [nonlinear\\_tracking.h](#).

The documentation for this class was generated from the following file:

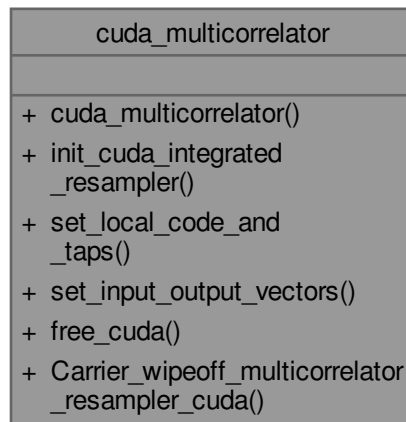
- [nonlinear\\_tracking.h](#)

## 10.65 cuda\_multicorrelator Class Reference

Class that implements carrier wipe-off and correlators using NVIDIA CUDA GPU accelerators.

```
#include <cuda_multicorrelator.h>
```

Collaboration diagram for `cuda_multicorrelator`:



### Public Member Functions

- `bool init_cuda_integrated_resampler` (int signal\_length\_samples, int code\_length\_chips, int n\_correlators)
- `bool set_local_code_and_taps` (int code\_length\_chips, const std::complex< float > \*local\_codes\_in, float \*shifts\_chips, int n\_correlators)
- `bool set_input_output_vectors` (std::complex< float > \*corr\_out, std::complex< float > \*sig\_in)
- `bool free_cuda` ()
- `bool Carrier_wipeoff_multicorrelator_resampler_cuda` (float rem\_carrier\_phase\_in\_rad, float phase\_step\_rad, float code\_phase\_step\_chips, float rem\_code\_phase\_chips, int signal\_length\_samples, int n\_correlators)

### 10.65.1 Detailed Description

Class that implements carrier wipe-off and correlators using NVIDIA CUDA GPU accelerators.

Definition at line 111 of file [cuda\\_multicorrelator.h](#).

The documentation for this class was generated from the following file:

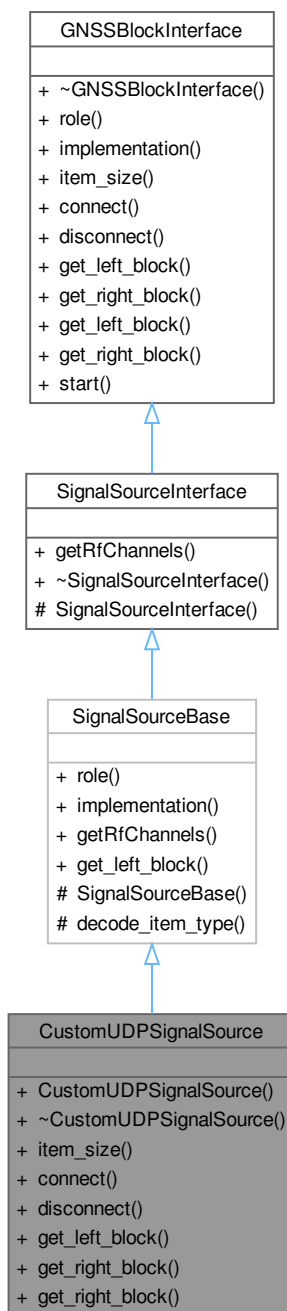
- [cuda\\_multicorrelator.h](#)

## 10.66 CustomUDPSignalSource Class Reference

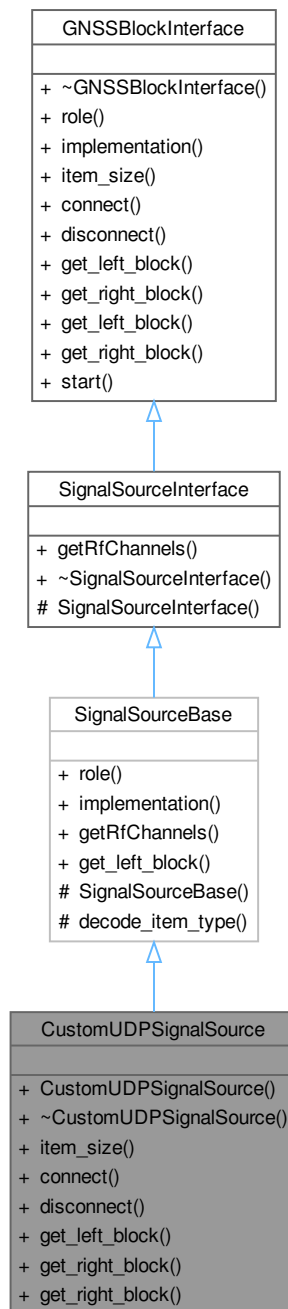
This class reads from UDP packets, which streams interleaved I/Q samples over a network.

```
#include <custom_udp_signal_source.h>
```

Inheritance diagram for CustomUDPSignalSource:



Collaboration diagram for CustomUDPSignalSource:



### Public Member Functions

- **CustomUDPSignalSource** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_stream, unsigned int out\_stream, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override

- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel) override

### Public Member Functions inherited from [SignalSourceBase](#)

- std::string [role](#) () final
- std::string [implementation](#) () final
- size\_t [getRfChannels](#) () const override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### Additional Inherited Members

### Protected Member Functions inherited from [SignalSourceBase](#)

- [SignalSourceBase](#) ([ConfigurationInterface](#) const \*configuration, std::string role, std::string impl)  
*Constructor.*
- size\_t [decode\\_item\\_type](#) (std::string const &item\_type, bool \*is\_interleaved=nullptr, bool throw\_on\_↔ error=false)  
*utility for decoding passed ".item\_type" values*

## 10.66.1 Detailed Description

This class reads from UDP packets, which streams interleaved I/Q samples over a network. Definition at line 43 of file [custom\\_udp\\_signal\\_source.h](#).

## 10.66.2 Member Function Documentation

### 10.66.2.1 connect()

```
void CustomUDPSignalSource::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.66.2.2 disconnect()

```
void CustomUDPSignalSource::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.66.2.3 get\_left\_block()

```
gr::basic_block_sptr CustomUDPSignalSource::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.66.2.4 get\_right\_block() [1/2]

```
gr::basic_block_sptr CustomUDPSignalSource::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.66.2.5 `get_right_block()` [2/2]

```
gr::basic_block_sptr CustomUDPSignalSource::get_right_block (
    int RF_channel) [override], [virtual]
```

Reimplemented from [GNSSBlockInterface](#).

### 10.66.2.6 `item_size()`

```
size_t CustomUDPSignalSource::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

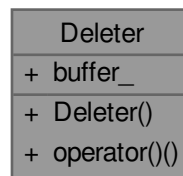
Definition at line 52 of file [custom\\_udp\\_signal\\_source.h](#).

The documentation for this class was generated from the following file:

- [custom\\_udp\\_signal\\_source.h](#)

## 10.67 Deleter Struct Reference

Collaboration diagram for Deleter:



### Public Member Functions

- [Deleter](#) (void \*buffer)
- `template<typename WordType>`  
void [operator\(\)](#) () const

### Public Attributes

- void \* [buffer\\_](#)

### 10.67.1 Detailed Description

Definition at line 74 of file [ion\\_gsms\\_chunk\\_data.h](#).

### 10.67.2 Constructor & Destructor Documentation

#### 10.67.2.1 `Deleter()`

```
Deleter::Deleter (
    void * buffer) [inline], [explicit]
```

Definition at line 78 of file [ion\\_gsms\\_chunk\\_data.h](#).

## 10.67.3 Member Function Documentation

### 10.67.3.1 operator>()

```
template<typename WordType>
void Deleter::operator() () const [inline]
Definition at line 82 of file ion\_gsms\_chunk\_data.h.
```

## 10.67.4 Member Data Documentation

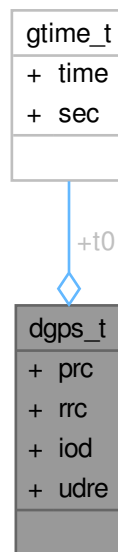
### 10.67.4.1 buffer\_

```
void* Deleter::buffer_
Definition at line 76 of file ion\_gsms\_chunk\_data.h.
The documentation for this struct was generated from the following file:
```

- [ion\\_gsms\\_chunk\\_data.h](#)

## 10.68 dgps\_t Struct Reference

Collaboration diagram for dgps\_t:



### Public Attributes

- [gtime\\_t t0](#)
- double [prc](#)
- double [rrc](#)
- int [iod](#)
- double [udre](#)

### 10.68.1 Detailed Description

Definition at line 645 of file [rtklib.h](#).



## 10.68.2 Member Data Documentation

### 10.68.2.1 iod

`int dgps_t::iod`

Definition at line 650 of file [rtklib.h](#).

### 10.68.2.2 prc

`double dgps_t::prc`

Definition at line 648 of file [rtklib.h](#).

### 10.68.2.3 rrc

`double dgps_t::rrc`

Definition at line 649 of file [rtklib.h](#).

### 10.68.2.4 t0

`ptime_t dgps_t::t0`

Definition at line 647 of file [rtklib.h](#).

### 10.68.2.5 udre

`double dgps_t::udre`

Definition at line 651 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

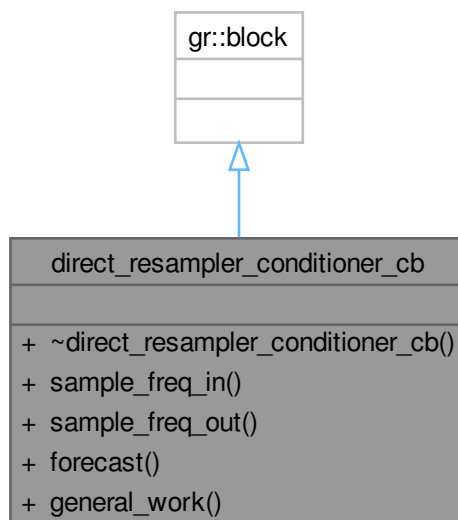
- [rtklib.h](#)

## 10.69 direct\_resampler\_conditioner\_cb Class Reference

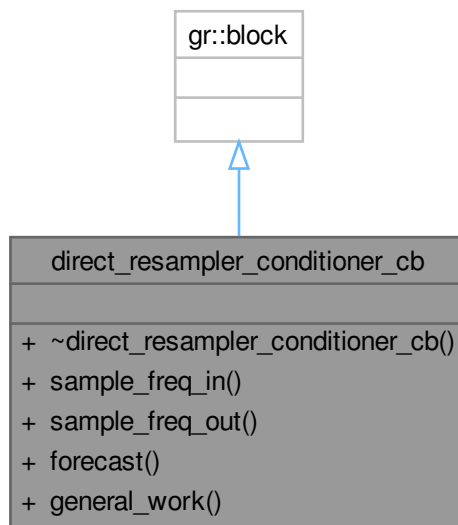
This class implements a direct resampler conditioner for `std::complex<signed char>`

```
#include <direct_resampler_conditioner_cb.h>
```

Inheritance diagram for `direct_resampler_conditioner_cb`:



Collaboration diagram for `direct_resampler_conditioner_cb`:



### Public Member Functions

- unsigned int [sample\\_freq\\_in](#) () const
- unsigned int [sample\\_freq\\_out](#) () const
- void **forecast** (int noutput\_items, gr\_vector\_int &ninput\_items\_required)
- int **general\_work** (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)

### Friends

- `direct_resampler_conditioner_cb_sptr` **direct\_resampler\_make\_conditioner\_cb** (double sample\_freq\_in, double sample\_freq\_out)

## 10.69.1 Detailed Description

This class implements a direct resampler conditioner for `std::complex<signed char>`

Direct resampling without interpolation

Definition at line 46 of file [direct\\_resampler\\_conditioner\\_cb.h](#).

## 10.69.2 Member Function Documentation

### 10.69.2.1 sample\_freq\_in()

```
unsigned int direct_resampler_conditioner_cb::sample_freq_in () const [inline]
```

Definition at line 51 of file [direct\\_resampler\\_conditioner\\_cb.h](#).

### 10.69.2.2 sample\_freq\_out()

```
unsigned int direct_resampler_conditioner_cb::sample_freq_out () const [inline]
```

Definition at line 56 of file [direct\\_resampler\\_conditioner\\_cb.h](#).

The documentation for this class was generated from the following file:

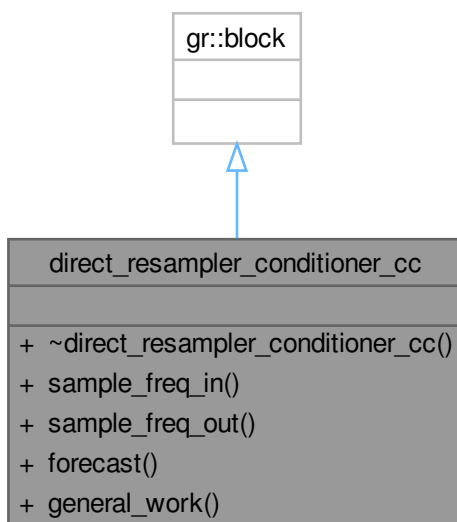
- [direct\\_resampler\\_conditioner\\_cb.h](#)

## 10.70 direct\_resampler\_conditioner\_cc Class Reference

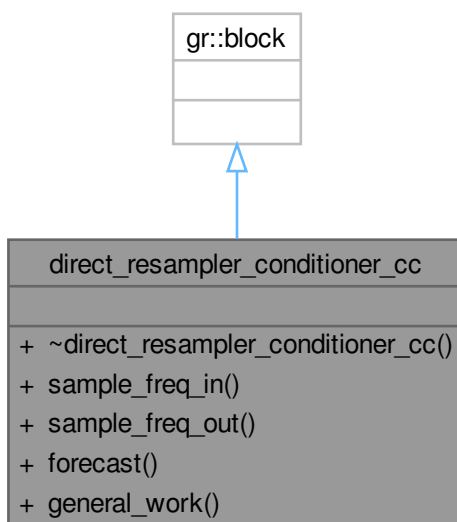
This class implements a direct resampler conditioner for complex data.

```
#include <direct_resampler_conditioner_cc.h>
```

Inheritance diagram for direct\_resampler\_conditioner\_cc:



Collaboration diagram for direct\_resampler\_conditioner\_cc:



## Public Member Functions

- unsigned int [sample\\_freq\\_in](#) () const
- unsigned int [sample\\_freq\\_out](#) () const
- void **forecast** (int noutput\_items, gr\_vector\_int &ninput\_items\_required)
- int **general\_work** (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)

## Friends

- direct\_resampler\_conditioner\_cc\_sptr **direct\_resampler\_make\_conditioner\_cc** (double sample\_freq\_in, double sample\_freq\_out)

### 10.70.1 Detailed Description

This class implements a direct resampler conditioner for complex data.

Direct resampling without interpolation

Definition at line 51 of file [direct\\_resampler\\_conditioner\\_cc.h](#).

### 10.70.2 Member Function Documentation

#### 10.70.2.1 [sample\\_freq\\_in\(\)](#)

```
unsigned int direct_resampler_conditioner_cc::sample_freq_in () const [inline]
```

Definition at line 55 of file [direct\\_resampler\\_conditioner\\_cc.h](#).

#### 10.70.2.2 [sample\\_freq\\_out\(\)](#)

```
unsigned int direct_resampler_conditioner_cc::sample_freq_out () const [inline]
```

Definition at line 60 of file [direct\\_resampler\\_conditioner\\_cc.h](#).

The documentation for this class was generated from the following file:

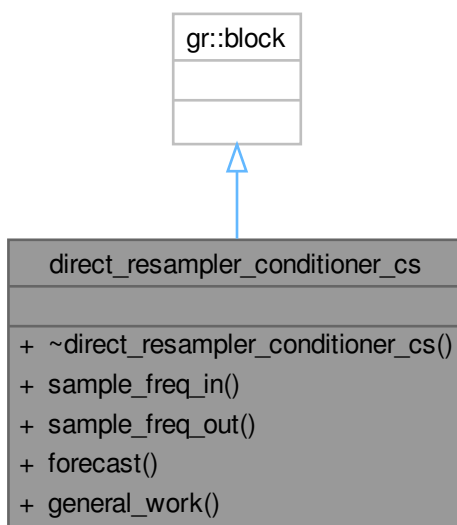
- [direct\\_resampler\\_conditioner\\_cc.h](#)

## 10.71 [direct\\_resampler\\_conditioner\\_cs](#) Class Reference

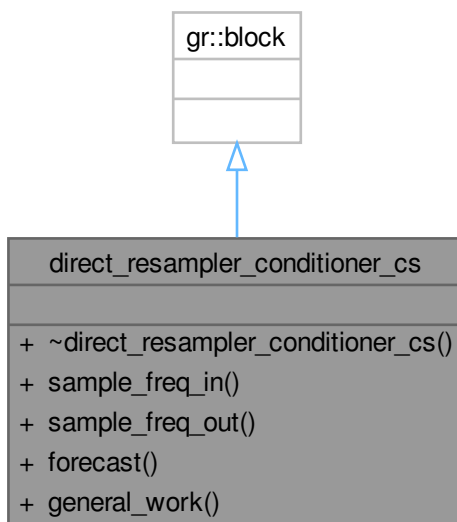
This class implements a direct resampler conditioner for `std::complex<short>`

```
#include <direct_resampler_conditioner_cs.h>
```

Inheritance diagram for direct\_resampler\_conditioner\_cs:



Collaboration diagram for direct\_resampler\_conditioner\_cs:



### Public Member Functions

- unsigned int `sample_freq_in` () const
- unsigned int `sample_freq_out` () const
- void **forecast** (int noutput\_items, gr\_vector\_int &ninput\_items\_required)

- int **general\_work** (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)

#### Friends

- direct\_resampler\_conditioner\_cs\_sptr **direct\_resampler\_make\_conditioner\_cs** (double sample\_freq\_in, double sample\_freq\_out)

### 10.71.1 Detailed Description

This class implements a direct resampler conditioner for `std::complex<short>`

Direct resampling without interpolation

Definition at line 45 of file [direct\\_resampler\\_conditioner\\_cs.h](#).

### 10.71.2 Member Function Documentation

#### 10.71.2.1 sample\_freq\_in()

```
unsigned int direct_resampler_conditioner_cs::sample_freq_in () const [inline]
```

Definition at line 50 of file [direct\\_resampler\\_conditioner\\_cs.h](#).

#### 10.71.2.2 sample\_freq\_out()

```
unsigned int direct_resampler_conditioner_cs::sample_freq_out () const [inline]
```

Definition at line 55 of file [direct\\_resampler\\_conditioner\\_cs.h](#).

The documentation for this class was generated from the following file:

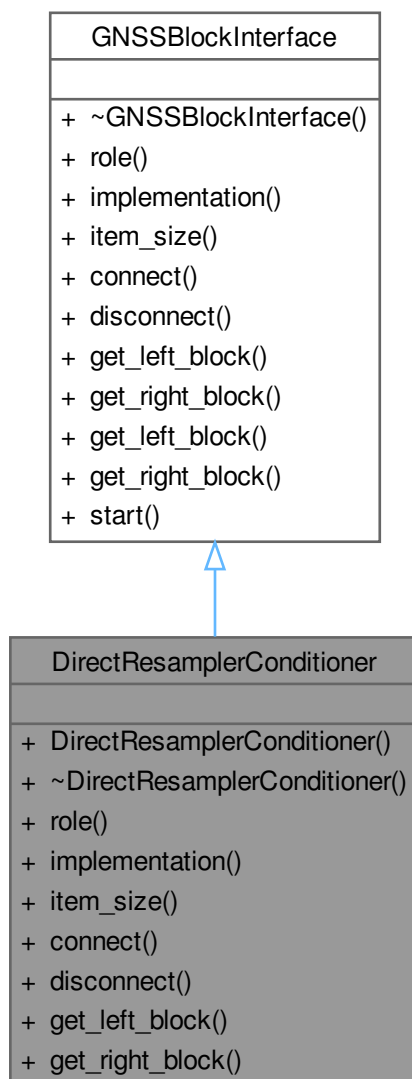
- [direct\\_resampler\\_conditioner\\_cs.h](#)

## 10.72 DirectResamplerConditioner Class Reference

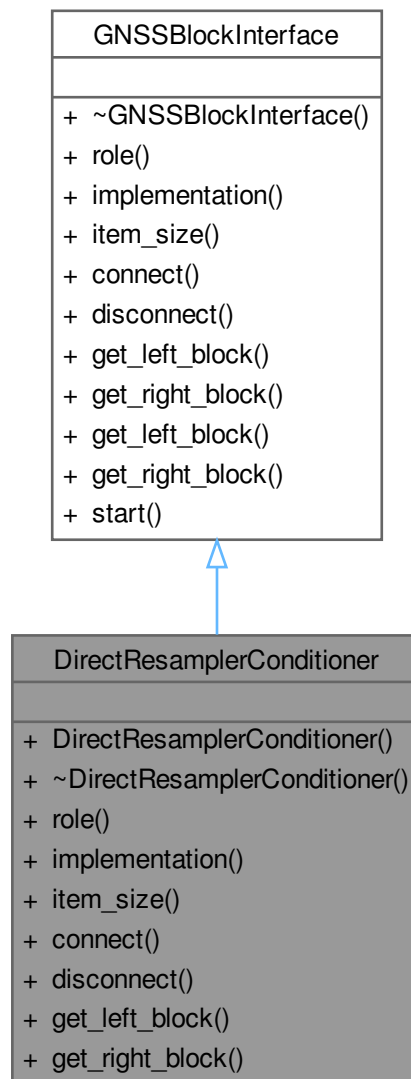
Interface of an adapter of a direct resampler conditioner block to a SignalConditionerInterface.

```
#include <direct_resampler_conditioner.h>
```

Inheritance diagram for DirectResamplerConditioner:



Collaboration diagram for DirectResamplerConditioner:



### Public Member Functions

- **DirectResamplerConditioner** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_stream, unsigned int out\_stream)
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "Direct\_Resampler".
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override



## Public Member Functions inherited from GNSSBlockInterface

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.72.1 Detailed Description

Interface of an adapter of a direct resampler conditioner block to a SignalConditionerInterface.

Definition at line 38 of file [direct\\_resampler\\_conditioner.h](#).

### 10.72.2 Member Function Documentation

#### 10.72.2.1 connect()

```
void DirectResamplerConditioner::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.72.2.2 disconnect()

```
void DirectResamplerConditioner::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.72.2.3 get\_left\_block()

```
gr::basic_block_sptr DirectResamplerConditioner::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.72.2.4 get\_right\_block()

```
gr::basic_block_sptr DirectResamplerConditioner::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.72.2.5 implementation()

```
std::string DirectResamplerConditioner::implementation () [inline], [override], [virtual]
```

Returns "Direct\_Resampler".

Implements [GNSSBlockInterface](#).

Definition at line 53 of file [direct\\_resampler\\_conditioner.h](#).

#### 10.72.2.6 item\_size()

```
size_t DirectResamplerConditioner::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 58 of file [direct\\_resampler\\_conditioner.h](#).

#### 10.72.2.7 role()

```
std::string DirectResamplerConditioner::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

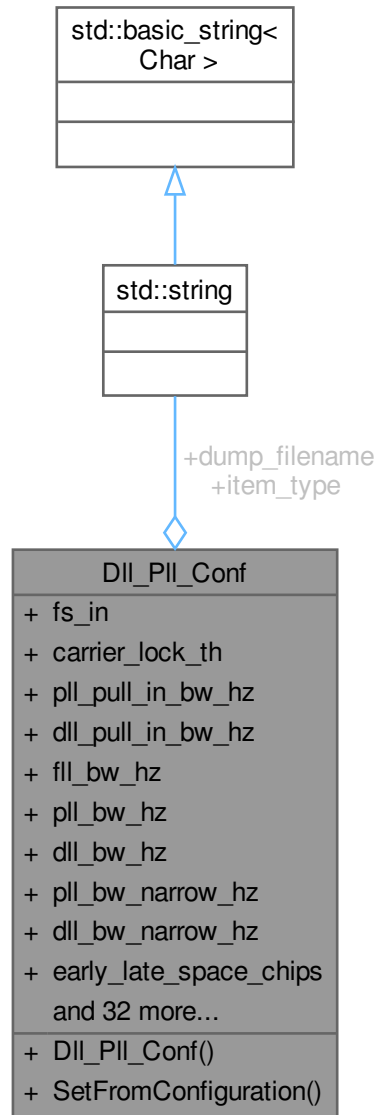
Definition at line 47 of file [direct\\_resampler\\_conditioner.h](#).

The documentation for this class was generated from the following file:

- [direct\\_resampler\\_conditioner.h](#)

## 10.73 Dll\_Pll\_Conf Class Reference

Collaboration diagram for Dll\_Pll\_Conf:



### Public Member Functions

- void **SetFromConfiguration** (const [ConfigurationInterface](#) \*configuration, const std::string &role)

### Public Attributes

- std::string [item\\_type](#) {"gr\_complex"}
- std::string [dump\\_filename](#) {"./dll\_pll\_dump.dat"}
- double [fs\\_in](#) {2000000.0}
- double [carrier\\_lock\\_th](#) {0.0}
- float [pll\\_pull\\_in\\_bw\\_hz](#) {50.0}

- float [dll\\_pull\\_in\\_bw\\_hz](#) {3.0}
- float [fll\\_bw\\_hz](#) {35.0}
- float [pll\\_bw\\_hz](#) {35.0}
- float [dll\\_bw\\_hz](#) {2.0}
- float [pll\\_bw\\_narrow\\_hz](#) {5.0}
- float [dll\\_bw\\_narrow\\_hz](#) {0.75}
- float [early\\_late\\_space\\_chips](#) {0.25}
- float [very\\_early\\_late\\_space\\_chips](#) {0.5}
- float [early\\_late\\_space\\_narrow\\_chips](#) {0.15}
- float [very\\_early\\_late\\_space\\_narrow\\_chips](#) {0.5}
- float [slope](#) {1.0}
- float [spc](#) {0.5}
- float [y\\_intercept](#) {1.0}
- float [cn0\\_smoother\\_alpha](#) {0.002}
- float [carrier\\_lock\\_test\\_smoother\\_alpha](#) {0.002}
- uint32\_t [pull\\_in\\_time\\_s](#) {10U}
- uint32\_t [bit\\_synchronization\\_time\\_limit\\_s](#) {20U}
- uint32\_t [vector\\_length](#) {0U}
- uint32\_t [smoother\\_length](#) {10U}
- int32\_t [fll\\_filter\\_order](#) {1}
- int32\_t [pll\\_filter\\_order](#) {3}
- int32\_t [dll\\_filter\\_order](#) {2}
- int32\_t [extend\\_correlation\\_symbols](#) {1}
- int32\_t [cn0\\_samples](#) {0}
- int32\_t [cn0\\_smoother\\_samples](#) {200}
- int32\_t [carrier\\_lock\\_test\\_smoother\\_samples](#) {25}
- int32\_t [cn0\\_min](#) {0}
- int32\_t [max\\_code\\_lock\\_fail](#) {0}
- int32\_t [max\\_carrier\\_lock\\_fail](#) {0}
- char [signal](#) [3] {}
- char [system](#) {'G'}
- bool [enable\\_fll\\_pull\\_in](#) {false}
- bool [enable\\_fll\\_steady\\_state](#) {false}
- bool [track\\_pilot](#) {true}
- bool [enable\\_doppler\\_correction](#) {false}
- bool [carrier\\_aiding](#) {true}
- bool [high\\_dyn](#) {false}
- bool [dump](#) {false}
- bool [dump\\_mat](#) {true}

### 10.73.1 Detailed Description

Definition at line 32 of file [dll\\_pll\\_conf.h](#).

### 10.73.2 Member Data Documentation

#### 10.73.2.1 bit\_synchronization\_time\_limit\_s

```
uint32_t Dll_Pll_Conf::bit_synchronization_time_limit_s {20U}
```

Definition at line 60 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.2 carrier\_aiding

```
bool Dll_Pll_Conf::carrier_aiding {true}
```

Definition at line 79 of file [dll\\_pll\\_conf.h](#).

### 10.73.2.3 carrier\_lock\_test\_smoother\_alpha

```
float Dll_Pll_Conf::carrier_lock_test_smoother_alpha {0.002}
```

Definition at line 58 of file [dll\\_pll\\_conf.h](#).

### 10.73.2.4 carrier\_lock\_test\_smoother\_samples

```
int32_t Dll_Pll_Conf::carrier_lock_test_smoother_samples {25}
```

Definition at line 69 of file [dll\\_pll\\_conf.h](#).

### 10.73.2.5 carrier\_lock\_th

```
double Dll_Pll_Conf::carrier_lock_th {0.0}
```

Definition at line 42 of file [dll\\_pll\\_conf.h](#).

### 10.73.2.6 cn0\_min

```
int32_t Dll_Pll_Conf::cn0_min {0}
```

Definition at line 70 of file [dll\\_pll\\_conf.h](#).

### 10.73.2.7 cn0\_samples

```
int32_t Dll_Pll_Conf::cn0_samples {0}
```

Definition at line 67 of file [dll\\_pll\\_conf.h](#).

### 10.73.2.8 cn0\_smoother\_alpha

```
float Dll_Pll_Conf::cn0_smoother_alpha {0.002}
```

Definition at line 57 of file [dll\\_pll\\_conf.h](#).

### 10.73.2.9 cn0\_smoother\_samples

```
int32_t Dll_Pll_Conf::cn0_smoother_samples {200}
```

Definition at line 68 of file [dll\\_pll\\_conf.h](#).

### 10.73.2.10 dll\_bw\_hz

```
float Dll_Pll_Conf::dll_bw_hz {2.0}
```

Definition at line 47 of file [dll\\_pll\\_conf.h](#).

### 10.73.2.11 dll\_bw\_narrow\_hz

```
float Dll_Pll_Conf::dll_bw_narrow_hz {0.75}
```

Definition at line 49 of file [dll\\_pll\\_conf.h](#).

### 10.73.2.12 dll\_filter\_order

```
int32_t Dll_Pll_Conf::dll_filter_order {2}
```

Definition at line 65 of file [dll\\_pll\\_conf.h](#).

### 10.73.2.13 dll\_pull\_in\_bw\_hz

```
float Dll_Pll_Conf::dll_pull_in_bw_hz {3.0}
```

Definition at line 44 of file [dll\\_pll\\_conf.h](#).

### 10.73.2.14 dump

```
bool Dll_Pll_Conf::dump {false}
```

Definition at line 81 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.15 dump\_filename

```
std::string Dll_Pll_Conf::dump_filename {"/dll_pll_dump.dat"}
```

Definition at line 40 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.16 dump\_mat

```
bool Dll_Pll_Conf::dump_mat {true}
```

Definition at line 82 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.17 early\_late\_space\_chips

```
float Dll_Pll_Conf::early_late_space_chips {0.25}
```

Definition at line 50 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.18 early\_late\_space\_narrow\_chips

```
float Dll_Pll_Conf::early_late_space_narrow_chips {0.15}
```

Definition at line 52 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.19 enable\_doppler\_correction

```
bool Dll_Pll_Conf::enable_doppler_correction {false}
```

Definition at line 78 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.20 enable\_fll\_pull\_in

```
bool Dll_Pll_Conf::enable_fll_pull_in {false}
```

Definition at line 75 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.21 enable\_fll\_steady\_state

```
bool Dll_Pll_Conf::enable_fll_steady_state {false}
```

Definition at line 76 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.22 extend\_correlation\_symbols

```
int32_t Dll_Pll_Conf::extend_correlation_symbols {1}
```

Definition at line 66 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.23 fll\_bw\_hz

```
float Dll_Pll_Conf::fll_bw_hz {35.0}
```

Definition at line 45 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.24 fll\_filter\_order

```
int32_t Dll_Pll_Conf::fll_filter_order {1}
```

Definition at line 63 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.25 fs\_in

```
double Dll_Pll_Conf::fs_in {2000000.0}
```

Definition at line 41 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.26 high\_dyn

```
bool Dll_Pll_Conf::high_dyn {false}
```

Definition at line 80 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.27 item\_type

```
std::string Dll_Pll_Conf::item_type {"gr_complex"}
```

Definition at line 39 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.28 max\_carrier\_lock\_fail

```
int32_t Dll_Pll_Conf::max_carrier_lock_fail {0}
```

Definition at line 72 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.29 max\_code\_lock\_fail

```
int32_t Dll_Pll_Conf::max_code_lock_fail {0}
```

Definition at line 71 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.30 pll\_bw\_hz

```
float Dll_Pll_Conf::pll_bw_hz {35.0}
```

Definition at line 46 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.31 pll\_bw\_narrow\_hz

```
float Dll_Pll_Conf::pll_bw_narrow_hz {5.0}
```

Definition at line 48 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.32 pll\_filter\_order

```
int32_t Dll_Pll_Conf::pll_filter_order {3}
```

Definition at line 64 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.33 pll\_pull\_in\_bw\_hz

```
float Dll_Pll_Conf::pll_pull_in_bw_hz {50.0}
```

Definition at line 43 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.34 pull\_in\_time\_s

```
uint32_t Dll_Pll_Conf::pull_in_time_s {10U}
```

Definition at line 59 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.35 signal

```
char Dll_Pll_Conf::signal[3] {}
```

Definition at line 73 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.36 slope

```
float Dll_Pll_Conf::slope {1.0}
```

Definition at line 54 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.37 smoother\_length

```
uint32_t Dll_Pll_Conf::smoother_length {10U}
```

Definition at line 62 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.38 spc

```
float Dll_Pll_Conf::spc {0.5}
```

Definition at line 55 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.39 system

```
char Dll_Pll_Conf::system {'G'}
```

Definition at line 74 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.40 track\_pilot

```
bool Dll_Pll_Conf::track_pilot {true}
```

Definition at line 77 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.41 vector\_length

```
uint32_t Dll_Pll_Conf::vector_length {0U}
```

Definition at line 61 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.42 very\_early\_late\_space\_chips

```
float Dll_Pll_Conf::very_early_late_space_chips {0.5}
```

Definition at line 51 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.43 very\_early\_late\_space\_narrow\_chips

```
float Dll_Pll_Conf::very_early_late_space_narrow_chips {0.5}
```

Definition at line 53 of file [dll\\_pll\\_conf.h](#).

#### 10.73.2.44 y\_intercept

```
float Dll_Pll_Conf::y_intercept {1.0}
```

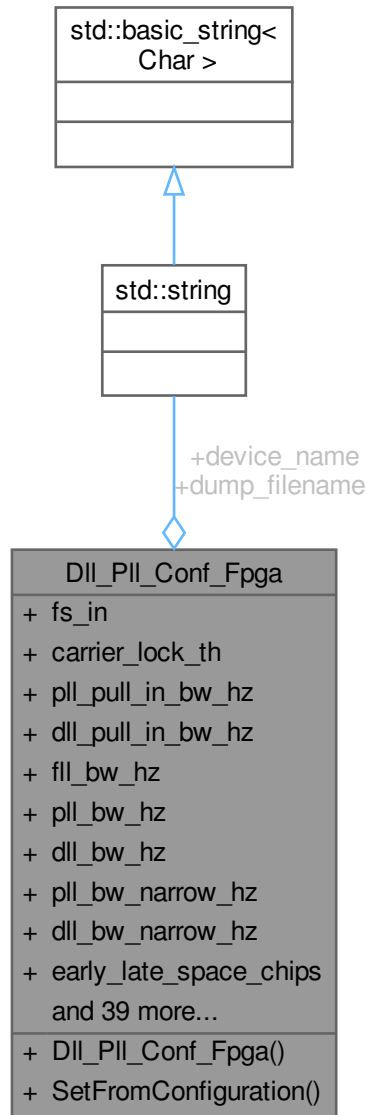
Definition at line 56 of file [dll\\_pll\\_conf.h](#).

The documentation for this class was generated from the following file:

- [dll\\_pll\\_conf.h](#)

## 10.74 Dll\_Pll\_Conf\_Fpga Class Reference

Collaboration diagram for Dll\_Pll\_Conf\_Fpga:



### Public Member Functions

- void **SetFromConfiguration** (const [ConfigurationInterface](#) \*configuration, const std::string &role)

### Public Attributes

- std::string [device\\_name](#) {"dev/uio"}
- std::string [dump\\_filename](#) {"./dll\_pll\_dump.dat"}
- double [fs\\_in](#) {12500000.0}
- double [carrier\\_lock\\_th](#) {0.0}
- float [pll\\_pull\\_in\\_bw\\_hz](#) {50.0}



- float [dll\\_pull\\_in\\_bw\\_hz](#) {3.0}
- float [fill\\_bw\\_hz](#) {35.0}
- float [pll\\_bw\\_hz](#) {5.0}
- float [dll\\_bw\\_hz](#) {0.5}
- float [pll\\_bw\\_narrow\\_hz](#) {2.0}
- float [dll\\_bw\\_narrow\\_hz](#) {0.25}
- float [early\\_late\\_space\\_chips](#) {0.25}
- float [very\\_early\\_late\\_space\\_chips](#) {0.5}
- float [early\\_late\\_space\\_narrow\\_chips](#) {0.15}
- float [very\\_early\\_late\\_space\\_narrow\\_chips](#) {0.5}
- float [slope](#) {1.0}
- float [spc](#) {0.5}
- float [y\\_intercept](#) {1.0}
- float [cn0\\_smoother\\_alpha](#) {0.002}
- float [carrier\\_lock\\_test\\_smoother\\_alpha](#) {0.002}
- uint32\_t [pull\\_in\\_time\\_s](#) {10U}
- uint32\_t [bit\\_synchronization\\_time\\_limit\\_s](#) {70U}
- uint32\_t [vector\\_length](#) {0U}
- uint32\_t [smoother\\_length](#) {10U}
- uint32\_t [code\\_length\\_chips](#) {0U}
- uint32\_t [code\\_samples\\_per\\_chip](#) {0U}
- uint32\_t [extend\\_fpga\\_integration\\_periods](#) {1}
- uint32\_t [fpga\\_integration\\_period](#) {0}
- int32\_t [fill\\_filter\\_order](#) {1}
- int32\_t [pll\\_filter\\_order](#) {3}
- int32\_t [dll\\_filter\\_order](#) {2}
- int32\_t [extend\\_correlation\\_symbols](#) {1}
- int32\_t [cn0\\_samples](#) {0}
- int32\_t [cn0\\_min](#) {0}
- int32\_t [max\\_code\\_lock\\_fail](#) {0}
- int32\_t [max\\_carrier\\_lock\\_fail](#) {0}
- int32\_t [cn0\\_smoother\\_samples](#) {200}
- int32\_t [carrier\\_lock\\_test\\_smoother\\_samples](#) {25}
- int32\_t \* [ca\\_codes](#) {nullptr}
- int32\_t \* [data\\_codes](#) {nullptr}
- char [signal](#) [3] {}
- char [system](#) {'G'}
- bool [extended\\_correlation\\_in\\_fpga](#) {false}
- bool [track\\_pilot](#) {true}
- bool [enable\\_doppler\\_correction](#) {false}
- bool [enable\\_fill\\_pull\\_in](#) {false}
- bool [enable\\_fill\\_steady\\_state](#) {false}
- bool [carrier\\_aiding](#) {true}
- bool [high\\_dyn](#) {false}
- bool [dump](#) {false}
- bool [dump\\_mat](#) {true}

### 10.74.1 Detailed Description

Definition at line 34 of file [dll\\_pll\\_conf\\_fpga.h](#).

### 10.74.2 Member Data Documentation

#### 10.74.2.1 bit\_synchronization\_time\_limit\_s

```
uint32_t Dll_Pll_Conf_Fpga::bit_synchronization_time_limit_s {70U}
```

Definition at line 65 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.2 ca\_codes

```
int32_t* Dll_Pll_Conf_Fpga::ca_codes {nullptr}
```

Definition at line 85 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.3 carrier\_aiding

```
bool Dll_Pll_Conf_Fpga::carrier_aiding {true}
```

Definition at line 96 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.4 carrier\_lock\_test\_smoother\_alpha

```
float Dll_Pll_Conf_Fpga::carrier_lock_test_smoother_alpha {0.002}
```

Definition at line 62 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.5 carrier\_lock\_test\_smoother\_samples

```
int32_t Dll_Pll_Conf_Fpga::carrier_lock_test_smoother_samples {25}
```

Definition at line 82 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.6 carrier\_lock\_th

```
double Dll_Pll_Conf_Fpga::carrier_lock_th {0.0}
```

Definition at line 45 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.7 cn0\_min

```
int32_t Dll_Pll_Conf_Fpga::cn0_min {0}
```

Definition at line 78 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.8 cn0\_samples

```
int32_t Dll_Pll_Conf_Fpga::cn0_samples {0}
```

Definition at line 77 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.9 cn0\_smoother\_alpha

```
float Dll_Pll_Conf_Fpga::cn0_smoother_alpha {0.002}
```

Definition at line 61 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.10 cn0\_smoother\_samples

```
int32_t Dll_Pll_Conf_Fpga::cn0_smoother_samples {200}
```

Definition at line 81 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.11 code\_length\_chips

```
uint32_t Dll_Pll_Conf_Fpga::code_length_chips {0U}
```

Definition at line 68 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.12 code\_samples\_per\_chip

```
uint32_t Dll_Pll_Conf_Fpga::code_samples_per_chip {0U}
```

Definition at line 69 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.13 data\_codes

```
int32_t* Dll_Pll_Conf_Fpga::data_codes {nullptr}
```

Definition at line 86 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.14 device\_name

std::string Dll\_Pll\_Conf\_Fpga::device\_name {"/dev/uio"}  
Definition at line 41 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.15 dll\_bw\_hz

float Dll\_Pll\_Conf\_Fpga::dll\_bw\_hz {0.5}  
Definition at line 51 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.16 dll\_bw\_narrow\_hz

float Dll\_Pll\_Conf\_Fpga::dll\_bw\_narrow\_hz {0.25}  
Definition at line 53 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.17 dll\_filter\_order

int32\_t Dll\_Pll\_Conf\_Fpga::dll\_filter\_order {2}  
Definition at line 75 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.18 dll\_pull\_in\_bw\_hz

float Dll\_Pll\_Conf\_Fpga::dll\_pull\_in\_bw\_hz {3.0}  
Definition at line 48 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.19 dump

bool Dll\_Pll\_Conf\_Fpga::dump {false}  
Definition at line 98 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.20 dump\_filename

std::string Dll\_Pll\_Conf\_Fpga::dump\_filename {"/dll\_pll\_dump.dat"}  
Definition at line 42 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.21 dump\_mat

bool Dll\_Pll\_Conf\_Fpga::dump\_mat {true}  
Definition at line 99 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.22 early\_late\_space\_chips

float Dll\_Pll\_Conf\_Fpga::early\_late\_space\_chips {0.25}  
Definition at line 54 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.23 early\_late\_space\_narrow\_chips

float Dll\_Pll\_Conf\_Fpga::early\_late\_space\_narrow\_chips {0.15}  
Definition at line 56 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.24 enable\_doppler\_correction

bool Dll\_Pll\_Conf\_Fpga::enable\_doppler\_correction {false}  
Definition at line 93 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.25 enable\_fll\_pull\_in

bool Dll\_Pll\_Conf\_Fpga::enable\_fll\_pull\_in {false}  
Definition at line 94 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.26 enable\_fll\_steady\_state

```
bool Dll_Pll_Conf_Fpga::enable_fll_steady_state {false}
```

Definition at line 95 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.27 extend\_correlation\_symbols

```
int32_t Dll_Pll_Conf_Fpga::extend_correlation_symbols {1}
```

Definition at line 76 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.28 extend\_fpga\_integration\_periods

```
uint32_t Dll_Pll_Conf_Fpga::extend_fpga_integration_periods {1}
```

Definition at line 70 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.29 extended\_correlation\_in\_fpga

```
bool Dll_Pll_Conf_Fpga::extended_correlation_in_fpga {false}
```

Definition at line 91 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.30 fll\_bw\_hz

```
float Dll_Pll_Conf_Fpga::fll_bw_hz {35.0}
```

Definition at line 49 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.31 fll\_filter\_order

```
int32_t Dll_Pll_Conf_Fpga::fll_filter_order {1}
```

Definition at line 73 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.32 fpga\_integration\_period

```
uint32_t Dll_Pll_Conf_Fpga::fpga_integration_period {0}
```

Definition at line 71 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.33 fs\_in

```
double Dll_Pll_Conf_Fpga::fs_in {12500000.0}
```

Definition at line 44 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.34 high\_dyn

```
bool Dll_Pll_Conf_Fpga::high_dyn {false}
```

Definition at line 97 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.35 max\_carrier\_lock\_fail

```
int32_t Dll_Pll_Conf_Fpga::max_carrier_lock_fail {0}
```

Definition at line 80 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.36 max\_code\_lock\_fail

```
int32_t Dll_Pll_Conf_Fpga::max_code_lock_fail {0}
```

Definition at line 79 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.37 pll\_bw\_hz

```
float Dll_Pll_Conf_Fpga::pll_bw_hz {5.0}
```

Definition at line 50 of file [dll\\_pll\\_conf\\_fpga.h](#).

**10.74.2.38 pll\_bw\_narrow\_hz**

```
float Dll_Pll_Conf_Fpga::pll_bw_narrow_hz {2.0}
```

Definition at line 52 of file [dll\\_pll\\_conf\\_fpga.h](#).

**10.74.2.39 pll\_filter\_order**

```
int32_t Dll_Pll_Conf_Fpga::pll_filter_order {3}
```

Definition at line 74 of file [dll\\_pll\\_conf\\_fpga.h](#).

**10.74.2.40 pll\_pull\_in\_bw\_hz**

```
float Dll_Pll_Conf_Fpga::pll_pull_in_bw_hz {50.0}
```

Definition at line 47 of file [dll\\_pll\\_conf\\_fpga.h](#).

**10.74.2.41 pull\_in\_time\_s**

```
uint32_t Dll_Pll_Conf_Fpga::pull_in_time_s {10U}
```

Definition at line 64 of file [dll\\_pll\\_conf\\_fpga.h](#).

**10.74.2.42 signal**

```
char Dll_Pll_Conf_Fpga::signal[3] {}
```

Definition at line 88 of file [dll\\_pll\\_conf\\_fpga.h](#).

**10.74.2.43 slope**

```
float Dll_Pll_Conf_Fpga::slope {1.0}
```

Definition at line 58 of file [dll\\_pll\\_conf\\_fpga.h](#).

**10.74.2.44 smoother\_length**

```
uint32_t Dll_Pll_Conf_Fpga::smoother_length {10U}
```

Definition at line 67 of file [dll\\_pll\\_conf\\_fpga.h](#).

**10.74.2.45 spc**

```
float Dll_Pll_Conf_Fpga::spc {0.5}
```

Definition at line 59 of file [dll\\_pll\\_conf\\_fpga.h](#).

**10.74.2.46 system**

```
char Dll_Pll_Conf_Fpga::system {'G'}
```

Definition at line 89 of file [dll\\_pll\\_conf\\_fpga.h](#).

**10.74.2.47 track\_pilot**

```
bool Dll_Pll_Conf_Fpga::track_pilot {true}
```

Definition at line 92 of file [dll\\_pll\\_conf\\_fpga.h](#).

**10.74.2.48 vector\_length**

```
uint32_t Dll_Pll_Conf_Fpga::vector_length {0U}
```

Definition at line 66 of file [dll\\_pll\\_conf\\_fpga.h](#).

**10.74.2.49 very\_early\_late\_space\_chips**

```
float Dll_Pll_Conf_Fpga::very_early_late_space_chips {0.5}
```

Definition at line 55 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.50 very\_early\_late\_space\_narrow\_chips

```
float Dll_Pll_Conf_Fpga::very_early_late_space_narrow_chips {0.5}
```

Definition at line 57 of file [dll\\_pll\\_conf\\_fpga.h](#).

#### 10.74.2.51 y\_intercept

```
float Dll_Pll_Conf_Fpga::y_intercept {1.0}
```

Definition at line 60 of file [dll\\_pll\\_conf\\_fpga.h](#).

The documentation for this class was generated from the following file:

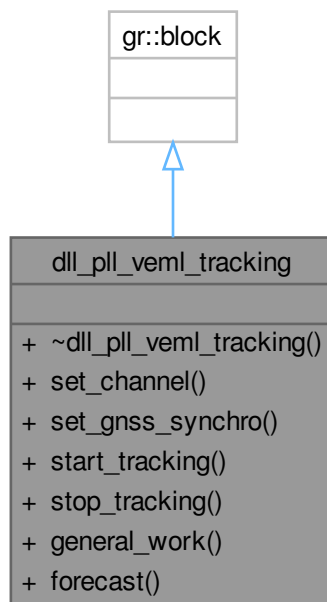
- [dll\\_pll\\_conf\\_fpga.h](#)

### 10.75 dll\_pll\_veml\_tracking Class Reference

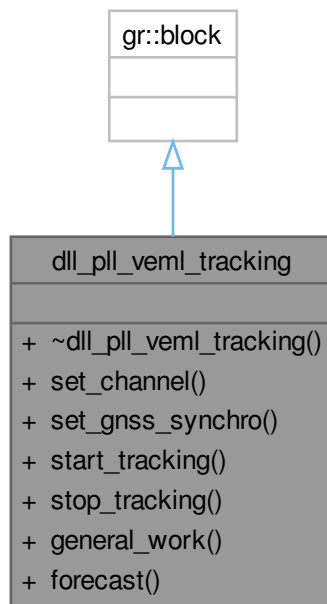
This class implements a code DLL + carrier PLL tracking block.

```
#include <dll_pll_veml_tracking.h>
```

Inheritance diagram for `dll_pll_veml_tracking`:



Collaboration diagram for dll\_pll\_veml\_tracking:



### Public Member Functions

- void **set\_channel** (uint32\_t channel)
- void **set\_gnss\_synchro** ([Gnss\\_Synchro](#) \*p\_gnss\_synchro)
- void **start\_tracking** ()
- void **stop\_tracking** ()
- int **general\_work** (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items) override
- void **forecast** (int noutput\_items, gr\_vector\_int &ninput\_items\_required) override

### Friends

- `dll_pll_veml_tracking_sptr` **dll\_pll\_veml\_make\_tracking** (const [Dll\\_Pll\\_Conf](#) &conf\_)

### 10.75.1 Detailed Description

This class implements a code DLL + carrier PLL tracking block.

Definition at line 58 of file [dll\\_pll\\_veml\\_tracking.h](#).

The documentation for this class was generated from the following file:

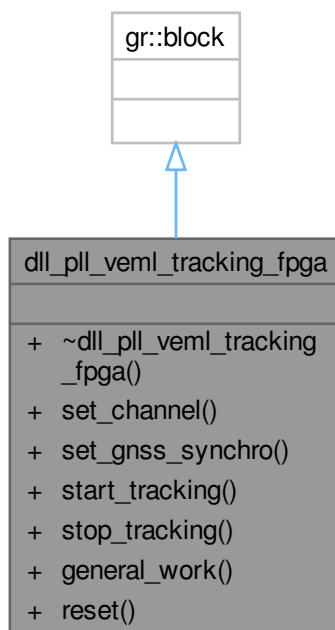
- [dll\\_pll\\_veml\\_tracking.h](#)

## 10.76 dll\_pll\_veml\_tracking\_fpga Class Reference

This class implements a code DLL + carrier PLL tracking block.

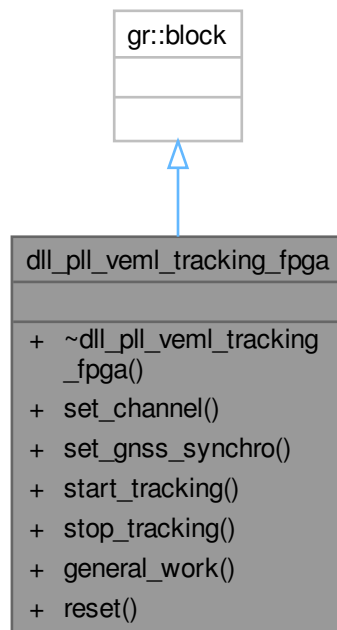
#include <dll\_pll\_veml\_tracking\_fpga.h>

Inheritance diagram for `dll_pll_veml_tracking_fpga`:





Collaboration diagram for `dll_pll_veml_tracking_fpga`:



## Public Member Functions

- `~dll_pll_veml_tracking_fpga ()`  
*Destructor.*
- `void set_channel (uint32_t channel, const std::string &device_io_name)`  
*Set the channel number and configure some multicorrelator parameters.*
- `void set_gnss_synchro (Gnss_Synchro *p_gnss_synchro)`  
*This function is used with two purposes: 1 -> To set the gnss\_synchro 2 -> A set\_gnss\_synchro command with a valid PRN is received when the system is going to run acquisition with that PRN. We can use this command to pre-initialize tracking parameters and variables before the actual acquisition process takes place. In this way we minimize the latency between acquisition and tracking once the acquisition has been made.*
- `void start_tracking ()`  
*This function starts the tracking process.*
- `void stop_tracking ()`  
*This function sets a flag that makes general\_work to stop in order to finish the tracking process.*
- `int general_work (int noutput_items, gr_vector_int &ninput_items, gr_vector_const_void_star &input_items, gr_vector_void_star &output_items)`  
*General Work.*
- `void reset ()`  
*This function disables the HW multicorrelator in the FPGA in order to stop the tracking process.*

## Friends

- `dll_pll_veml_tracking_fpga_sptr dll_pll_veml_make_tracking_fpga (const Dll_Pll_Conf_Fpga &conf_)`

### 10.76.1 Detailed Description

This class implements a code DLL + carrier PLL tracking block.  
Definition at line 58 of file [dll\\_pll\\_veml\\_tracking\\_fpga.h](#).

### 10.76.2 Constructor & Destructor Documentation

#### 10.76.2.1 ~dll\_pll\_veml\_tracking\_fpga()

```
dll_pll_veml_tracking_fpga::~dll_pll_veml_tracking_fpga ()
```

Destructor.

### 10.76.3 Member Function Documentation

#### 10.76.3.1 general\_work()

```
int dll_pll_veml_tracking_fpga::general_work (
    int noutput_items,
    gr_vector_int & ninput_items,
    gr_vector_const_void_star & input_items,
    gr_vector_void_star & output_items)
```

General Work.

#### 10.76.3.2 reset()

```
void dll_pll_veml_tracking_fpga::reset ()
```

This function disables the HW multicorrelator in the FPGA in order to stop the tracking process.

#### 10.76.3.3 set\_channel()

```
void dll_pll_veml_tracking_fpga::set_channel (
    uint32_t channel,
    const std::string & device_io_name)
```

Set the channel number and configure some multicorrelator parameters.

#### 10.76.3.4 set\_gnss\_synchro()

```
void dll_pll_veml_tracking_fpga::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro)
```

This function is used with two purposes: 1 -> To set the gnss\_synchro 2 -> A set\_gnss\_synchro command with a valid PRN is received when the system is going to run acquisition with that PRN. We can use this command to pre-initialize tracking parameters and variables before the actual acquisition process takes place. In this way we minimize the latency between acquisition and tracking once the acquisition has been made.

#### 10.76.3.5 start\_tracking()

```
void dll_pll_veml_tracking_fpga::start_tracking ()
```

This function starts the tracking process.

#### 10.76.3.6 stop\_tracking()

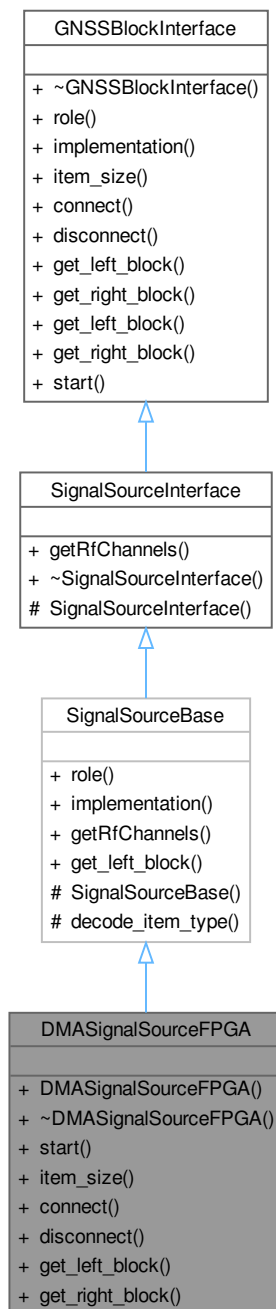
```
void dll_pll_veml_tracking_fpga::stop_tracking ()
```

This function sets a flag that makes general\_work to stop in order to finish the tracking process.  
The documentation for this class was generated from the following file:

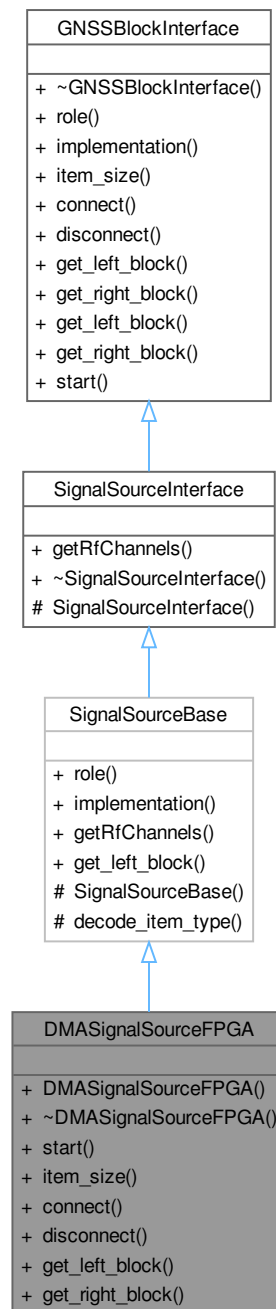
- [dll\\_pll\\_veml\\_tracking\\_fpga.h](#)

## 10.77 DMASignalSourceFPGA Class Reference

Inheritance diagram for DMASignalSourceFPGA:



Collaboration diagram for DMASignalSourceFPGA:



## Public Member Functions

- **DMASignalSourceFPGA** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_stream, unsigned int out\_stream, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)
- void [start](#) () override  
*Start the flow of samples if needed.*
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override

- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override

### Public Member Functions inherited from [SignalSourceBase](#)

- std::string [role](#) () final
- std::string [implementation](#) () final
- size\_t [getRfChannels](#) () const override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)

### Additional Inherited Members

### Protected Member Functions inherited from [SignalSourceBase](#)

- [SignalSourceBase](#) ([ConfigurationInterface](#) const \*configuration, std::string role, std::string impl)  
*Constructor.*
- size\_t [decode\\_item\\_type](#) (std::string const &item\_type, bool \*is\_interleaved=nullptr, bool throw\_on\_↵  
error=false)  
*utility for decoding passed ".item\_type" values*

## 10.77.1 Detailed Description

Definition at line 44 of file [dma\\_signal\\_source\\_fpga.h](#).

## 10.77.2 Member Function Documentation

### 10.77.2.1 connect()

```
void DMASignalSourceFPGA::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.77.2.2 disconnect()

```
void DMASignalSourceFPGA::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.77.2.3 get\_left\_block()

```
gr::basic_block_sptr DMASignalSourceFPGA::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.77.2.4 get\_right\_block()

```
gr::basic_block_sptr DMASignalSourceFPGA::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.77.2.5 item\_size()

```
size_t DMASignalSourceFPGA::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).  
Definition at line 55 of file [dma\\_signal\\_source\\_fpga.h](#).

### 10.77.2.6 start()

```
void DMASignalSourceFPGA::start () [override], [virtual]
```

Start the flow of samples if needed.

Reimplemented from [GNSSBlockInterface](#).

The documentation for this class was generated from the following file:

- [dma\\_signal\\_source\\_fpga.h](#)

## 10.78 DSM\_dsm\_header Class Reference

Collaboration diagram for DSM\_dsm\_header:

DSM_dsm_header
+ dsm_id
+ dsm_block_id
+ DSM_dsm_header()

### Public Attributes

- `uint8_t dsm_id {}`
- `uint8_t dsm_block_id {}`

### 10.78.1 Detailed Description

Definition at line 44 of file [osnma\\_data.h](#).

## 10.78.2 Member Data Documentation

### 10.78.2.1 dsm\_block\_id

```
uint8_t DSM_dsm_header::dsm_block_id {}
```

Definition at line 49 of file [osnma\\_data.h](#).

### 10.78.2.2 dsm\_id

```
uint8_t DSM_dsm_header::dsm_id {}
```

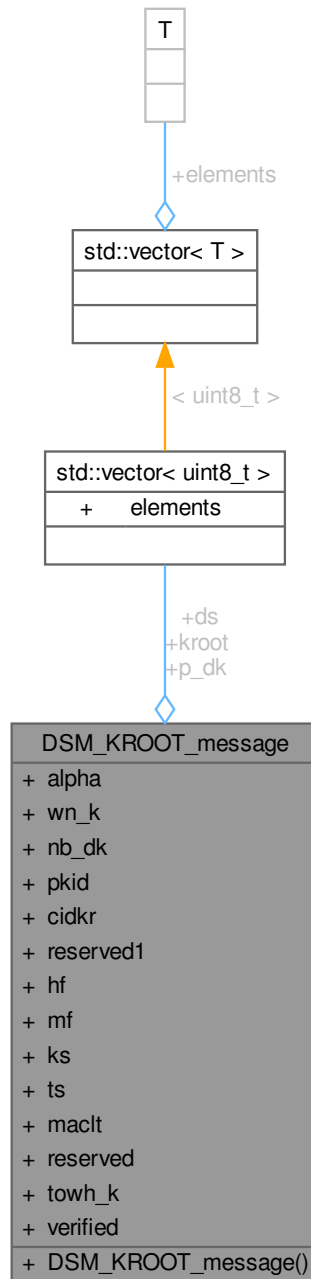
Definition at line 48 of file [osnma\\_data.h](#).

The documentation for this class was generated from the following file:

- [osnma\\_data.h](#)

## 10.79 DSM\_KROOT\_message Class Reference

Collaboration diagram for DSM\_KROOT\_message:



### Public Attributes

- `std::vector< uint8_t > kroot`
- `std::vector< uint8_t > ds`
- `std::vector< uint8_t > p_dk`
- `uint64_t alpha {}`
- `uint16_t wn_k {}`

- uint8\_t `nb_dk` {}
- uint8\_t `pkid` {}
- uint8\_t `cidkr` {}
- uint8\_t `reserved1` {}
- uint8\_t `hf` {}
- uint8\_t `mf` {}
- uint8\_t `ks` {}
- uint8\_t `ts` {}
- uint8\_t `mac1t` {}
- uint8\_t `reserved` {}
- uint8\_t `towh_k` {}
- bool `verified` {false}

### 10.79.1 Detailed Description

Definition at line 98 of file [osnma\\_data.h](#).

### 10.79.2 Member Data Documentation

#### 10.79.2.1 alpha

```
uint64_t DSM_KROOT_message::alpha {}
```

Definition at line 106 of file [osnma\\_data.h](#).

#### 10.79.2.2 cidkr

```
uint8_t DSM_KROOT_message::cidkr {}
```

Definition at line 110 of file [osnma\\_data.h](#).

#### 10.79.2.3 ds

```
std::vector<uint8_t> DSM_KROOT_message::ds
```

Definition at line 104 of file [osnma\\_data.h](#).

#### 10.79.2.4 hf

```
uint8_t DSM_KROOT_message::hf {}
```

Definition at line 112 of file [osnma\\_data.h](#).

#### 10.79.2.5 kroot

```
std::vector<uint8_t> DSM_KROOT_message::kroot
```

Definition at line 103 of file [osnma\\_data.h](#).

#### 10.79.2.6 ks

```
uint8_t DSM_KROOT_message::ks {}
```

Definition at line 114 of file [osnma\\_data.h](#).

#### 10.79.2.7 mac1t

```
uint8_t DSM_KROOT_message::mac1t {}
```

Definition at line 116 of file [osnma\\_data.h](#).

#### 10.79.2.8 mf

```
uint8_t DSM_KROOT_message::mf {}
```

Definition at line 113 of file [osnma\\_data.h](#).



### 10.79.2.9 nb\_dk

```
uint8_t DSM_KROOT_message::nb_dk {}
```

Definition at line 108 of file [osnma\\_data.h](#).

### 10.79.2.10 p\_dk

```
std::vector<uint8_t> DSM_KROOT_message::p_dk
```

Definition at line 105 of file [osnma\\_data.h](#).

### 10.79.2.11 pkid

```
uint8_t DSM_KROOT_message::pkid {}
```

Definition at line 109 of file [osnma\\_data.h](#).

### 10.79.2.12 reserved

```
uint8_t DSM_KROOT_message::reserved {}
```

Definition at line 117 of file [osnma\\_data.h](#).

### 10.79.2.13 reserved1

```
uint8_t DSM_KROOT_message::reserved1 {}
```

Definition at line 111 of file [osnma\\_data.h](#).

### 10.79.2.14 towh\_k

```
uint8_t DSM_KROOT_message::towh_k {}
```

Definition at line 118 of file [osnma\\_data.h](#).

### 10.79.2.15 ts

```
uint8_t DSM_KROOT_message::ts {}
```

Definition at line 115 of file [osnma\\_data.h](#).

### 10.79.2.16 verified

```
bool DSM_KROOT_message::verified {false}
```

Definition at line 119 of file [osnma\\_data.h](#).

### 10.79.2.17 wn\_k

```
uint16_t DSM_KROOT_message::wn_k {}
```

Definition at line 107 of file [osnma\\_data.h](#).

The documentation for this class was generated from the following file:

- [osnma\\_data.h](#)

## 10.80 DSM\_nma\_header Class Reference

Collaboration diagram for DSM\_nma\_header:

DSM_nma_header
+ nmas
+ cid
+ cpks
+ reserved
+ DSM_nma_header()

### Public Attributes

- uint8\_t [nmas](#) {}
- uint8\_t [cid](#) {}
- uint8\_t [cpks](#) {}
- bool [reserved](#) {}

### 10.80.1 Detailed Description

Definition at line 33 of file [osnma\\_data.h](#).

### 10.80.2 Member Data Documentation

#### 10.80.2.1 cid

```
uint8_t DSM_nma_header::cid {}
```

Definition at line 38 of file [osnma\\_data.h](#).

#### 10.80.2.2 cpks

```
uint8_t DSM_nma_header::cpks {}
```

Definition at line 39 of file [osnma\\_data.h](#).

#### 10.80.2.3 nmas

```
uint8_t DSM_nma_header::nmas {}
```

Definition at line 37 of file [osnma\\_data.h](#).

#### 10.80.2.4 reserved

```
bool DSM_nma_header::reserved {}
```

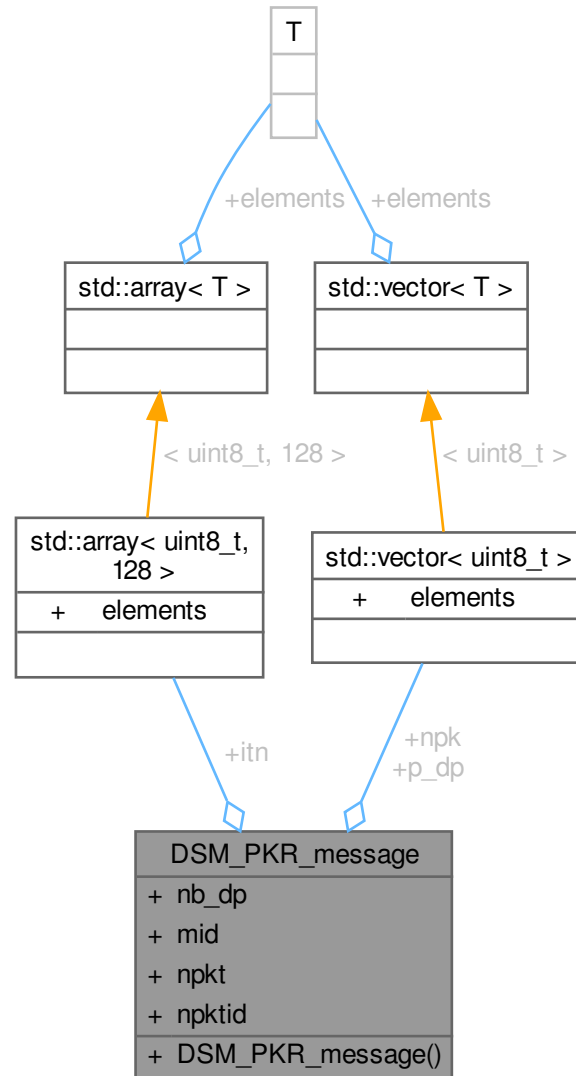
Definition at line 40 of file [osnma\\_data.h](#).

The documentation for this class was generated from the following file:

- [osnma\\_data.h](#)

## 10.81 DSM\_PKR\_message Class Reference

Collaboration diagram for DSM\_PKR\_message:



### Public Attributes

- `std::array< uint8_t, 128 > itn {}`
- `std::vector< uint8_t > npk`
- `std::vector< uint8_t > p_dp`
- `uint8_t nb_dp {}`
- `uint8_t mid {}`
- `uint8_t npkt {}`
- `uint8_t npktid {}`

### 10.81.1 Detailed Description

Definition at line 83 of file [osnma\\_data.h](#).

### 10.81.2 Member Data Documentation

#### 10.81.2.1 itn

```
std::array<uint8_t, 128> DSM_PKR_message::itn {}
```

Definition at line 88 of file [osnma\\_data.h](#).

#### 10.81.2.2 mid

```
uint8_t DSM_PKR_message::mid {}
```

Definition at line 92 of file [osnma\\_data.h](#).

#### 10.81.2.3 nb\_dp

```
uint8_t DSM_PKR_message::nb_dp {}
```

Definition at line 91 of file [osnma\\_data.h](#).

#### 10.81.2.4 npk

```
std::vector<uint8_t> DSM_PKR_message::npk
```

Definition at line 89 of file [osnma\\_data.h](#).

#### 10.81.2.5 npkt

```
uint8_t DSM_PKR_message::npkt {}
```

Definition at line 93 of file [osnma\\_data.h](#).

#### 10.81.2.6 npktid

```
uint8_t DSM_PKR_message::npktid {}
```

Definition at line 94 of file [osnma\\_data.h](#).

#### 10.81.2.7 p\_dp

```
std::vector<uint8_t> DSM_PKR_message::p_dp
```

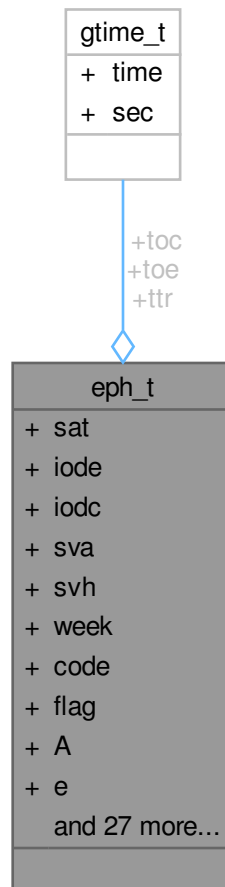
Definition at line 90 of file [osnma\\_data.h](#).

The documentation for this class was generated from the following file:

- [osnma\\_data.h](#)

## 10.82 eph\_t Struct Reference

Collaboration diagram for eph\_t:



### Public Attributes

- int [sat](#)
- int [iode](#)
- int [iodc](#)
- int [sva](#)
- int [svh](#)
- int [week](#)
- int [code](#)
- int [flag](#)
- [gtime\\_t](#) [toe](#)
- [gtime\\_t](#) [toc](#)
- [gtime\\_t](#) [ttr](#)
- double [A](#)
- double [e](#)
- double [i0](#)
- double [OMG0](#)

- double [omg](#)
- double [M0](#)
- double [deln](#)
- double [OMGd](#)
- double [idot](#)
- double [crc](#)
- double [crs](#)
- double [cuc](#)
- double [cus](#)
- double [cic](#)
- double [cis](#)
- double [toes](#)
- double [fit](#)
- double [f0](#)
- double [f1](#)
- double [f2](#)
- double [tgd](#) [4]
- double [isc](#) [4]
- double [Adot](#)
- double [ndot](#)
- float [has\\_clock\\_correction\\_m](#)
- float [has\\_orbit\\_radial\\_correction\\_m](#)
- float [has\\_orbit\\_in\\_track\\_correction\\_m](#)
- float [has\\_orbit\\_cross\\_track\\_correction\\_m](#)
- bool [apply\\_has\\_corrections](#)

### 10.82.1 Detailed Description

Definition at line [432](#) of file [rtklib.h](#).

### 10.82.2 Member Data Documentation

#### 10.82.2.1 A

```
double eph_t::A
```

Definition at line [443](#) of file [rtklib.h](#).

#### 10.82.2.2 Adot

```
double eph_t::Adot
```

Definition at line [453](#) of file [rtklib.h](#).

#### 10.82.2.3 apply\_has\_corrections

```
bool eph_t::apply_has_corrections
```

Definition at line [458](#) of file [rtklib.h](#).

#### 10.82.2.4 cic

```
double eph_t::cic
```

Definition at line [444](#) of file [rtklib.h](#).

#### 10.82.2.5 cis

```
double eph_t::cis
```

Definition at line [444](#) of file [rtklib.h](#).

### 10.82.2.6 code

int eph\_t::code

Definition at line 439 of file [rtklib.h](#).

### 10.82.2.7 crc

double eph\_t::crc

Definition at line 444 of file [rtklib.h](#).

### 10.82.2.8 crs

double eph\_t::crs

Definition at line 444 of file [rtklib.h](#).

### 10.82.2.9 cuc

double eph\_t::cuc

Definition at line 444 of file [rtklib.h](#).

### 10.82.2.10 cus

double eph\_t::cus

Definition at line 444 of file [rtklib.h](#).

### 10.82.2.11 deln

double eph\_t::deln

Definition at line 443 of file [rtklib.h](#).

### 10.82.2.12 e

double eph\_t::e

Definition at line 443 of file [rtklib.h](#).

### 10.82.2.13 f0

double eph\_t::f0

Definition at line 447 of file [rtklib.h](#).

### 10.82.2.14 f1

double eph\_t::f1

Definition at line 447 of file [rtklib.h](#).

### 10.82.2.15 f2

double eph\_t::f2

Definition at line 447 of file [rtklib.h](#).

### 10.82.2.16 fit

double eph\_t::fit

Definition at line 446 of file [rtklib.h](#).

### 10.82.2.17 flag

int eph\_t::flag

Definition at line 440 of file [rtklib.h](#).

**10.82.2.18 has\_clock\_correction\_m**

float eph\_t::has\_clock\_correction\_m  
Definition at line 454 of file [rtklib.h](#).

**10.82.2.19 has\_orbit\_cross\_track\_correction\_m**

float eph\_t::has\_orbit\_cross\_track\_correction\_m  
Definition at line 457 of file [rtklib.h](#).

**10.82.2.20 has\_orbit\_in\_track\_correction\_m**

float eph\_t::has\_orbit\_in\_track\_correction\_m  
Definition at line 456 of file [rtklib.h](#).

**10.82.2.21 has\_orbit\_radial\_correction\_m**

float eph\_t::has\_orbit\_radial\_correction\_m  
Definition at line 455 of file [rtklib.h](#).

**10.82.2.22 i0**

double eph\_t::i0  
Definition at line 443 of file [rtklib.h](#).

**10.82.2.23 idot**

double eph\_t::idot  
Definition at line 443 of file [rtklib.h](#).

**10.82.2.24 iode**

int eph\_t::iode  
Definition at line 435 of file [rtklib.h](#).

**10.82.2.25 iode**

int eph\_t::iode  
Definition at line 435 of file [rtklib.h](#).

**10.82.2.26 isc**

double eph\_t::isc[4]  
Definition at line 452 of file [rtklib.h](#).

**10.82.2.27 M0**

double eph\_t::M0  
Definition at line 443 of file [rtklib.h](#).

**10.82.2.28 ndot**

double eph\_t::ndot  
Definition at line 453 of file [rtklib.h](#).

**10.82.2.29 omg**

double eph\_t::omg  
Definition at line 443 of file [rtklib.h](#).



**10.82.2.30 OMG0**

```
double eph_t::OMG0
```

Definition at line 443 of file [rtklib.h](#).

**10.82.2.31 OMGd**

```
double eph_t::OMGd
```

Definition at line 443 of file [rtklib.h](#).

**10.82.2.32 sat**

```
int eph_t::sat
```

Definition at line 434 of file [rtklib.h](#).

**10.82.2.33 sva**

```
int eph_t::sva
```

Definition at line 436 of file [rtklib.h](#).

**10.82.2.34 svh**

```
int eph_t::svh
```

Definition at line 437 of file [rtklib.h](#).

**10.82.2.35 tgd**

```
double eph_t::tgd[4]
```

Definition at line 448 of file [rtklib.h](#).

**10.82.2.36 toc**

```
gtime_t eph_t::toc
```

Definition at line 441 of file [rtklib.h](#).

**10.82.2.37 toe**

```
gtime_t eph_t::toe
```

Definition at line 441 of file [rtklib.h](#).

**10.82.2.38 toes**

```
double eph_t::toes
```

Definition at line 445 of file [rtklib.h](#).

**10.82.2.39 ttr**

```
gtime_t eph_t::ttr
```

Definition at line 441 of file [rtklib.h](#).

**10.82.2.40 week**

```
int eph_t::week
```

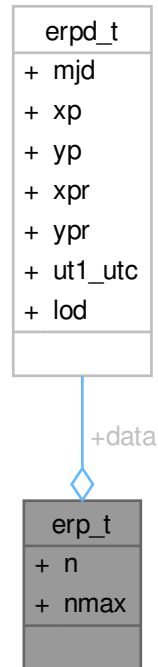
Definition at line 438 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.83 `erp_t` Struct Reference

Collaboration diagram for `erp_t`:



### Public Attributes

- `int n`
- `int nmax`
- `erpd_t * data`

### 10.83.1 Detailed Description

Definition at line 392 of file [rtklib.h](#).

### 10.83.2 Member Data Documentation

#### 10.83.2.1 `data`

`erpd_t* erp_t::data`

Definition at line 395 of file [rtklib.h](#).

#### 10.83.2.2 `n`

`int erp_t::n`

Definition at line 394 of file [rtklib.h](#).

#### 10.83.2.3 `nmax`

`int erp_t::nmax`

Definition at line 394 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.84 erpd\_t Struct Reference

Collaboration diagram for erpd\_t:

erpd_t
+ mjd
+ xp
+ yp
+ xpr
+ ypr
+ ut1_utc
+ lod

### Public Attributes

- double [mjd](#)
- double [xp](#)
- double [yp](#)
- double [xpr](#)
- double [ypr](#)
- double [ut1\\_utc](#)
- double [lod](#)

### 10.84.1 Detailed Description

Definition at line [382](#) of file [rtklib.h](#).

### 10.84.2 Member Data Documentation

#### 10.84.2.1 lod

`double erpd_t::lod`

Definition at line [388](#) of file [rtklib.h](#).

#### 10.84.2.2 mjd

`double erpd_t::mjd`

Definition at line [384](#) of file [rtklib.h](#).

#### 10.84.2.3 ut1\_utc

`double erpd_t::ut1_utc`

Definition at line [387](#) of file [rtklib.h](#).

**10.84.2.4 xp**

```
double erpd_t::xp
```

Definition at line 385 of file [rtklib.h](#).

**10.84.2.5 xpr**

```
double erpd_t::xpr
```

Definition at line 386 of file [rtklib.h](#).

**10.84.2.6 yp**

```
double erpd_t::yp
```

Definition at line 385 of file [rtklib.h](#).

**10.84.2.7 ypr**

```
double erpd_t::ypr
```

Definition at line 386 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

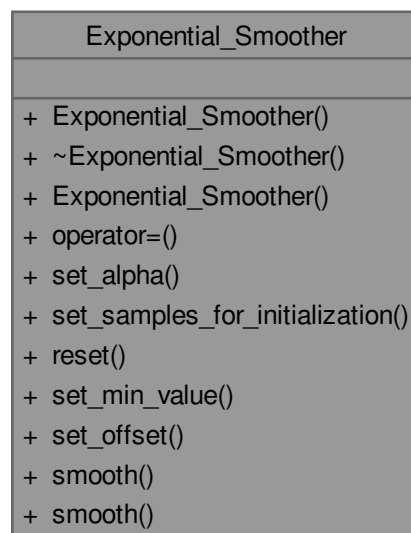
- [rtklib.h](#)

**10.85 Exponential\_Smoother Class Reference**

Class that implements a first-order exponential smoother.

```
#include <exponential_smoother.h>
```

Collaboration diagram for Exponential\_Smoother:

**Public Member Functions**

- [Exponential\\_Smoother](#) ()

*Constructor.*

- [~Exponential\\_Smoother](#) ()=default  
*Destructor.*
- [Exponential\\_Smoother](#) ([Exponential\\_Smoother](#) &&)=default  
*Move operator.*
- [Exponential\\_Smoother](#) & [operator=](#) ([Exponential\\_Smoother](#) &&)=default  
*Move assignment operator.*
- void [set\\_alpha](#) (float alpha)  
 *$0 < \alpha < 1$ . The higher, the most responsive, but more variance. Default value: 0.001*
- void [set\\_samples\\_for\\_initialization](#) (int num\_samples)  
*Number of samples averaged for initialization. Default value: 200.*
- void [reset](#) ()
- void [set\\_min\\_value](#) (float value)
- void [set\\_offset](#) (float offset)
- float [smooth](#) (float raw)
- double [smooth](#) (double raw)

### 10.85.1 Detailed Description

Class that implements a first-order exponential smoother.

`smoothed_value[k] = alpha * raw + (1-alpha) * smoothed_value[k-1]`

The length of the initialization can be controlled with [set\\_samples\\_for\\_initialization\(int num\\_samples\)](#)

Definition at line 39 of file [exponential\\_smoother.h](#).

### 10.85.2 Constructor & Destructor Documentation

#### 10.85.2.1 Exponential\_Smoother() [1/2]

`Exponential_Smoother::Exponential_Smoother ()`

Constructor.

Referenced by [Exponential\\_Smoother\(\)](#), and [operator=\(\)](#).

#### 10.85.2.2 ~Exponential\_Smoother()

`Exponential_Smoother::~Exponential_Smoother () [default]`

Destructor.

#### 10.85.2.3 Exponential\_Smoother() [2/2]

`Exponential_Smoother::Exponential_Smoother (  
    Exponential\_Smoother && ) [default]`

Move operator.

References [Exponential\\_Smoother\(\)](#).

Here is the call graph for this function:



### 10.85.3 Member Function Documentation

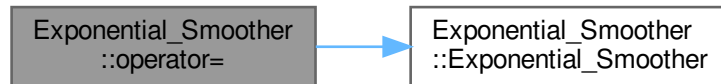
#### 10.85.3.1 `operator=()`

```
Exponential_Smoother & Exponential_Smoother::operator= (
    Exponential_Smoother && ) [default]
```

Move assignment operator.

References [Exponential\\_Smoother\(\)](#).

Here is the call graph for this function:



#### 10.85.3.2 `set_alpha()`

```
void Exponential_Smoother::set_alpha (
    float alpha)
```

$0 < \alpha < 1$ . The higher, the most responsive, but more variance. Default value: 0.001

#### 10.85.3.3 `set_samples_for_initialization()`

```
void Exponential_Smoother::set_samples_for_initialization (
    int num_samples)
```

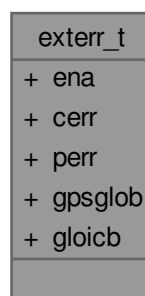
Number of samples averaged for initialization. Default value: 200.

The documentation for this class was generated from the following file:

- [exponential\\_smoother.h](#)

## 10.86 `exterr_t` Struct Reference

Collaboration diagram for `exterr_t`:



## Public Attributes

- int [ena](#) [4]
- double [cerr](#) [4][[NFREQ](#) \*2]
- double [perr](#) [4][[NFREQ](#) \*2]
- double [gpsglob](#) [[NFREQ](#)]
- double [gloicb](#) [[NFREQ](#)]

## 10.86.1 Detailed Description

Definition at line [925](#) of file [rtklib.h](#).

## 10.86.2 Member Data Documentation

### 10.86.2.1 cerr

```
double exterr_t::cerr[4][NFREQ *2]
```

Definition at line [928](#) of file [rtklib.h](#).

### 10.86.2.2 ena

```
int exterr_t::ena[4]
```

Definition at line [927](#) of file [rtklib.h](#).

### 10.86.2.3 gloicb

```
double exterr_t::gloicb[NFREQ]
```

Definition at line [931](#) of file [rtklib.h](#).

### 10.86.2.4 gpsglob

```
double exterr_t::gpsglob[NFREQ]
```

Definition at line [930](#) of file [rtklib.h](#).

### 10.86.2.5 perr

```
double exterr_t::perr[4][NFREQ *2]
```

Definition at line [929](#) of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.87 fcbd\_t Struct Reference

Collaboration diagram for fcbd\_t:



### Public Attributes

- [gtime\\_t ts](#)
- [gtime\\_t te](#)
- double [bias](#) [MAXSAT][3]
- double [std](#) [MAXSAT][3]

### 10.87.1 Detailed Description

Definition at line 557 of file [rtklib.h](#).

### 10.87.2 Member Data Documentation

#### 10.87.2.1 bias

```
double fcbd_t::bias[MAXSAT][3]
```

Definition at line 560 of file [rtklib.h](#).

#### 10.87.2.2 std

```
double fcbd_t::std[MAXSAT][3]
```

Definition at line 561 of file [rtklib.h](#).

#### 10.87.2.3 te

```
gtime_t fcbd_t::te
```

Definition at line 559 of file [rtklib.h](#).



### 10.87.2.4 ts

`ptime_t fcbd_t::ts`

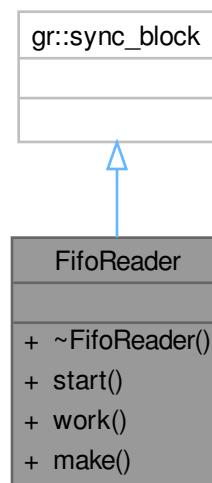
Definition at line 559 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

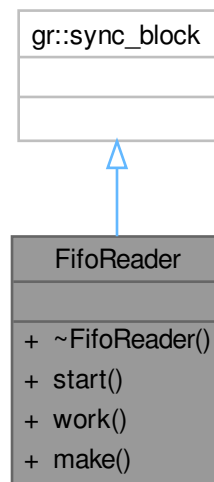
- [rtklib.h](#)

## 10.88 FifoReader Class Reference

Inheritance diagram for FifoReader:



Collaboration diagram for FifoReader:



## Public Types

- using `sptr` = `gnss_shared_ptr<FifoReader>`  
static function to create a class instance

## Public Member Functions

- bool `start` ()  
initialize istream resource for FIFO
- int `work` (int noutput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)

## Static Public Member Functions

- static `sptr make` (const std::string &file\_name, const std::string &sample\_type)

## 10.88.1 Detailed Description

Definition at line 31 of file `fifo_reader.h`.

## 10.88.2 Member Typedef Documentation

### 10.88.2.1 `sptr`

using `FifoReader::sptr` = `gnss_shared_ptr<FifoReader>`  
static function to create a class instance  
Definition at line 35 of file `fifo_reader.h`.

## 10.88.3 Member Function Documentation

### 10.88.3.1 `start()`

```
bool FifoReader::start ()
```

initialize istream resource for FIFO

The documentation for this class was generated from the following file:

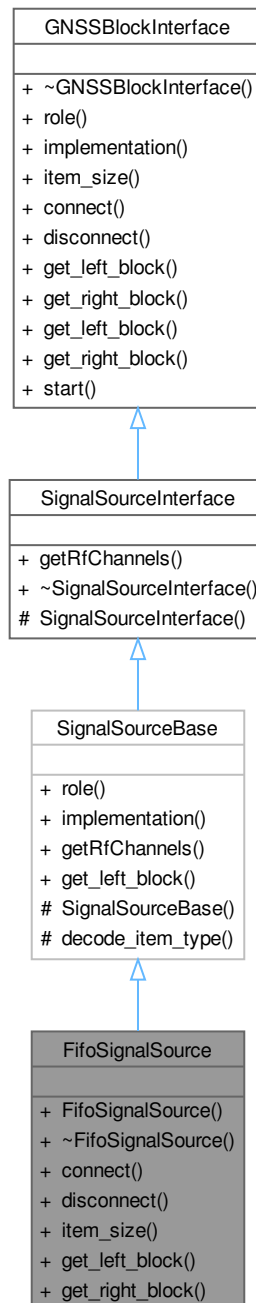
- [fifo\\_reader.h](#)

## 10.89 FifoSignalSource Class Reference

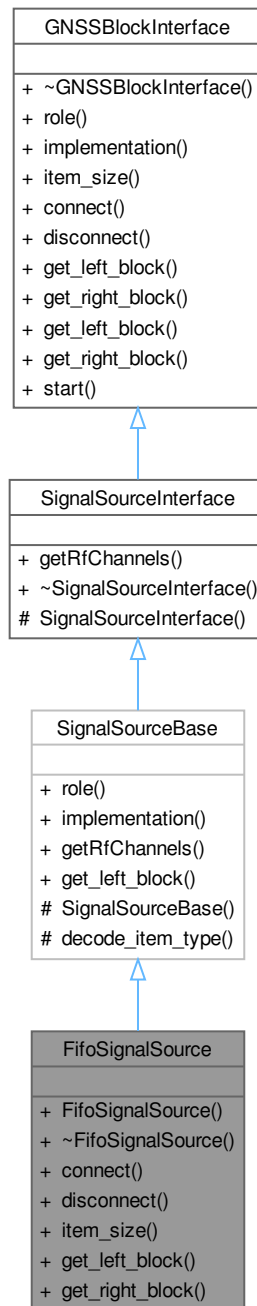
Class that reads a sample stream from a Unix FIFO.

```
#include <fifo_signal_source.h>
```

Inheritance diagram for FifoSignalSource:



Collaboration diagram for FifoSignalSource:



## Public Member Functions

- **FifoSignalSource** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)
- void [connect](#) (gr::top\_block\_sptr top\_block) override  
*override methods from [GNSSBlockInterface](#)*
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- size\_t [item\\_size](#) () override

- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override

### Public Member Functions inherited from [SignalSourceBase](#)

- std::string [role](#) () final
- std::string [implementation](#) () final
- size\_t [getRfChannels](#) () const override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### Additional Inherited Members

### Protected Member Functions inherited from [SignalSourceBase](#)

- [SignalSourceBase](#) ([ConfigurationInterface](#) const \*configuration, std::string role, std::string impl)  
*Constructor.*
- size\_t [decode\\_item\\_type](#) (std::string const &item\_type, bool \*is\_interleaved=nullptr, bool throw\_on\_error=false)  
*utility for decoding passed ".item\_type" values*

## 10.89.1 Detailed Description

Class that reads a sample stream from a Unix FIFO.

This class supports the following properties:

.filename - the path to the input file

- may be overridden by the -signal\_source or -s command-line arguments

.sample\_type - data type read out from the FIFO. default is short ;

- note: not output format. that is always gr\_complex

.dump - whether to archive input data

.dump\_filename - if dumping, path to file for output

Definition at line 50 of file [fifo\\_signal\\_source.h](#).

## 10.89.2 Member Function Documentation

### 10.89.2.1 connect()

```
void FifoSignalSource::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

override methods from [GNSSBlockInterface](#)  
Implements [GNSSBlockInterface](#).

### 10.89.2.2 disconnect()

```
void FifoSignalSource::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.89.2.3 `get_left_block()`

`gr::basic_block_sptr FifoSignalSource::get_left_block () [override], [virtual]`

Implements [GNSSBlockInterface](#).

### 10.89.2.4 `get_right_block()`

`gr::basic_block_sptr FifoSignalSource::get_right_block () [override], [virtual]`

Implements [GNSSBlockInterface](#).

### 10.89.2.5 `item_size()`

`size_t FifoSignalSource::item_size () [override], [virtual]`

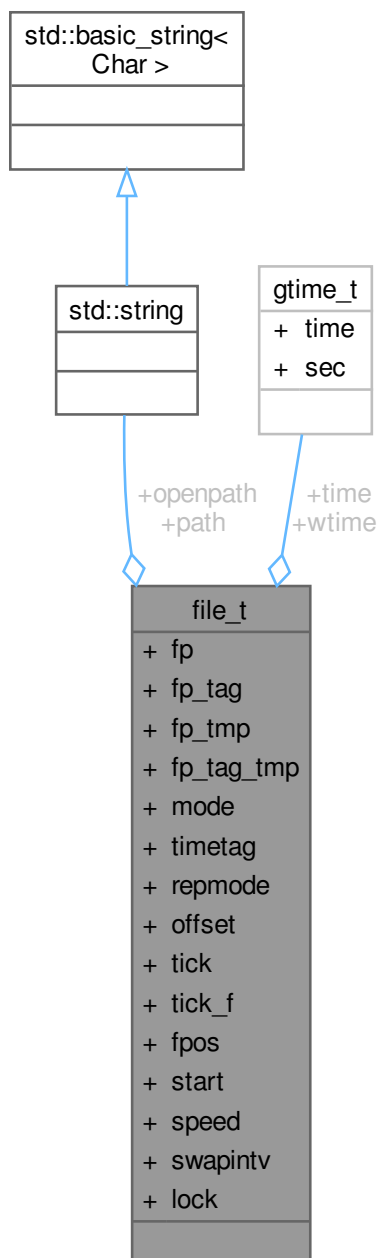
Implements [GNSSBlockInterface](#).

The documentation for this class was generated from the following file:

- [fifo\\_signal\\_source.h](#)

## 10.90 file\_t Struct Reference

Collaboration diagram for file\_t:



### Public Attributes

- FILE \* `fp` = nullptr
- FILE \* `fp_tag` = nullptr
- FILE \* `fp_tmp` = nullptr
- FILE \* `fp_tag_tmp` = nullptr
- std::string `path`

- `std::string openpath`
- `int mode = 0`
- `int timetag`
- `int repmode = 0`
- `int offset = 0`
- `gtime_t time = {}`
- `gtime_t wtime = {}`
- `unsigned int tick = 0`
- `unsigned int tick_f = 0`
- `unsigned int fpos = 0`
- `double start = 0`
- `double speed = 0`
- `double swapintv = 0`
- `lock_t lock`

### 10.90.1 Detailed Description

Definition at line 1117 of file [rtklib.h](#).

### 10.90.2 Member Data Documentation

#### 10.90.2.1 fp

```
FILE* file_t::fp = nullptr
```

Definition at line 1119 of file [rtklib.h](#).

#### 10.90.2.2 fp\_tag

```
FILE* file_t::fp_tag = nullptr
```

Definition at line 1120 of file [rtklib.h](#).

#### 10.90.2.3 fp\_tag\_tmp

```
FILE* file_t::fp_tag_tmp = nullptr
```

Definition at line 1122 of file [rtklib.h](#).

#### 10.90.2.4 fp\_tmp

```
FILE* file_t::fp_tmp = nullptr
```

Definition at line 1121 of file [rtklib.h](#).

#### 10.90.2.5 fpos

```
unsigned int file_t::fpos = 0
```

Definition at line 1133 of file [rtklib.h](#).

#### 10.90.2.6 lock

```
lock_t file_t::lock
```

Definition at line 1137 of file [rtklib.h](#).

#### 10.90.2.7 mode

```
int file_t::mode = 0
```

Definition at line 1125 of file [rtklib.h](#).

#### 10.90.2.8 offset

```
int file_t::offset = 0
```

Definition at line 1128 of file [rtklib.h](#).



### 10.90.2.9 openpath

`std::string file_t::openpath`  
Definition at line 1124 of file [rtklib.h](#).

### 10.90.2.10 path

`std::string file_t::path`  
Definition at line 1123 of file [rtklib.h](#).

### 10.90.2.11 repmode

`int file_t::repmode = 0`  
Definition at line 1127 of file [rtklib.h](#).

### 10.90.2.12 speed

`double file_t::speed = 0`  
Definition at line 1135 of file [rtklib.h](#).

### 10.90.2.13 start

`double file_t::start = 0`  
Definition at line 1134 of file [rtklib.h](#).

### 10.90.2.14 swapintv

`double file_t::swapintv = 0`  
Definition at line 1136 of file [rtklib.h](#).

### 10.90.2.15 tick

`unsigned int file_t::tick = 0`  
Definition at line 1131 of file [rtklib.h](#).

### 10.90.2.16 tick\_f

`unsigned int file_t::tick_f = 0`  
Definition at line 1132 of file [rtklib.h](#).

### 10.90.2.17 time

`gtime_t file_t::time = {}`  
Definition at line 1129 of file [rtklib.h](#).

### 10.90.2.18 timetag

`int file_t::timetag`  
Definition at line 1126 of file [rtklib.h](#).

### 10.90.2.19 wtime

`gtime_t file_t::wtime = {}`  
Definition at line 1130 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

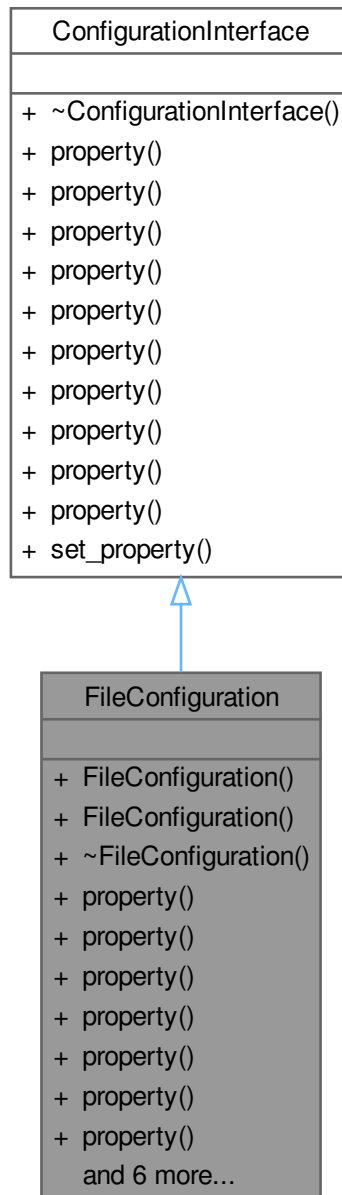
- [rtklib.h](#)

## 10.91 FileConfiguration Class Reference

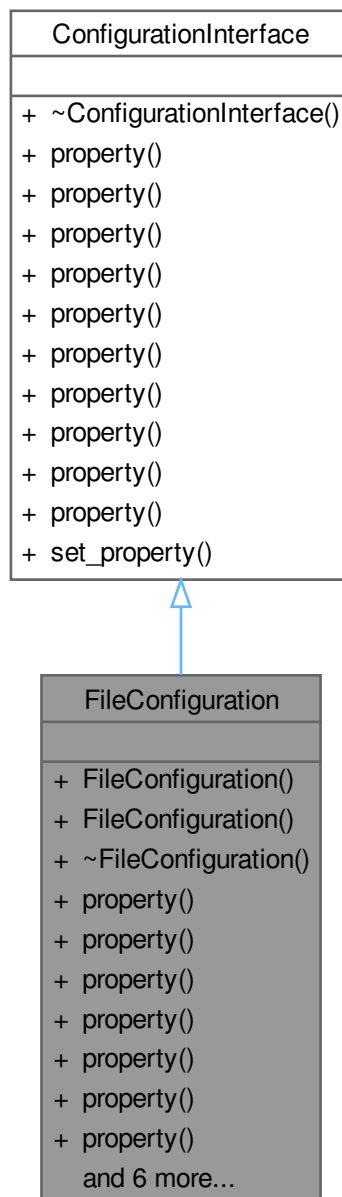
This class is an implementation of the interface [ConfigurationInterface](#).

```
#include <file_configuration.h>
```

Inheritance diagram for FileConfiguration:



Collaboration diagram for FileConfiguration:



### Public Member Functions

- **FileConfiguration** (std::string filename)
- std::string **property** (std::string property\_name, std::string default\_value) const override
- bool **property** (std::string property\_name, bool default\_value) const override
- int64\_t **property** (std::string property\_name, int64\_t default\_value) const override
- uint64\_t **property** (std::string property\_name, uint64\_t default\_value) const override
- int32\_t **property** (std::string property\_name, int32\_t default\_value) const override
- uint32\_t **property** (std::string property\_name, uint32\_t default\_value) const override
- int16\_t **property** (std::string property\_name, int16\_t default\_value) const override

- uint16\_t [property](#) (std::string property\_name, uint16\_t default\_value) const override
- float [property](#) (std::string property\_name, float default\_value) const override
- double [property](#) (std::string property\_name, double default\_value) const override
- void [set\\_property](#) (std::string property\_name, std::string value) override
- bool [is\\_present](#) (const std::string &property\_name) const
- bool [has\\_section](#) () const

### 10.91.1 Detailed Description

This class is an implementation of the interface [ConfigurationInterface](#).

Derived from [ConfigurationInterface](#), this class implements an interface to a configuration file. This implementation has a text file as the source for the values of the parameters. The file is in the INI format, containing sections and pairs of names and values. For more information about the INI format, see [https://en.wikipedia.org/wiki/INI\\_file](https://en.wikipedia.org/wiki/INI_file)

Definition at line 48 of file [file\\_configuration.h](#).

### 10.91.2 Member Function Documentation

#### 10.91.2.1 [property\(\)](#) [1/10]

```
bool FileConfiguration::property (
    std::string property_name,
    bool default_value) const [override], [virtual]
```

Implements [ConfigurationInterface](#).

#### 10.91.2.2 [property\(\)](#) [2/10]

```
double FileConfiguration::property (
    std::string property_name,
    double default_value) const [override], [virtual]
```

Implements [ConfigurationInterface](#).

#### 10.91.2.3 [property\(\)](#) [3/10]

```
float FileConfiguration::property (
    std::string property_name,
    float default_value) const [override], [virtual]
```

Implements [ConfigurationInterface](#).

#### 10.91.2.4 [property\(\)](#) [4/10]

```
int16_t FileConfiguration::property (
    std::string property_name,
    int16_t default_value) const [override], [virtual]
```

Implements [ConfigurationInterface](#).

#### 10.91.2.5 [property\(\)](#) [5/10]

```
int32_t FileConfiguration::property (
    std::string property_name,
    int32_t default_value) const [override], [virtual]
```

Implements [ConfigurationInterface](#).

#### 10.91.2.6 [property\(\)](#) [6/10]

```
int64_t FileConfiguration::property (
    std::string property_name,
    int64_t default_value) const [override], [virtual]
```

Implements [ConfigurationInterface](#).

**10.91.2.7 property()** [7/10]

```
std::string FileConfiguration::property (
    std::string property_name,
    std::string default_value) const [override], [virtual]
```

Implements [ConfigurationInterface](#).

**10.91.2.8 property()** [8/10]

```
uint16_t FileConfiguration::property (
    std::string property_name,
    uint16_t default_value) const [override], [virtual]
```

Implements [ConfigurationInterface](#).

**10.91.2.9 property()** [9/10]

```
uint32_t FileConfiguration::property (
    std::string property_name,
    uint32_t default_value) const [override], [virtual]
```

Implements [ConfigurationInterface](#).

**10.91.2.10 property()** [10/10]

```
uint64_t FileConfiguration::property (
    std::string property_name,
    uint64_t default_value) const [override], [virtual]
```

Implements [ConfigurationInterface](#).

**10.91.2.11 set\_property()**

```
void FileConfiguration::set_property (
    std::string property_name,
    std::string value) [override], [virtual]
```

Implements [ConfigurationInterface](#).

The documentation for this class was generated from the following file:

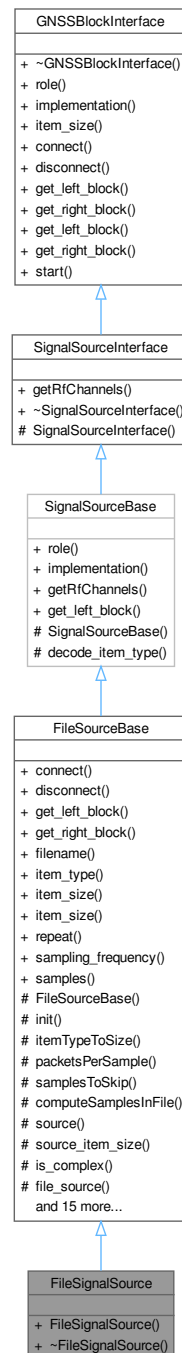
- [file\\_configuration.h](#)

## 10.92 FileSignalSource Class Reference

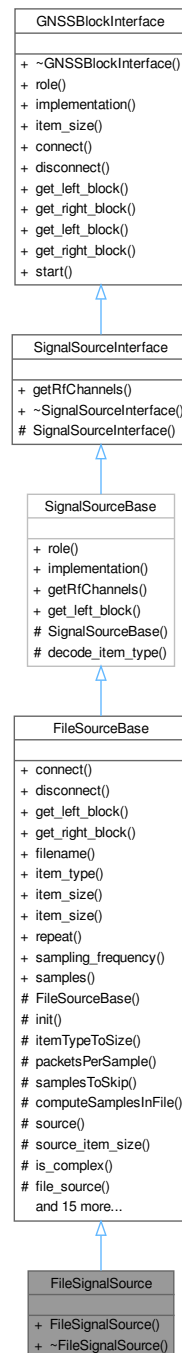
Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).

```
#include <file_signal_source.h>
```

Inheritance diagram for FileSignalSource:



Collaboration diagram for FileSignalSource:



## Public Member Functions

- **FileSignalSource** ([ConfigurationInterface](#) const \*configuration, std::string const &role, unsigned int in\_streams, unsigned int out\_streams, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)

## Public Member Functions inherited from [FileSourceBase](#)

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override

- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- std::string [filename](#) () const  
*The file to read.*
- std::string [item\\_type](#) () const  
*The item type.*
- size\_t [item\\_size](#) () override  
*The configured size of each item.*
- virtual size\_t [item\\_size](#) () const
- bool [repeat](#) () const  
*Whether to repeat reading after end-of-file.*
- int64\_t [sampling\\_frequency](#) () const  
*The sampling frequency of the source file.*
- uint64\_t [samples](#) () const  
*The number of samples in the file.*

### Public Member Functions inherited from [SignalSourceBase](#)

- std::string [role](#) () final
- std::string [implementation](#) () final
- size\_t [getRfChannels](#) () const override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()  
*Start the flow of samples if needed.*

### Additional Inherited Members

### Protected Member Functions inherited from [FileSourceBase](#)

- [FileSourceBase](#) ([ConfigurationInterface](#) const \*configuration, std::string const &role, std::string impl, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue, std::string default\_item\_type="short")  
*Constructor.*
- void [init](#) ()  
*Perform post-construction initialization.*
- virtual std::tuple< size\_t, bool > [itemTypeToSize](#) ()  
*Compute the item size, from the [item\\_type](#)(). Subclasses may constrain types that don't make.*
- virtual double [packetsPerSample](#) () const  
*The number of (possibly unpacked) samples in a (raw) file sample (default=1)*
- virtual size\_t [samplesToSkip](#) () const  
*Compute the number of samples to skip.*
- size\_t [computeSamplesInFile](#) () const  
*Compute the number of samples in the file.*
- virtual gnss\_shared\_ptr< gr::block > [source](#) () const  
*Abstracted front-end source. Sub-classes may override if they create specialized chains to decode source files into a usable format.*
- virtual size\_t [source\\_item\\_size](#) () const  
*For complex source chains, the size of the file item may not be the same as the size of the.*
- bool [is\\_complex](#) () const
- gnss\_shared\_ptr< gr::block > [file\\_source](#) () const



- gnss\_shared\_ptr< gr::block > **valve** () const
- gnss\_shared\_ptr< gr::block > **throttle** () const
- gnss\_shared\_ptr< gr::block > **sink** () const
- gr::blocks::file\_source::sptr **create\_file\_source** ()
- gr::blocks::throttle::sptr **create\_throttle** ()
- gnss\_shared\_ptr< gr::block > **create\_valve** ()
- gr::blocks::file\_sink::sptr **create\_sink** ()
- virtual void **create\_file\_source\_hook** ()
- virtual void **create\_throttle\_hook** ()
- virtual void **create\_valve\_hook** ()
- virtual void **create\_sink\_hook** ()
- virtual void **pre\_connect\_hook** (gr::top\_block\_sptr top\_block)
- virtual void **post\_connect\_hook** (gr::top\_block\_sptr top\_block)
- virtual void **pre\_disconnect\_hook** (gr::top\_block\_sptr top\_block)
- virtual void **post\_disconnect\_hook** (gr::top\_block\_sptr top\_block)

### Protected Member Functions inherited from [SignalSourceBase](#)

- [SignalSourceBase](#) ([ConfigurationInterface](#) const \*configuration, std::string role, std::string impl)  
*Constructor.*
- size\_t [decode\\_item\\_type](#) (std::string const &item\_type, bool \*is\_interleaved=nullptr, bool throw\_on\_↵  
error=false)  
*utility for decoding passed ".item\_type" values*

#### 10.92.1 Detailed Description

Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).

Definition at line 41 of file [file\\_signal\\_source.h](#).

The documentation for this class was generated from the following file:

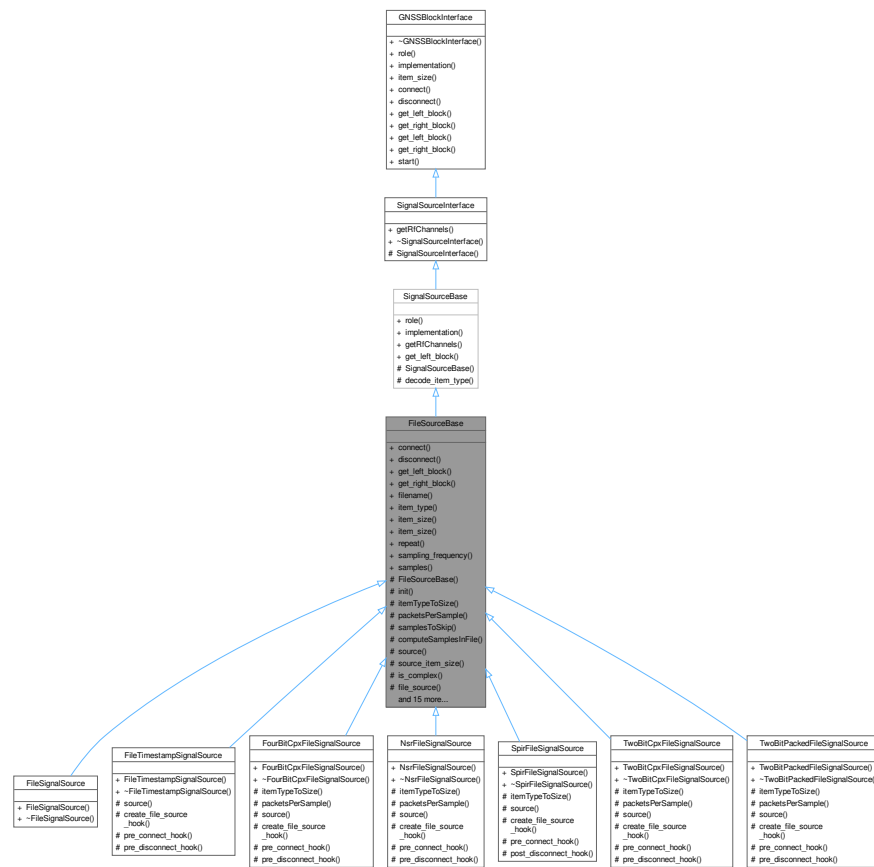
- [file\\_signal\\_source.h](#)

## 10.93 FileSourceBase Class Reference

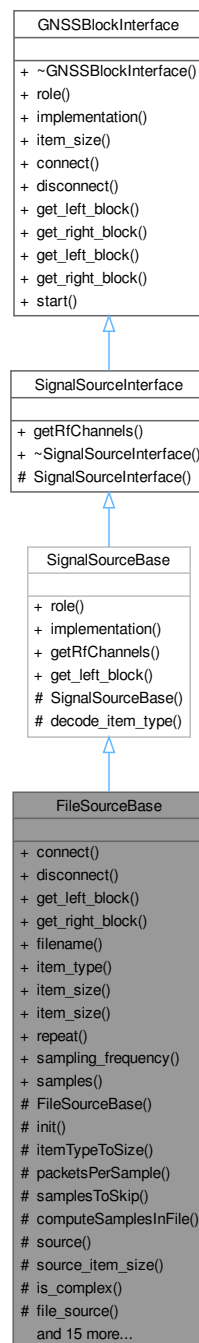
Base class to file-oriented [SignalSourceBase](#) GNSS blocks.

```
#include <file_source_base.h>
```

Inheritance diagram for FileSourceBase:



Collaboration diagram for FileSourceBase:



## Public Member Functions

- void `connect` (`gr::top_block_sptr top_block`) override
- void `disconnect` (`gr::top_block_sptr top_block`) override
- `gr::basic_block_sptr` `get_left_block` () override
- `gr::basic_block_sptr` `get_right_block` () override
- `std::string` `filename` () const

*The file to read.*

- `std::string item_type () const`  
*The item type.*
- `size_t item_size () override`  
*The configured size of each item.*
- `virtual size_t item_size () const`
- `bool repeat () const`  
*Whether to repeat reading after end-of-file.*
- `int64_t sampling_frequency () const`  
*The sampling frequency of the source file.*
- `uint64_t samples () const`  
*The number of samples in the file.*

## Public Member Functions inherited from [SignalSourceBase](#)

- `std::string role () final`
- `std::string implementation () final`
- `size_t getRfChannels () const override`
- `gr::basic_block_sptr get_left_block () override`

## Public Member Functions inherited from [GNSSBlockInterface](#)

- `virtual gr::basic_block_sptr get_left_block (int RF_channel)`
- `virtual gr::basic_block_sptr get_right_block (int RF_channel)`
- `virtual void start ()`  
*Start the flow of samples if needed.*

## Protected Member Functions

- [FileSourceBase](#) ([ConfigurationInterface](#) const \*configuration, std::string const &role, std::string impl, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue, std::string default\_item\_type="short")  
*Constructor.*
- `void init ()`  
*Perform post-construction initialization.*
- `virtual std::tuple< size_t, bool > itemTypeToSize ()`  
*Compute the item size, from the [item\\_type\(\)](#). Subclasses may constrain types that don't make.*
- `virtual double packetsPerSample () const`  
*The number of (possibly unpacked) samples in a (raw) file sample (default=1)*
- `virtual size_t samplesToSkip () const`  
*Compute the number of samples to skip.*
- `size_t computeSamplesInFile () const`  
*Compute the number of samples in the file.*
- `virtual gnss_shared_ptr< gr::block > source () const`  
*Abstracted front-end source. Sub-classes may override if they create specialized chains to decode source files into a usable format.*
- `virtual size_t source_item_size () const`  
*For complex source chains, the size of the file item may not be the same as the size of the.*
- `bool is_complex () const`
- `gnss_shared_ptr< gr::block > file_source () const`
- `gnss_shared_ptr< gr::block > valve () const`
- `gnss_shared_ptr< gr::block > throttle () const`
- `gnss_shared_ptr< gr::block > sink () const`
- `gr::blocks::file_source::sptr create_file_source ()`
- `gr::blocks::throttle::sptr create_throttle ()`

- gnss\_shared\_ptr< gr::block > **create\_valve** ()
- gr::blocks::file\_sink::sptr **create\_sink** ()
- virtual void **create\_file\_source\_hook** ()
- virtual void **create\_throttle\_hook** ()
- virtual void **create\_valve\_hook** ()
- virtual void **create\_sink\_hook** ()
- virtual void **pre\_connect\_hook** (gr::top\_block\_sptr top\_block)
- virtual void **post\_connect\_hook** (gr::top\_block\_sptr top\_block)
- virtual void **pre\_disconnect\_hook** (gr::top\_block\_sptr top\_block)
- virtual void **post\_disconnect\_hook** (gr::top\_block\_sptr top\_block)

## Protected Member Functions inherited from [SignalSourceBase](#)

- [SignalSourceBase](#) ([ConfigurationInterface](#) const \*configuration, std::string role, std::string impl)  
*Constructor.*
- size\_t [decode\\_item\\_type](#) (std::string const &item\_type, bool \*is\_interleaved=nullptr, bool throw\_on\_↔ error=false)  
*utility for decoding passed ".item\_type" values*

### 10.93.1 Detailed Description

Base class to file-oriented [SignalSourceBase](#) GNSS blocks.

This class supports the following properties:

.filename - the path to the input file

- may be overridden by the -signal\_source or -s command-line arguments

.samples - number of samples to process (default 0)

- if not specified or 0, read the entire file; otherwise stop after that many samples

.sampling\_frequency - the frequency of the sampled data (samples/second)

.item\_type - data type of the samples (default "short")

.header\_size - the size of a prefixed header to skip in "samples" (default 0)

.seconds\_to\_skip - number of seconds of lead-in data to skip over (default 0)

.enable\_throttle\_control - whether to stop reading if the upstream buffer is full (default false)

.repeat - whether to rewind and continue at end of file (default false)

(probably abstracted to the base class)

.dump - whether to archive input data

.dump\_filename - if dumping, path to file for output

Definition at line 66 of file [file\\_source\\_base.h](#).

### 10.93.2 Constructor & Destructor Documentation

#### 10.93.2.1 FileSourceBase()

```
FileSourceBase::FileSourceBase (
    ConfigurationInterface const * configuration,
    std::string const & role,
    std::string impl,
    Concurrent\_Queue< pmt::pmt_t > * queue,
    std::string default_item_type = "short") [protected]
```

Constructor.

Subclasses may want to assert default item types that are appropriate to the specific file type supported. Rather than require the item type to be specified in the config file, allow sub-classes to impose their will

### 10.93.3 Member Function Documentation

#### 10.93.3.1 computeSamplesInFile()

```
size_t FileSourceBase::computeSamplesInFile () const [protected]
```

Compute the number of samples in the file.

**10.93.3.2 connect()**

```
void FileSourceBase::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

**10.93.3.3 disconnect()**

```
void FileSourceBase::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

**10.93.3.4 filename()**

```
std::string FileSourceBase::filename () const
```

The file to read.

**10.93.3.5 get\_left\_block()**

```
gr::basic_block_sptr FileSourceBase::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

**10.93.3.6 get\_right\_block()**

```
gr::basic_block_sptr FileSourceBase::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

**10.93.3.7 init()**

```
void FileSourceBase::init () [protected]
```

Perform post-construction initialization.

**10.93.3.8 item\_size()**

```
size_t FileSourceBase::item_size () [override], [virtual]
```

The configured size of each item.

Implements [GNSSBlockInterface](#).

**10.93.3.9 item\_type()**

```
std::string FileSourceBase::item_type () const
```

The item type.

**10.93.3.10 itemTypeToSize()**

```
virtual std::tuple< size_t, bool > FileSourceBase::itemTypeToSize () [protected], [virtual]
```

Compute the item size, from the [item\\_type\(\)](#). Subclasses may constrain types that don't make.

Reimplemented in [FourBitCpxFileSignalSource](#), [NsrFileSignalSource](#), [SpirFileSignalSource](#), [TwoBitCpxFileSignalSource](#), and [TwoBitPackedFileSignalSource](#).

**10.93.3.11 packetsPerSample()**

```
virtual double FileSourceBase::packetsPerSample () const [protected], [virtual]
```

The number of (possibly unpacked) samples in a (raw) file sample (default=1)

Reimplemented in [FourBitCpxFileSignalSource](#), [NsrFileSignalSource](#), [TwoBitCpxFileSignalSource](#), and [TwoBitPackedFileSignalSource](#).

**10.93.3.12 repeat()**

```
bool FileSourceBase::repeat () const
```

Whether to repeat reading after end-of-file.

### 10.93.3.13 samples()

```
uint64_t FileSourceBase::samples () const
```

The number of samples in the file.

### 10.93.3.14 samplesToSkip()

```
virtual size_t FileSourceBase::samplesToSkip () const [protected], [virtual]
```

Compute the number of samples to skip.

### 10.93.3.15 sampling\_frequency()

```
int64_t FileSourceBase::sampling_frequency () const
```

The sampling frequency of the source file.

### 10.93.3.16 source()

```
virtual gnss_shared_ptr< gr::block > FileSourceBase::source () const [protected], [virtual]
```

Abstracted front-end source. Sub-classes may override if they create specialized chains to decode source files into a usable format.

Reimplemented in [FileTimestampSignalSource](#), [FourBitCpxFileSignalSource](#), [NsrFileSignalSource](#), [SpirFileSignalSource](#), [TwoBitCpxFileSignalSource](#), and [TwoBitPackedFileSignalSource](#).

### 10.93.3.17 source\_item\_size()

```
virtual size_t FileSourceBase::source_item_size () const [protected], [virtual]
```

For complex source chains, the size of the file item may not be the same as the size of the.

The documentation for this class was generated from the following file:

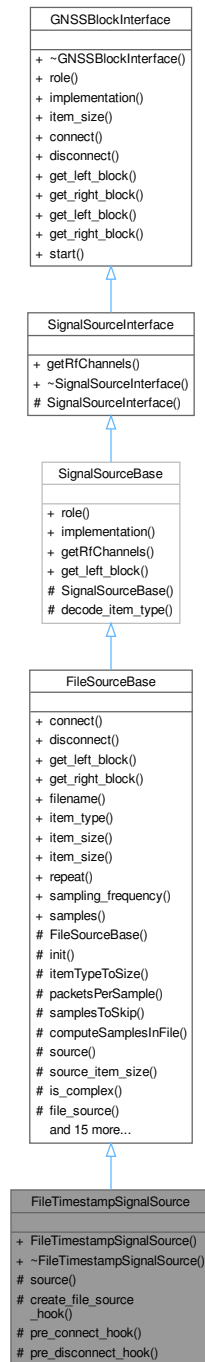
- [file\\_source\\_base.h](#)

## 10.94 FileTimestampSignalSource Class Reference

Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).

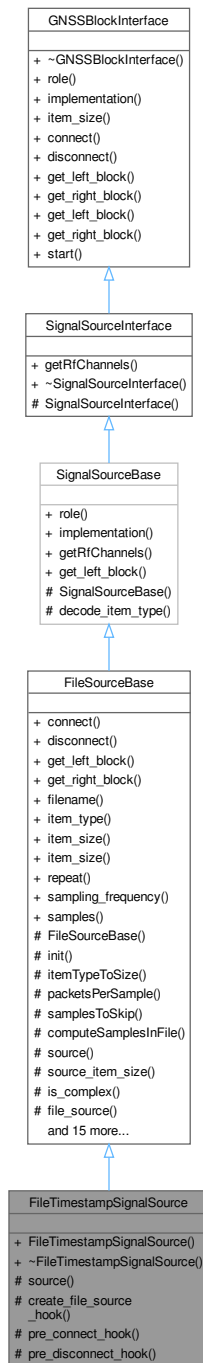
```
#include <file_timestamp_signal_source.h>
```

Inheritance diagram for FileTimestampSignalSource:





Collaboration diagram for FileTimestampSignalSource:



## Public Member Functions

- **FileTimestampSignalSource** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)

## Public Member Functions inherited from [FileSourceBase](#)

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override

- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- std::string [filename](#) () const  
*The file to read.*
- std::string [item\\_type](#) () const  
*The item type.*
- size\_t [item\\_size](#) () override  
*The configured size of each item.*
- virtual size\_t [item\\_size](#) () const
- bool [repeat](#) () const  
*Whether to repeat reading after end-of-file.*
- int64\_t [sampling\\_frequency](#) () const  
*The sampling frequency of the source file.*
- uint64\_t [samples](#) () const  
*The number of samples in the file.*

### Public Member Functions inherited from [SignalSourceBase](#)

- std::string [role](#) () final
- std::string [implementation](#) () final
- size\_t [getRfChannels](#) () const override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()  
*Start the flow of samples if needed.*

### Protected Member Functions

- gnss\_shared\_ptr< gr::block > [source](#) () const override  
*Abstracted front-end source. Sub-classes may override if they create specialized chains to decode source files into a usable format.*
- void [create\\_file\\_source\\_hook](#) () override
- void [pre\\_connect\\_hook](#) (gr::top\_block\_sptr top\_block) override
- void [pre\\_disconnect\\_hook](#) (gr::top\_block\_sptr top\_block) override

### Protected Member Functions inherited from [FileSourceBase](#)

- [FileSourceBase](#) ([ConfigurationInterface](#) const \*configuration, std::string const &role, std::string impl, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue, std::string default\_item\_type="short")  
*Constructor.*
- void [init](#) ()  
*Perform post-construction initialization.*
- virtual std::tuple< size\_t, bool > [itemTypeToSize](#) ()  
*Compute the item size, from the [item\\_type\(\)](#). Subclasses may constrain types that don't make.*
- virtual double [packetsPerSample](#) () const  
*The number of (possibly unpacked) samples in a (raw) file sample (default=1)*
- virtual size\_t [samplesToSkip](#) () const  
*Compute the number of samples to skip.*
- size\_t [computeSamplesInFile](#) () const  
*Compute the number of samples in the file.*

- virtual size\_t [source\\_item\\_size](#) () const  
*For complex source chains, the size of the file item may not be the same as the size of the.*
- bool **is\_complex** () const
- gnss\_shared\_ptr< gr::block > **file\_source** () const
- gnss\_shared\_ptr< gr::block > **valve** () const
- gnss\_shared\_ptr< gr::block > **throttle** () const
- gnss\_shared\_ptr< gr::block > **sink** () const
- gr::blocks::file\_source::sptr **create\_file\_source** ()
- gr::blocks::throttle::sptr **create\_throttle** ()
- gnss\_shared\_ptr< gr::block > **create\_valve** ()
- gr::blocks::file\_sink::sptr **create\_sink** ()
- virtual void **create\_throttle\_hook** ()
- virtual void **create\_valve\_hook** ()
- virtual void **create\_sink\_hook** ()
- virtual void **post\_connect\_hook** (gr::top\_block\_sptr top\_block)
- virtual void **post\_disconnect\_hook** (gr::top\_block\_sptr top\_block)

### Protected Member Functions inherited from [SignalSourceBase](#)

- [SignalSourceBase](#) ([ConfigurationInterface](#) const \*configuration, std::string role, std::string impl)  
*Constructor.*
- size\_t [decode\\_item\\_type](#) (std::string const &item\_type, bool \*is\_interleaved=nullptr, bool throw\_on\_↵ error=false)  
*utility for decoding passed ".item\_type" values*

#### 10.94.1 Detailed Description

Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).  
Definition at line 37 of file [file\\_timestamp\\_signal\\_source.h](#).

#### 10.94.2 Member Function Documentation

##### 10.94.2.1 [create\\_file\\_source\\_hook](#)()

```
void FileTimestampSignalSource::create_file_source_hook () [override], [protected], [virtual]
```

Reimplemented from [FileSourceBase](#).

##### 10.94.2.2 [pre\\_connect\\_hook](#)()

```
void FileTimestampSignalSource::pre_connect_hook (
    gr::top_block_sptr top_block) [override], [protected], [virtual]
```

Reimplemented from [FileSourceBase](#).

##### 10.94.2.3 [pre\\_disconnect\\_hook](#)()

```
void FileTimestampSignalSource::pre_disconnect_hook (
    gr::top_block_sptr top_block) [override], [protected], [virtual]
```

Reimplemented from [FileSourceBase](#).

##### 10.94.2.4 [source](#)()

```
gnss_shared_ptr< gr::block > FileTimestampSignalSource::source () const [override], [protected], [virtual]
```

Abstracted front-end source. Sub-classes may override if they create specialized chains to decode source files into a usable format.

Reimplemented from [FileSourceBase](#).

The documentation for this class was generated from the following file:

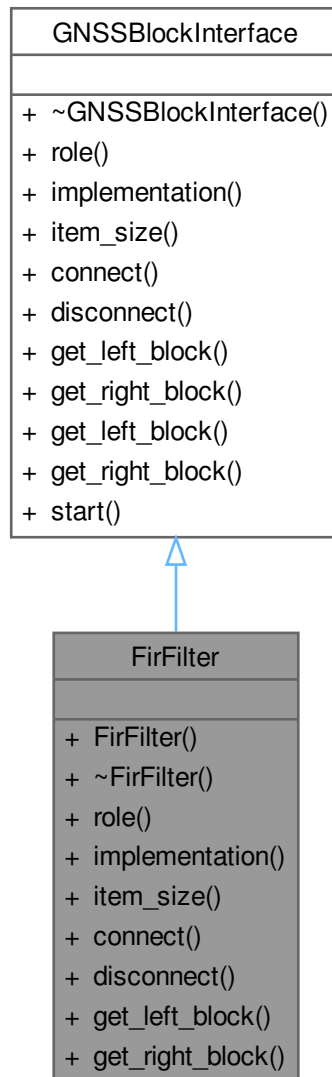
- [file\\_timestamp\\_signal\\_source.h](#)

## 10.95 FirFilter Class Reference

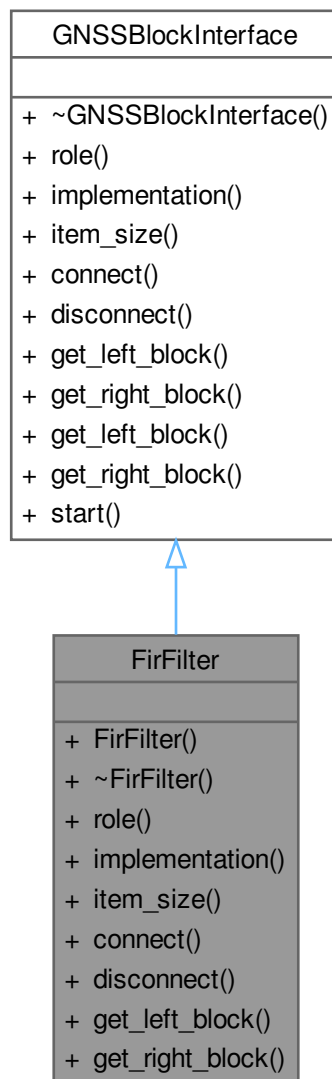
This class adapts a GNU Radio `gr_fir_filter` designed with `pm_remez`.

```
#include <fir_filter.h>
```

Inheritance diagram for `FirFilter`:



Collaboration diagram for FirFilter:



## Public Member Functions

- `FirFilter` (const `ConfigurationInterface` \*configuration, std::string role, unsigned int in\_streams, unsigned int out\_streams)  
*Constructor.*
- `~FirFilter` ()=default  
*Destructor.*
- std::string `role` () override
- std::string `implementation` () override  
*Returns "Fir\_Filter".*
- size\_t `item_size` () override
- void `connect` (gr::top\_block\_sptr top\_block) override
- void `disconnect` (gr::top\_block\_sptr top\_block) override

- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.95.1 Detailed Description

This class adapts a GNU Radio `gr_fir_filter` designed with `pm_remez`.

See Parks-McClellan FIR filter design, [https://en.wikipedia.org/wiki/Parks-McClellan\\_filter\\_design\\_algorithm](https://en.wikipedia.org/wiki/Parks-McClellan_filter_design_algorithm) Calculates the optimal (in the Chebyshev/minimax sense) FIR filter impulse response given a set of band edges, the desired response on those bands, and the weight given to the error in those bands.

Definition at line 59 of file [fir\\_filter.h](#).

### 10.95.2 Constructor & Destructor Documentation

#### 10.95.2.1 FirFilter()

```
FirFilter::FirFilter (
    const ConfigurationInterface * configuration,
    std::string role,
    unsigned int in_streams,
    unsigned int out_streams)
```

Constructor.

#### 10.95.2.2 ~FirFilter()

```
FirFilter::~~FirFilter () [default]
```

Destructor.

### 10.95.3 Member Function Documentation

#### 10.95.3.1 connect()

```
void FirFilter::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.95.3.2 disconnect()

```
void FirFilter::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.95.3.3 get\_left\_block()

```
gr::basic_block_sptr FirFilter::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.95.3.4 get\_right\_block()

```
gr::basic_block_sptr FirFilter::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.95.3.5 implementation()

```
std::string FirFilter::implementation () [inline], [override], [virtual]
```

Returns "Fir\_Filter".

Implements [GNSSBlockInterface](#).

Definition at line 77 of file [fir\\_filter.h](#).

### 10.95.3.6 item\_size()

```
size_t FirFilter::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 82 of file [fir\\_filter.h](#).

### 10.95.3.7 role()

```
std::string FirFilter::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 71 of file [fir\\_filter.h](#).

The documentation for this class was generated from the following file:

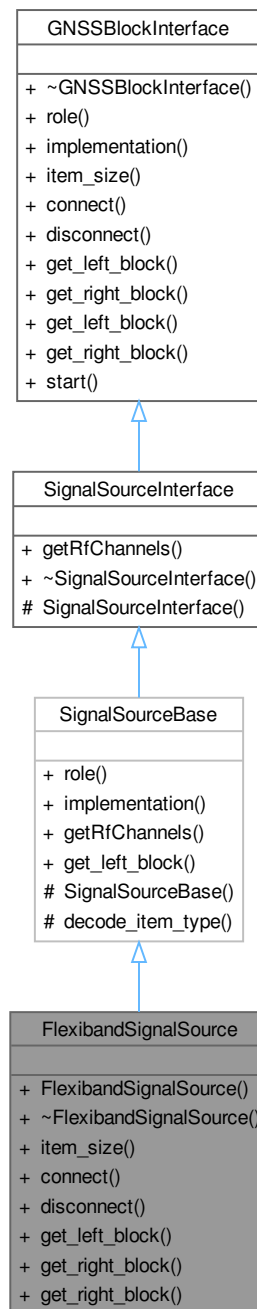
- [fir\\_filter.h](#)

## 10.96 FlexibandSignalSource Class Reference

This class configures and reads samples from Teleorbit Flexiband front-end. This software requires a Flexiband GNU Radio driver installed (not included with GNSS-SDR).

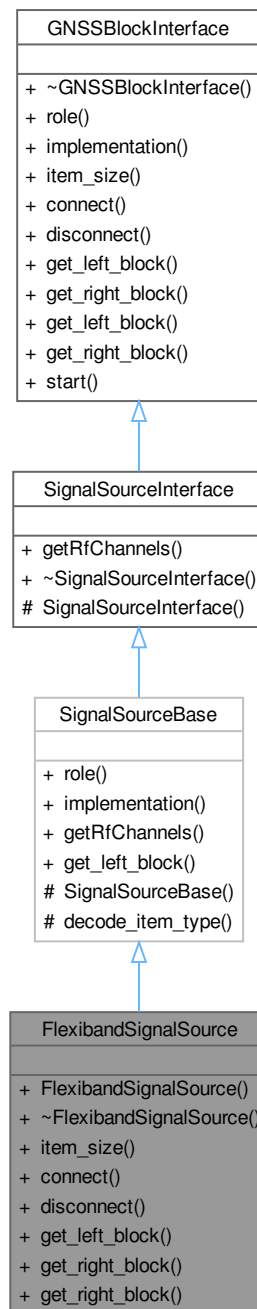
```
#include <flexiband_signal_source.h>
```

Inheritance diagram for FlexibandSignalSource:





Collaboration diagram for FlexibandSignalSource:



## Public Member Functions

- **FlexibandSignalSource** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_stream, unsigned int out\_stream, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override

- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel) override

### Public Member Functions inherited from [SignalSourceBase](#)

- std::string [role](#) () final
- std::string [implementation](#) () final
- size\_t [getRfChannels](#) () const override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### Additional Inherited Members

### Protected Member Functions inherited from [SignalSourceBase](#)

- [SignalSourceBase](#) ([ConfigurationInterface](#) const \*configuration, std::string role, std::string impl)  
*Constructor.*
- size\_t [decode\\_item\\_type](#) (std::string const &item\_type, bool \*is\_interleaved=nullptr, bool throw\_on\_↔ error=false)  
*utility for decoding passed ".item\_type" values*

## 10.96.1 Detailed Description

This class configures and reads samples from Teleorbit Flexiband front-end. This software requires a Flexiband GNU Radio driver installed (not included with GNSS-SDR).  
Definition at line 48 of file [flexiband\\_signal\\_source.h](#).

## 10.96.2 Member Function Documentation

### 10.96.2.1 connect()

```
void FlexibandSignalSource::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.96.2.2 disconnect()

```
void FlexibandSignalSource::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.96.2.3 get\_left\_block()

```
gr::basic_block_sptr FlexibandSignalSource::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.96.2.4 get\_right\_block() [1/2]

```
gr::basic_block_sptr FlexibandSignalSource::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.96.2.5 `get_right_block()` [2/2]

```
gr::basic_block_sptr FlexibandSignalSource::get_right_block (  
    int RF_channel) [override], [virtual]
```

Reimplemented from [GNSSBlockInterface](#).

### 10.96.2.6 `item_size()`

```
size_t FlexibandSignalSource::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

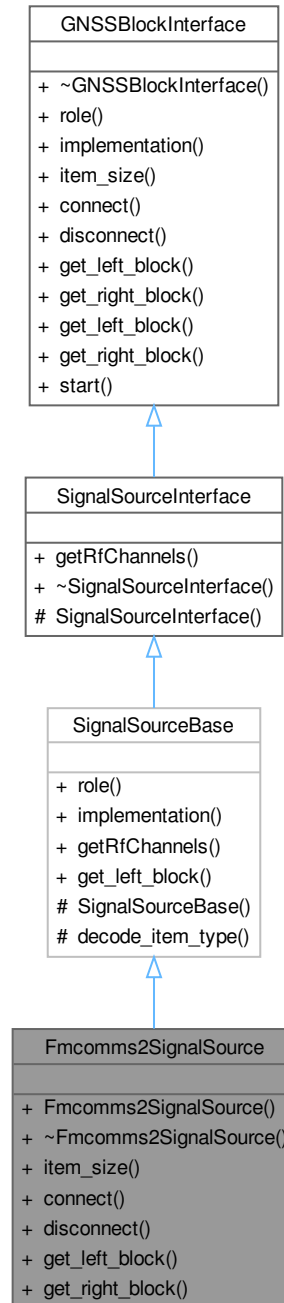
Definition at line 57 of file [flexiband\\_signal\\_source.h](#).

The documentation for this class was generated from the following file:

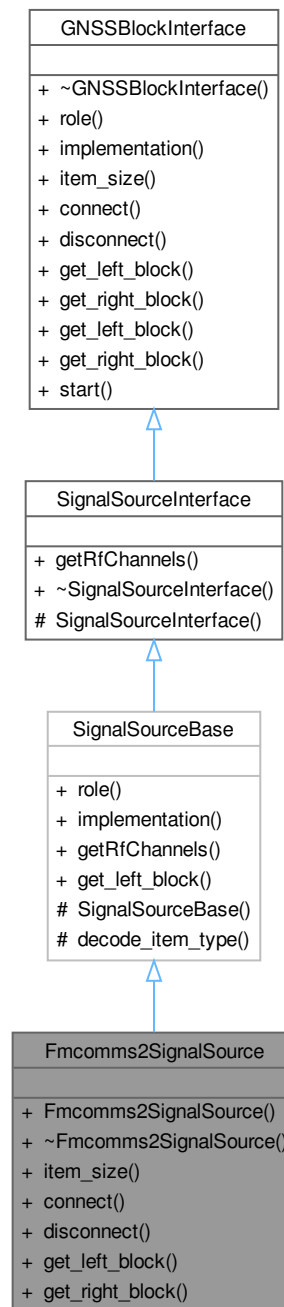
- [flexiband\\_signal\\_source.h](#)

## 10.97 Fmcomms2SignalSource Class Reference

Inheritance diagram for Fmcomms2SignalSource:



Collaboration diagram for Fmcomms2SignalSource:



### Public Member Functions

- **Fmcomms2SignalSource** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_stream, unsigned int out\_stream, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override

## Public Member Functions inherited from [SignalSourceBase](#)

- `std::string` [role](#) () final
- `std::string` [implementation](#) () final
- `size_t` [getRfChannels](#) () const override
- `gr::basic_block_sptr` [get\\_left\\_block](#) () override

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual `gr::basic_block_sptr` [get\\_left\\_block](#) (int RF\_channel)
- virtual `gr::basic_block_sptr` [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

## Additional Inherited Members

## Protected Member Functions inherited from [SignalSourceBase](#)

- [SignalSourceBase](#) ([ConfigurationInterface](#) const \*configuration, `std::string` role, `std::string` impl)  
*Constructor.*
- `size_t` [decode\\_item\\_type](#) (`std::string` const &item\_type, `bool` \*is\_interleaved=nullptr, `bool` throw\_on\_↵  
error=false)  
*utility for decoding passed ".item\_type" values*

### 10.97.1 Detailed Description

Definition at line 44 of file [fmcomms2\\_signal\\_source.h](#).

### 10.97.2 Member Function Documentation

#### 10.97.2.1 connect()

```
void Fmcomms2SignalSource::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.97.2.2 disconnect()

```
void Fmcomms2SignalSource::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.97.2.3 get\_left\_block()

```
gr::basic_block_sptr Fmcomms2SignalSource::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.97.2.4 get\_right\_block()

```
gr::basic_block_sptr Fmcomms2SignalSource::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.97.2.5 item\_size()

```
size_t Fmcomms2SignalSource::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

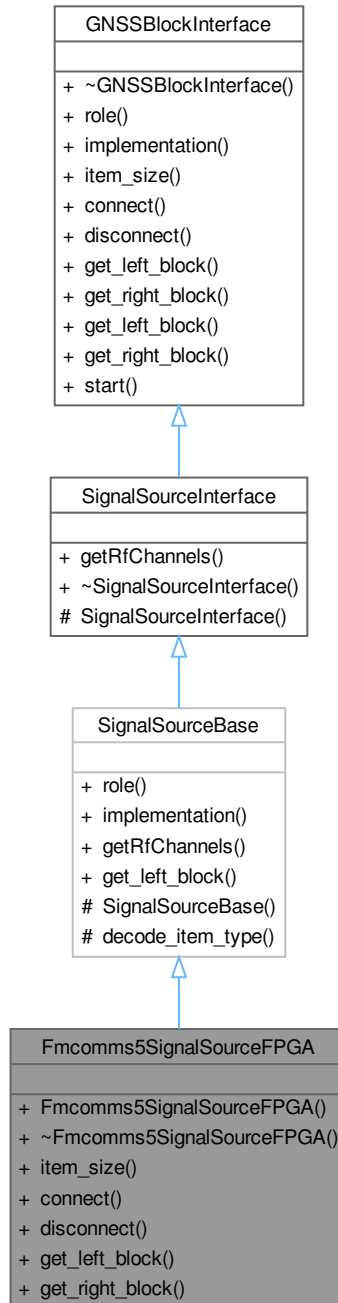
Definition at line 53 of file [fmcomms2\\_signal\\_source.h](#).

The documentation for this class was generated from the following file:

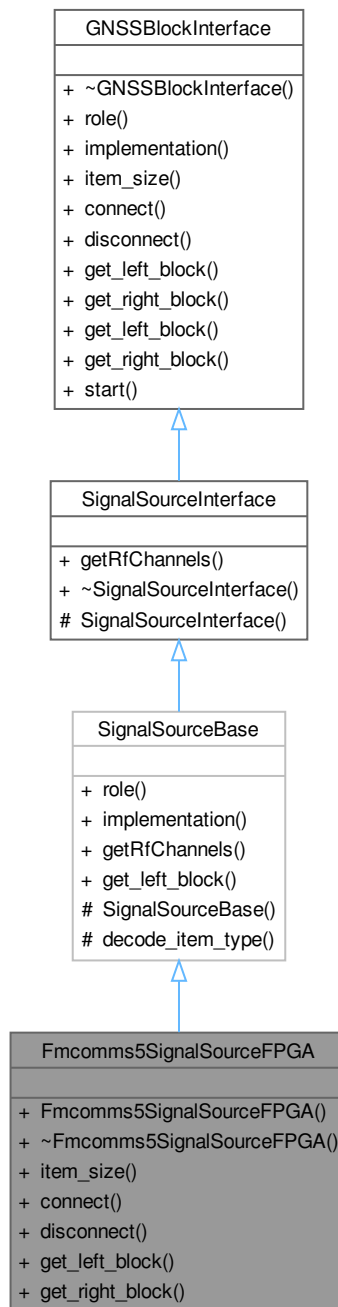
- [fmcomms2\\_signal\\_source.h](#)

## 10.98 Fmcomms5SignalSourceFPGA Class Reference

Inheritance diagram for Fmcomms5SignalSourceFPGA:



Collaboration diagram for Fmcomms5SignalSourceFPGA:



## Public Member Functions

- **Fmcomms5SignalSourceFPGA** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_stream, unsigned int out\_stream, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override



## Public Member Functions inherited from [SignalSourceBase](#)

- `std::string` [role](#) () final
- `std::string` [implementation](#) () final
- `size_t` [getRfChannels](#) () const override
- `gr::basic_block_sptr` [get\\_left\\_block](#) () override

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual `gr::basic_block_sptr` [get\\_left\\_block](#) (int RF\_channel)
- virtual `gr::basic_block_sptr` [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

## Additional Inherited Members

## Protected Member Functions inherited from [SignalSourceBase](#)

- [SignalSourceBase](#) ([ConfigurationInterface](#) const \*configuration, `std::string` role, `std::string` impl)  
*Constructor.*
- `size_t` [decode\\_item\\_type](#) (`std::string` const &item\_type, `bool` \*is\_interleaved=nullptr, `bool` throw\_on\_↵ error=false)  
*utility for decoding passed ".item\_type" values*

### 10.98.1 Detailed Description

Definition at line 46 of file [fmcomms5\\_signal\\_source\\_fpga.h](#).

### 10.98.2 Member Function Documentation

#### 10.98.2.1 connect()

```
void Fmcomms5SignalSourceFPGA::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.98.2.2 disconnect()

```
void Fmcomms5SignalSourceFPGA::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.98.2.3 get\_left\_block()

```
gr::basic_block_sptr Fmcomms5SignalSourceFPGA::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.98.2.4 get\_right\_block()

```
gr::basic_block_sptr Fmcomms5SignalSourceFPGA::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.98.2.5 item\_size()

```
size_t Fmcomms5SignalSourceFPGA::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 55 of file [fmcomms5\\_signal\\_source\\_fpga.h](#).

The documentation for this class was generated from the following file:

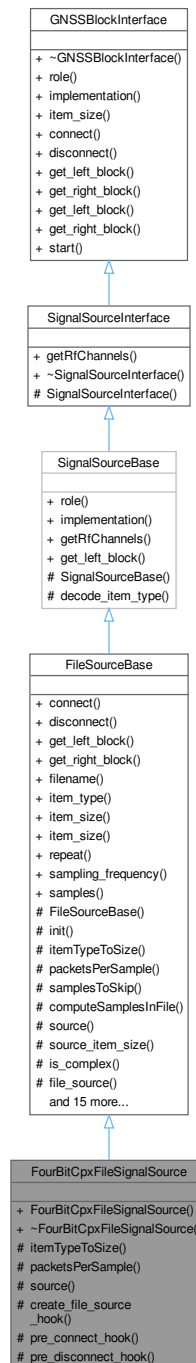
- [fmcomms5\\_signal\\_source\\_fpga.h](#)

## 10.99 FourBitCpxFileSignalSource Class Reference

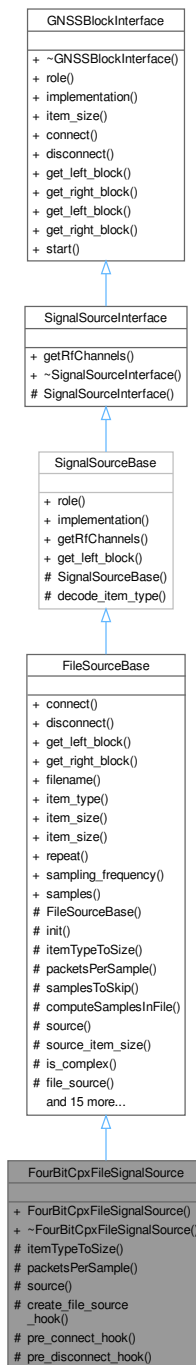
Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).

```
#include <four_bit_cpx_file_signal_source.h>
```

Inheritance diagram for FourBitCpxFileSignalSource:



Collaboration diagram for FourBitCpxFileSignalSource:



## Public Member Functions

- **FourBitCpxFileSignalSource** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)

## Public Member Functions inherited from [FileSourceBase](#)

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override

- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- std::string [filename](#) () const  
*The file to read.*
- std::string [item\\_type](#) () const  
*The item type.*
- size\_t [item\\_size](#) () override  
*The configured size of each item.*
- virtual size\_t [item\\_size](#) () const
- bool [repeat](#) () const  
*Whether to repeat reading after end-of-file.*
- int64\_t [sampling\\_frequency](#) () const  
*The sampling frequency of the source file.*
- uint64\_t [samples](#) () const  
*The number of samples in the file.*

### Public Member Functions inherited from [SignalSourceBase](#)

- std::string [role](#) () final
- std::string [implementation](#) () final
- size\_t [getRfChannels](#) () const override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()  
*Start the flow of samples if needed.*

### Protected Member Functions

- std::tuple< size\_t, bool > [itemTypeToSize](#) () override  
*Compute the item size, from the [item\\_type](#)(). Subclasses may constrain types that don't make.*
- double [packetsPerSample](#) () const override  
*The number of (possibly unpacked) samples in a (raw) file sample (default=1)*
- gnss\_shared\_ptr< gr::block > [source](#) () const override  
*Abstracted front-end source. Sub-classes may override if they create specialized chains to decode source files into a usable format.*
- void [create\\_file\\_source\\_hook](#) () override
- void [pre\\_connect\\_hook](#) (gr::top\_block\_sptr top\_block) override
- void [pre\\_disconnect\\_hook](#) (gr::top\_block\_sptr top\_block) override

### Protected Member Functions inherited from [FileSourceBase](#)

- [FileSourceBase](#) ([ConfigurationInterface](#) const \*configuration, std::string const &role, std::string impl, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue, std::string default\_item\_type="short")  
*Constructor.*
- void [init](#) ()  
*Perform post-construction initialization.*
- virtual size\_t [samplesToSkip](#) () const  
*Compute the number of samples to skip.*
- size\_t [computeSamplesInFile](#) () const  
*Compute the number of samples in the file.*

- virtual size\_t [source\\_item\\_size](#) () const

*For complex source chains, the size of the file item may not be the same as the size of the.*

- bool **is\_complex** () const
- gnss\_shared\_ptr< gr::block > **file\_source** () const
- gnss\_shared\_ptr< gr::block > **valve** () const
- gnss\_shared\_ptr< gr::block > **throttle** () const
- gnss\_shared\_ptr< gr::block > **sink** () const
- gr::blocks::file\_source::sptr **create\_file\_source** ()
- gr::blocks::throttle::sptr **create\_throttle** ()
- gnss\_shared\_ptr< gr::block > **create\_valve** ()
- gr::blocks::file\_sink::sptr **create\_sink** ()
- virtual void **create\_throttle\_hook** ()
- virtual void **create\_valve\_hook** ()
- virtual void **create\_sink\_hook** ()
- virtual void **post\_connect\_hook** (gr::top\_block\_sptr top\_block)
- virtual void **post\_disconnect\_hook** (gr::top\_block\_sptr top\_block)

## Protected Member Functions inherited from [SignalSourceBase](#)

- [SignalSourceBase](#) ([ConfigurationInterface](#) const \*configuration, std::string role, std::string impl)  
*Constructor.*
- size\_t [decode\\_item\\_type](#) (std::string const &item\_type, bool \*is\_interleaved=nullptr, bool throw\_on\_↔ error=false)  
*utility for decoding passed ".item\_type" values*

### 10.99.1 Detailed Description

Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).  
Definition at line 43 of file [four\\_bit\\_cpx\\_file\\_signal\\_source.h](#).

### 10.99.2 Member Function Documentation

#### 10.99.2.1 create\_file\_source\_hook()

void FourBitCpxFileSignalSource::create\_file\_source\_hook () [override], [protected], [virtual]  
Reimplemented from [FileSourceBase](#).

#### 10.99.2.2 itemTypeToSize()

std::tuple< size\_t, bool > FourBitCpxFileSignalSource::itemTypeToSize () [override], [protected], [virtual]

Compute the item size, from the [item\\_type](#)(). Subclasses may constrain types that don't make.  
Reimplemented from [FileSourceBase](#).

#### 10.99.2.3 packetsPerSample()

double FourBitCpxFileSignalSource::packetsPerSample () const [override], [protected], [virtual]  
The number of (possibly unpacked) samples in a (raw) file sample (default=1)  
Reimplemented from [FileSourceBase](#).

#### 10.99.2.4 pre\_connect\_hook()

void FourBitCpxFileSignalSource::pre\_connect\_hook (gr::top\_block\_sptr top\_block) [override], [protected], [virtual]  
Reimplemented from [FileSourceBase](#).

### 10.99.2.5 pre\_disconnect\_hook()

```
void FourBitCpxFileSignalSource::pre_disconnect_hook (
    gr::top_block_sptr top_block) [override], [protected], [virtual]
```

Reimplemented from [FileSourceBase](#).

### 10.99.2.6 source()

```
gnss_shared_ptr< gr::block > FourBitCpxFileSignalSource::source () const [override], [protected],
[virtual]
```

Abstracted front-end source. Sub-classes may override if they create specialized chains to decode source files into a usable format.

Reimplemented from [FileSourceBase](#).

The documentation for this class was generated from the following file:

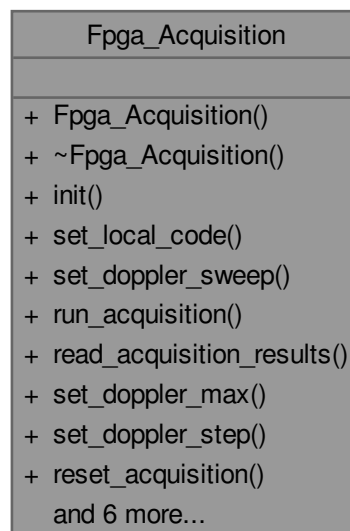
- [four\\_bit\\_cpx\\_file\\_signal\\_source.h](#)

## 10.100 Fpga\_Acquisition Class Reference

Class that implements carrier wipe-off and correlators.

```
#include <fpga_acquisition.h>
```

Collaboration diagram for Fpga\_Acquisition:



### Public Member Functions

- [Fpga\\_Acquisition](#) (std::string device\_name, uint32\_t select\_queue, std::vector< std::pair< uint32\_t, uint32\_t > > &downsampling\_filter\_specs, uint32\_t &max\_FFT\_size)  
*Constructor.*
- [~Fpga\\_Acquisition](#) ()=default  
*Destructor.*
- void [init](#) (uint32\_t nsamples, uint32\_t doppler\_max, uint32\_t d\_fft\_size, int64\_t resampled\_fs, uint32\_t downsampling\_filter\_num, uint32\_t excludelimit, uint32\_t \*all\_fft\_codes)  
*Initialize acquisition parameters.*

- bool [set\\_local\\_code](#) (uint32\_t PRN)  
*Select the code with the chosen PRN.*
- void [set\\_doppler\\_sweep](#) (uint32\_t num\_sweeps, uint32\_t doppler\_step, int32\_t doppler\_min)  
*Configure the doppler sweep parameters in the FPGA.*
- void [run\\_acquisition](#) ()  
*Run the acquisition process in the FPGA.*
- void [read\\_acquisition\\_results](#) (uint32\_t \*max\_index, float \*firstpeak, float \*secondpeak, uint64\_t \*initial\_sample, float \*power\_sum, uint32\_t \*doppler\_index, uint32\_t \*total\_blk\_exp)  
*Read the results of the acquisition process.*
- void [set\\_doppler\\_max](#) (uint32\_t doppler\_max)  
*Set maximum Doppler grid search.*
- void [set\\_doppler\\_step](#) (uint32\_t doppler\_step)  
*Set Doppler steps for the grid search.*
- void [reset\\_acquisition](#) ()  
*Reset the FPGA PL.*
- void [stop\\_acquisition](#) ()  
*stop the acquisition and the FPGA modules.*
- void [set\\_block\\_exp](#) (uint32\_t total\_block\_exp)  
*Set the block exponent of the FFT in the FPGA.*
- void [write\\_local\\_code](#) (void)  
*Write the PRN code in the FPGA.*
- void [configure\\_acquisition](#) (void)  
*Write the acquisition parameters into the FPGA.*
- void [open\\_device](#) ()  
*Open the device driver.*
- void [close\\_device](#) ()  
*Close the device driver.*

### 10.100.1 Detailed Description

Class that implements carrier wipe-off and correlators.

Definition at line 39 of file [fpga\\_acquisition.h](#).

### 10.100.2 Constructor & Destructor Documentation

#### 10.100.2.1 Fpga\_Acquisition()

```
Fpga_Acquisition::Fpga_Acquisition (
    std::string device_name,
    uint32_t select_queue,
    std::vector< std::pair< uint32_t, uint32_t > > & downsampling_filter_specs,
    uint32_t & max_FFT_size)
```

Constructor.

#### 10.100.2.2 ~Fpga\_Acquisition()

```
Fpga_Acquisition::~Fpga_Acquisition () [default]
```

Destructor.

### 10.100.3 Member Function Documentation

#### 10.100.3.1 close\_device()

```
void Fpga_Acquisition::close_device ()
```

Close the device driver.

**10.100.3.2 configure\_acquisition()**

```
void Fpga_Acquisition::configure_acquisition (
    void )
```

Write the acquisition parameters into the FPGA.

**10.100.3.3 init()**

```
void Fpga_Acquisition::init (
    uint32_t nsamples,
    uint32_t doppler_max,
    uint32_t d_fft_size,
    int64_t resampled_fs,
    uint32_t downsampling_filter_num,
    uint32_t excludelimit,
    uint32_t * all_fft_codes)
```

Initialize acquisition parameters.

**10.100.3.4 open\_device()**

```
void Fpga_Acquisition::open_device ()
```

Open the device driver.

**10.100.3.5 read\_acquisition\_results()**

```
void Fpga_Acquisition::read_acquisition_results (
    uint32_t * max_index,
    float * firstpeak,
    float * secondpeak,
    uint64_t * initial_sample,
    float * power_sum,
    uint32_t * doppler_index,
    uint32_t * total_blk_exp)
```

Read the results of the acquisition process.

**10.100.3.6 reset\_acquisition()**

```
void Fpga_Acquisition::reset_acquisition ()
```

Reset the FPGA PL.

**10.100.3.7 run\_acquisition()**

```
void Fpga_Acquisition::run_acquisition ()
```

Run the acquisition process in the FPGA.

**10.100.3.8 set\_block\_exp()**

```
void Fpga_Acquisition::set_block_exp (
    uint32_t total_block_exp)
```

Set the block exponent of the FFT in the FPGA.

**10.100.3.9 set\_doppler\_max()**

```
void Fpga_Acquisition::set_doppler_max (
    uint32_t doppler_max) [inline]
```

Set maximum Doppler grid search.

**Parameters**

<i>doppler_max</i>	- Maximum Doppler shift considered in the grid search [Hz].
--------------------	---



Definition at line 94 of file [fpga\\_acquisition.h](#).

### 10.100.3.10 set\_doppler\_step()

```
void Fpga_Acquisition::set_doppler_step (
    uint32_t doppler_step) [inline]
```

Set Doppler steps for the grid search.

#### Parameters

<i>doppler_step</i>	- Frequency bin of the search grid [Hz].
---------------------	--

Definition at line 103 of file [fpga\\_acquisition.h](#).

### 10.100.3.11 set\_doppler\_sweep()

```
void Fpga_Acquisition::set_doppler_sweep (
    uint32_t num_sweeps,
    uint32_t doppler_step,
    int32_t doppler_min)
```

Configure the doppler sweep parameters in the FPGA.

### 10.100.3.12 set\_local\_code()

```
bool Fpga_Acquisition::set_local_code (
    uint32_t PRN)
```

Select the code with the chosen PRN.

### 10.100.3.13 stop\_acquisition()

```
void Fpga_Acquisition::stop_acquisition ()
```

stop the acquisition and the FPGA modules.

### 10.100.3.14 write\_local\_code()

```
void Fpga_Acquisition::write_local_code (
    void )
```

Write the PRN code in the FPGA.

The documentation for this class was generated from the following file:

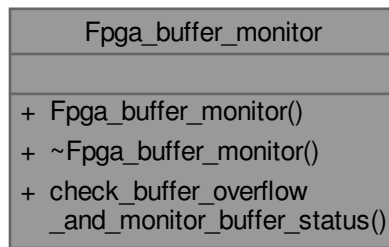
- [fpga\\_acquisition.h](#)

## 10.101 Fpga\_buffer\_monitor Class Reference

Class that checks the receiver buffer overflow flags and monitors the status of the receiver buffers.

```
#include <fpga_buffer_monitor.h>
```

Collaboration diagram for Fpga\_buffer\_monitor:



### Public Member Functions

- [Fpga\\_buffer\\_monitor](#) (uint32\_t num\_freq\_bands, bool dump, std::string dump\_filename)  
*Constructor.*
- [~Fpga\\_buffer\\_monitor](#) ()  
*Destructor.*
- void [check\\_buffer\\_overflow\\_and\\_monitor\\_buffer\\_status](#) ()  
*This function checks buffer overflow and monitors the FPGA buffer status.*

### 10.101.1 Detailed Description

Class that checks the receiver buffer overflow flags and monitors the status of the receiver buffers.  
Definition at line 42 of file [fpga\\_buffer\\_monitor.h](#).

### 10.101.2 Constructor & Destructor Documentation

#### 10.101.2.1 Fpga\_buffer\_monitor()

```
Fpga_buffer_monitor::Fpga_buffer_monitor (
    uint32_t num_freq_bands,
    bool dump,
    std::string dump_filename) [explicit]
```

Constructor.

#### 10.101.2.2 ~Fpga\_buffer\_monitor()

```
Fpga_buffer_monitor::~Fpga_buffer_monitor ()
```

Destructor.

### 10.101.3 Member Function Documentation

#### 10.101.3.1 check\_buffer\_overflow\_and\_monitor\_buffer\_status()

```
void Fpga_buffer_monitor::check_buffer_overflow_and_monitor_buffer_status ()
```

This function checks buffer overflow and monitors the FPGA buffer status.

The documentation for this class was generated from the following file:

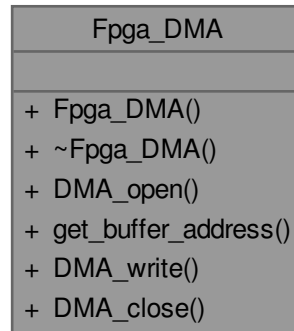
- [fpga\\_buffer\\_monitor.h](#)

## 10.102 Fpga\_DMA Class Reference

Class that controls the switch DMA in the FPGA.

```
#include <fpga_dma-proxy.h>
```

Collaboration diagram for Fpga\_DMA:



### Public Member Functions

- [Fpga\\_DMA](#) ()=default  
*Default constructor.*
- [~Fpga\\_DMA](#) ()=default  
*Default destructor.*
- int [DMA\\_open](#) (void)  
*Open the DMA device driver.*
- int8\_t \* [get\\_buffer\\_address](#) (void)  
*Obtain DMA buffer address.*
- int [DMA\\_write](#) (int nbytes) const  
*Transfer DMA data.*
- int [DMA\\_close](#) (void) const  
*Close the DMA device driver.*

### 10.102.1 Detailed Description

Class that controls the switch DMA in the FPGA.

Definition at line 26 of file [fpga\\_dma-proxy.h](#).

### 10.102.2 Constructor & Destructor Documentation

#### 10.102.2.1 Fpga\_DMA()

```
Fpga_DMA::Fpga_DMA () [default]
```

Default constructor.

#### 10.102.2.2 ~Fpga\_DMA()

```
Fpga_DMA::~Fpga_DMA () [default]
```

Default destructor.

### 10.102.3 Member Function Documentation

#### 10.102.3.1 DMA\_close()

```
int Fpga_DMA::DMA_close (
    void ) const
```

Close the DMA device driver.

#### 10.102.3.2 DMA\_open()

```
int Fpga_DMA::DMA_open (
    void )
```

Open the DMA device driver.

#### 10.102.3.3 DMA\_write()

```
int Fpga_DMA::DMA_write (
    int nbytes) const
```

Transfer DMA data.

#### 10.102.3.4 get\_buffer\_address()

```
int8_t * Fpga_DMA::get_buffer_address (
    void )
```

Obtain DMA buffer address.

The documentation for this class was generated from the following file:

- [fpga\\_dma-proxy.h](#)

## 10.103 Fpga\_dynamic\_bit\_selection Class Reference

Class that controls the switch in the FPGA, which connects the FPGA acquisition and multicorrelator modules to either the DMA or the Analog Front-End.

```
#include <fpga_dynamic_bit_selection.h>
```

Collaboration diagram for Fpga\_dynamic\_bit\_selection:

Fpga_dynamic_bit_selection
<ul style="list-style-type: none"> <li>+ Fpga_dynamic_bit_selection()</li> <li>+ ~Fpga_dynamic_bit_selection()</li> <li>+ bit_selection()</li> </ul>

### Public Member Functions

- [Fpga\\_dynamic\\_bit\\_selection](#) (bool enable\_rx1\_band, bool enable\_rx2\_band)  
*Constructor.*
- [~Fpga\\_dynamic\\_bit\\_selection](#) ()  
*Destructor.*
- void [bit\\_selection](#) (void)  
*This function configures the switch in the FPGA.*

### 10.103.1 Detailed Description

Class that controls the switch in the FPGA, which connects the FPGA acquisition and multicorrelator modules to either the DMA or the Analog Front-End.

Definition at line 40 of file [fpga\\_dynamic\\_bit\\_selection.h](#).

### 10.103.2 Constructor & Destructor Documentation

#### 10.103.2.1 Fpga\_dynamic\_bit\_selection()

```
Fpga_dynamic_bit_selection::Fpga_dynamic_bit_selection (
    bool enable_rx1_band,
    bool enable_rx2_band) [explicit]
```

Constructor.

#### 10.103.2.2 ~Fpga\_dynamic\_bit\_selection()

```
Fpga_dynamic_bit_selection::~~Fpga_dynamic_bit_selection ()
```

Destructor.

### 10.103.3 Member Function Documentation

#### 10.103.3.1 bit\_selection()

```
void Fpga_dynamic_bit_selection::bit_selection (
    void )
```

This function configures the switch in the FPGA.

The documentation for this class was generated from the following file:

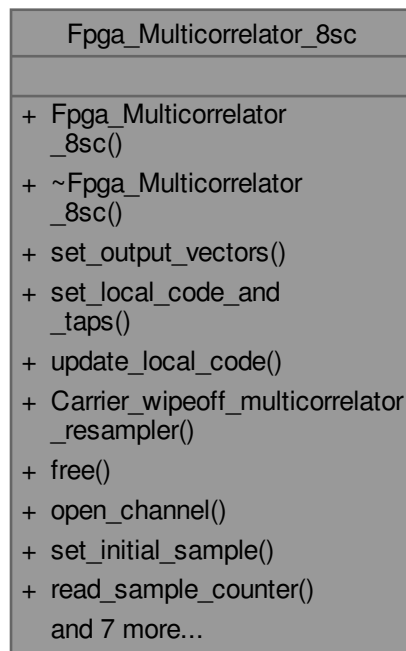
- [fpga\\_dynamic\\_bit\\_selection.h](#)

## 10.104 Fpga\_Multicorrelator\_8sc Class Reference

Class that implements carrier wipe-off and correlators.

```
#include <fpga_multicorrelator.h>
```

Collaboration diagram for Fpga\_Multicorrelator\_8sc:



## Public Member Functions

- [Fpga\\_Multicorrelator\\_8sc](#) (int32\_t n\_correlators, int32\_t \*ca\_codes, int32\_t \*data\_codes, uint32\_t code\_length\_chips, bool track\_pilot, uint32\_t code\_samples\_per\_chip)  
*Constructor.*
- [~Fpga\\_Multicorrelator\\_8sc](#) ()  
*Destructor.*
- void [set\\_output\\_vectors](#) (gr\_complex \*corr\_out, gr\_complex \*Prompt\_Data)  
*Configure pointers to the FPGA multicorrelator results.*
- void [set\\_local\\_code\\_and\\_taps](#) (float \*shifts\_chips, float \*prompt\_data\_shift, int32\_t PRN)  
*Configure the local code in the FPGA multicorrelator.*
- void [update\\_local\\_code](#) ()  
*Configure code phase and code rate parameters in the FPGA.*
- void [Carrier\\_wipeoff\\_multicorrelator\\_resampler](#) (float rem\_carrier\_phase\_in\_rad, float phase\_step\_rad, float carrier\_phase\_rate\_step\_rad, float rem\_code\_phase\_chips, float code\_phase\_step\_chips, float code\_phase\_rate\_step\_chips, int32\_t signal\_length\_samples)  
*Perform a multicorrelation.*
- bool [free](#) ()  
*Stop the correlation process in the FPGA and free code phase and code rate parameters.*
- void [open\\_channel](#) (const std::string &device\_io\_name, uint32\_t channel)  
*Open the FPGA device driver.*
- void [set\\_initial\\_sample](#) (uint64\_t samples\_offset)  
*Set the initial sample number where the tracking process begins.*
- uint64\_t [read\\_sample\\_counter](#) ()  
*Read the sample counter in the FPGA.*

- void `lock_channel` ()  
*Start the tracking process in the FPGA.*
- void `unlock_channel` ()  
*finish the tracking process in the FPGA*
- void `set_secondary_code_lengths` (uint32\_t secondary\_code\_0\_length, uint32\_t secondary\_code\_1\_length)  
*Set the secondary code length in the FPGA. This is only used when extended coherent integration is enabled in the FPGA. If tracking the pilot is enabled then secondary\_code\_0\_length is the length of the pilot secondary code and secondary\_code\_1\_length is the length of the data secondary code. If tracking the pilot is disabled then secondary\_code\_0\_length is the length of the data secondary code, and secondary\_code\_1\_length must be set to zero.*
- void `initialize_secondary_code` (uint32\_t secondary\_code, std::string \*secondary\_code\_string)  
*Initialize the secondary code in the FPGA. If tracking the pilot is enabled then the pilot secondary code is configured when secondary\_code = 0 and the data secondary code is configured when secondary\_code = 1. If tracking the pilot is disabled then the data secondary code is configured when secondary\_code = 0.*
- void `update_prn_code_length` (uint32\_t first\_prn\_length, uint32\_t next\_prn\_length)  
*Set the PRN length in the FPGA in number of samples. This function is only used then extended coherent integration is enabled in the FPGA. The FPGA allows for the configuration of two PRN lengths. When the length of the extended coherent integration is bigger than the length of the PRN code, the FPGA uses the first\_length\_secondary\_code as the length of the PRN code immediately following the beginning of the extended coherent integration, and the next\_length\_secondary\_code as the length of the remaining PRN codes. The purpose of this is to have the option to allow the FPGA to compensate for a possible deviation between the nominal value of the PRN code length and the measured PRN code length in the PRN immediately following the start of the coherent integration only. If this option is not used then write the same value to first\_length\_secondary\_code and next\_length\_secondary\_code.*
- void `enable_secondary_codes` ()  
*Enable the use of secondary codes in the FPGA.*
- void `disable_secondary_codes` ()  
*Disable the use of secondary codes in the FPGA.*

### 10.104.1 Detailed Description

Class that implements carrier wipe-off and correlators.

Definition at line 40 of file [fpga\\_multicorrelator.h](#).

### 10.104.2 Constructor & Destructor Documentation

#### 10.104.2.1 Fpga\_Multicorrelator\_8sc()

```
Fpga_Multicorrelator_8sc::Fpga_Multicorrelator_8sc (
    int32_t n_correlators,
    int32_t * ca_codes,
    int32_t * data_codes,
    uint32_t code_length_chips,
    bool track_pilot,
    uint32_t code_samples_per_chip)
```

Constructor.

#### 10.104.2.2 ~Fpga\_Multicorrelator\_8sc()

```
Fpga_Multicorrelator_8sc::~Fpga_Multicorrelator_8sc ()
```

Destructor.

### 10.104.3 Member Function Documentation

#### 10.104.3.1 Carrier\_wipeoff\_multicorrelator\_resampler()

```
void Fpga_Multicorrelator_8sc::Carrier_wipeoff_multicorrelator_resampler (
    float rem_carrier_phase_in_rad,
    float phase_step_rad,
    float carrier_phase_rate_step_rad,
```

```
float rem_code_phase_chips,
float code_phase_step_chips,
float code_phase_rate_step_chips,
int32_t signal_length_samples)
```

Perform a multicorrelation.

#### 10.104.3.2 disable\_secondary\_codes()

```
void Fpga_Multicorrelator_8sc::disable_secondary_codes ()
```

Disable the use of secondary codes in the FPGA.

#### 10.104.3.3 enable\_secondary\_codes()

```
void Fpga_Multicorrelator_8sc::enable_secondary_codes ()
```

Enable the use of secondary codes in the FPGA.

#### 10.104.3.4 free()

```
bool Fpga_Multicorrelator_8sc::free ()
```

Stop the correlation process in the FPGA and free code phase and code rate parameters.

#### 10.104.3.5 initialize\_secondary\_code()

```
void Fpga_Multicorrelator_8sc::initialize_secondary_code (
    uint32_t secondary_code,
    std::string * secondary_code_string)
```

Initialize the secondary code in the FPGA. If tracking the pilot is enabled then the pilot secondary code is configured when `secondary_code = 0` and the data secondary code is configured when `secondary_code = 1`. If tracking the pilot is disabled then the data secondary code is configured when `secondary_code = 0`.

#### 10.104.3.6 lock\_channel()

```
void Fpga_Multicorrelator_8sc::lock_channel ()
```

Start the tracking process in the FPGA.

#### 10.104.3.7 open\_channel()

```
void Fpga_Multicorrelator_8sc::open_channel (
    const std::string & device_io_name,
    uint32_t channel)
```

Open the FPGA device driver.

#### 10.104.3.8 read\_sample\_counter()

```
uint64_t Fpga_Multicorrelator_8sc::read_sample_counter ()
```

Read the sample counter in the FPGA.

#### 10.104.3.9 set\_initial\_sample()

```
void Fpga_Multicorrelator_8sc::set_initial_sample (
    uint64_t samples_offset)
```

Set the initial sample number where the tracking process begins.

#### 10.104.3.10 set\_local\_code\_and\_taps()

```
void Fpga_Multicorrelator_8sc::set_local_code_and_taps (
    float * shifts_chips,
    float * prompt_data_shift,
    int32_t PRN)
```

Configure the local code in the FPGA multicorrelator.



### 10.104.3.11 set\_output\_vectors()

```
void Fpga_Multicorrelator_8sc::set_output_vectors (
    gr_complex * corr_out,
    gr_complex * Prompt_Data)
```

Configure pointers to the FPGA multicorrelator results.

### 10.104.3.12 set\_secondary\_code\_lengths()

```
void Fpga_Multicorrelator_8sc::set_secondary_code_lengths (
    uint32_t secondary_code_0_length,
    uint32_t secondary_code_1_length)
```

Set the secondary code length in the FPGA. This is only used when extended coherent integration is enabled in the FPGA. If tracking the pilot is enabled then secondary\_code\_0\_length is the length of the pilot secondary code and secondary\_code\_1\_length is the length of the data secondary code. If tracking the pilot is disabled then secondary\_code\_0\_length is the length of the data secondary code, and secondary\_code\_1\_length must be set to zero.

### 10.104.3.13 unlock\_channel()

```
void Fpga_Multicorrelator_8sc::unlock_channel ()
```

finish the tracking process in the FPGA

### 10.104.3.14 update\_local\_code()

```
void Fpga_Multicorrelator_8sc::update_local_code ()
```

Configure code phase and code rate parameters in the FPGA.

### 10.104.3.15 update\_prn\_code\_length()

```
void Fpga_Multicorrelator_8sc::update_prn_code_length (
    uint32_t first_prn_length,
    uint32_t next_prn_length)
```

Set the PRN length in the FPGA in number of samples. This function is only used then extended coherent integration is enabled in the FPGA. The FPGA allows for the configuration of two PRN lengths. When the length of the extended coherent integration is bigger than the length of the PRN code, the FPGA uses the first\_length\_secondary\_code as the length of the PRN code immediately following the beginning of the extended coherent integration, and the next\_length\_secondary\_code as the length of the remaining PRN codes. The purpose of this is to have the option to allow the FPGA to compensate for a possible deviation between the nominal value of the PRN code length and the measured PRN code length in the PRN immediately following the start of the coherent integration only. If this option is not used then write the same value to first\_length\_secondary\_code and next\_length\_secondary\_code.

The documentation for this class was generated from the following file:

- [fpga\\_multicorrelator.h](#)

## 10.105 Fpga\_spidev Class Reference

Collaboration diagram for Fpga\_spidev:

Fpga_spidev
<ul style="list-style-type: none"> <li>+ Fpga_spidev()</li> <li>+ ~Fpga_spidev()</li> <li>+ write_reg32()</li> <li>+ read_reg32()</li> <li>+ SPI_open()</li> <li>+ SPI_close()</li> </ul>

### Public Member Functions

- [Fpga\\_spidev](#) ()=default  
*Default constructor.*
- [~Fpga\\_spidev](#) ()=default  
*Default destructor.*
- int [write\\_reg32](#) (char addr, uint32\_t data)  
*write a register through the SPI.*
- int [read\\_reg32](#) (uint8\_t addr, uint32\_t \*copy\_to)  
*read a register through the SPI.*
- int [SPI\\_open](#) (void)  
*Open the SPI device driver.*
- int [SPI\\_close](#) (void) const  
*Close the SPI device driver.*

### 10.105.1 Detailed Description

Definition at line 22 of file [fpga\\_spidev.h](#).

### 10.105.2 Constructor & Destructor Documentation

#### 10.105.2.1 Fpga\_spidev()

`Fpga_spidev::Fpga_spidev () [default]`  
Default constructor.

#### 10.105.2.2 ~Fpga\_spidev()

`Fpga_spidev::~~Fpga_spidev () [default]`  
Default destructor.

### 10.105.3 Member Function Documentation

#### 10.105.3.1 read\_reg32()

```
int Fpga_spidev::read_reg32 (
    uint8_t addr,
    uint32_t * copy_to)
```

read a register through the SPI.

#### 10.105.3.2 SPI\_close()

```
int Fpga_spidev::SPI_close (
    void ) const
```

Close the SPI device driver.

#### 10.105.3.3 SPI\_open()

```
int Fpga_spidev::SPI_open (
    void )
```

Open the SPI device driver.

#### 10.105.3.4 write\_reg32()

```
int Fpga_spidev::write_reg32 (
    char addr,
    uint32_t data)
```

write a register through the SPI.

The documentation for this class was generated from the following file:

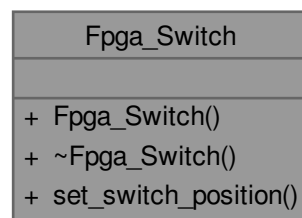
- [fpga\\_spidev.h](#)

## 10.106 Fpga\_Switch Class Reference

Class that controls the switch in the FPGA, which connects the FPGA acquisition and multicorrelator modules to either the DMA or the Analog Front-End.

```
#include <fpga_switch.h>
```

Collaboration diagram for Fpga\_Switch:



### Public Member Functions

- [Fpga\\_Switch](#) (void)  
*Constructor.*
- [~Fpga\\_Switch](#) ()  
*Destructor.*

- void [set\\_switch\\_position](#) (int32\_t switch\_position)

*This function configures the switch in the eFPGA.*

### 10.106.1 Detailed Description

Class that controls the switch in the FPGA, which connects the FPGA acquisition and multicorrelator modules to either the DMA or the Analog Front-End.

Definition at line 39 of file [fpga\\_switch.h](#).

### 10.106.2 Constructor & Destructor Documentation

#### 10.106.2.1 Fpga\_Switch()

```
Fpga_Switch::Fpga_Switch (  
    void )
```

Constructor.

#### 10.106.2.2 ~Fpga\_Switch()

```
Fpga_Switch::~~Fpga_Switch ()
```

Destructor.

### 10.106.3 Member Function Documentation

#### 10.106.3.1 set\_switch\_position()

```
void Fpga_Switch::set_switch_position (  
    int32_t switch_position)
```

This function configures the switch in the eFPGA.

The documentation for this class was generated from the following file:

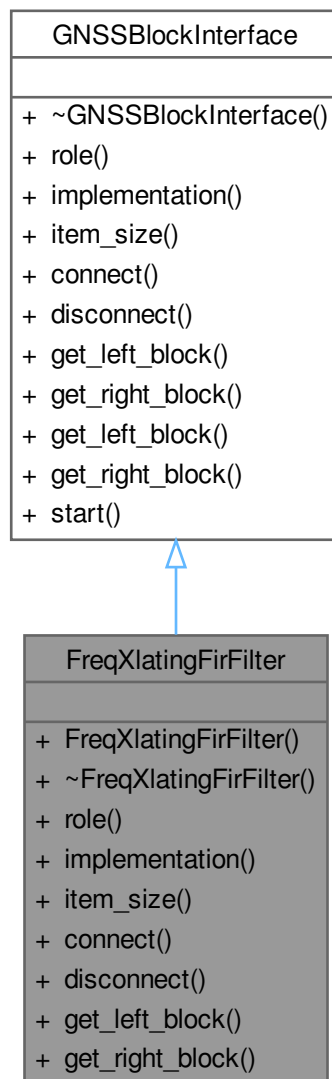
- [fpga\\_switch.h](#)

## 10.107 FreqXlatingFirFilter Class Reference

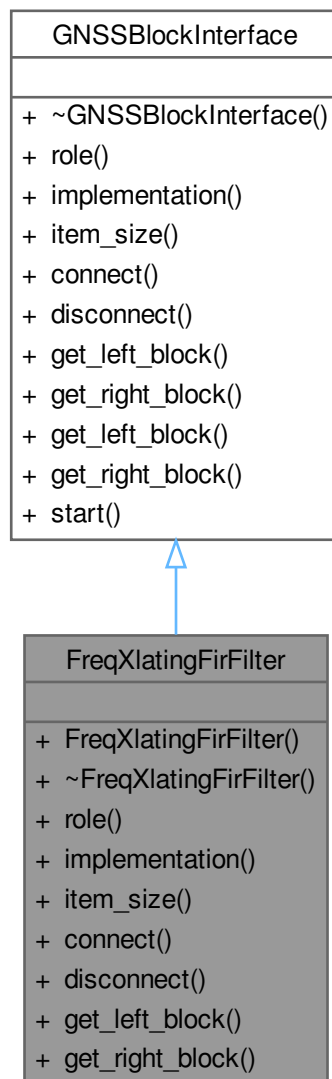
This class adapts a gnuradio `gr_freq_xlating_fir_filter` designed with `pm_remez`.

```
#include <freq_xlating_fir_filter.h>
```

Inheritance diagram for FreqXlatingFirFilter:



Collaboration diagram for FreqXlatingFirFilter:



### Public Member Functions

- **FreqXlatingFirFilter** (const [ConfigurationInterface](#) \*configuration, std::string role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "Freq\_Xlating\_Fir\_Filter".
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override

## Public Member Functions inherited from GNSSBlockInterface

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.107.1 Detailed Description

This class adapts a gnuradio gr\_freq\_xlating\_fir\_filter designed with pm\_remez.

Construct a FIR filter with the given taps and a composite frequency translation that shifts intermediate\_freq\_ down to zero Hz. The frequency translation logically comes before the filtering operation.

See Parks-McClellan FIR filter design, [https://en.wikipedia.org/wiki/Parks-McClellan\\_filter\\_design\\_algorithm](https://en.wikipedia.org/wiki/Parks-McClellan_filter_design_algorithm) Calculates the optimal (in the Chebyshev/minimax sense) FIR filter impulse response given a set of band edges, the desired response on those bands, and the weight given to the error in those bands.

Definition at line 58 of file [freq\\_xlating\\_fir\\_filter.h](#).

### 10.107.2 Member Function Documentation

#### 10.107.2.1 connect()

```
void FreqXlatingFirFilter::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.107.2.2 disconnect()

```
void FreqXlatingFirFilter::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.107.2.3 get\_left\_block()

```
gr::basic_block_sptr FreqXlatingFirFilter::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.107.2.4 get\_right\_block()

```
gr::basic_block_sptr FreqXlatingFirFilter::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.107.2.5 implementation()

```
std::string FreqXlatingFirFilter::implementation () [inline], [override], [virtual]
```

Returns "Freq\_Xlating\_Fir\_Filter".

Implements [GNSSBlockInterface](#).

Definition at line 73 of file [freq\\_xlating\\_fir\\_filter.h](#).

#### 10.107.2.6 item\_size()

```
size_t FreqXlatingFirFilter::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 78 of file [freq\\_xlating\\_fir\\_filter.h](#).

#### 10.107.2.7 role()

```
std::string FreqXlatingFirFilter::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

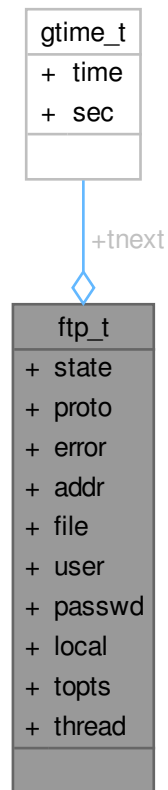
Definition at line 67 of file [freq\\_xlating\\_fir\\_filter.h](#).

The documentation for this class was generated from the following file:

- [freq\\_xlating\\_fir\\_filter.h](#)

## 10.108 ftp\_t Struct Reference

Collaboration diagram for ftp\_t:



### Public Attributes

- int `state`
- int `proto`
- int `error`
- char `addr` [1024]
- char `file` [1024]
- char `user` [256]
- char `passwd` [256]
- char `local` [1024]
- int `topts` [4]
- `gtime_t` `tnext`
- `pthread_t` `thread`

### 10.108.1 Detailed Description

Definition at line 1184 of file [rtklib.h](#).



## 10.108.2 Member Data Documentation

### 10.108.2.1 addr

```
char ftp_t::addr[1024]
```

Definition at line 1190 of file [rtklib.h](#).

### 10.108.2.2 error

```
int ftp_t::error
```

Definition at line 1188 of file [rtklib.h](#).

### 10.108.2.3 file

```
char ftp_t::file[1024]
```

Definition at line 1191 of file [rtklib.h](#).

### 10.108.2.4 local

```
char ftp_t::local[1024]
```

Definition at line 1194 of file [rtklib.h](#).

### 10.108.2.5 passwd

```
char ftp_t::passwd[256]
```

Definition at line 1193 of file [rtklib.h](#).

### 10.108.2.6 proto

```
int ftp_t::proto
```

Definition at line 1187 of file [rtklib.h](#).

### 10.108.2.7 state

```
int ftp_t::state
```

Definition at line 1186 of file [rtklib.h](#).

### 10.108.2.8 thread

```
pthread_t ftp_t::thread
```

Definition at line 1197 of file [rtklib.h](#).

### 10.108.2.9 tnext

```
gtime_t ftp_t::tnext
```

Definition at line 1196 of file [rtklib.h](#).

### 10.108.2.10 topts

```
int ftp_t::topts[4]
```

Definition at line 1195 of file [rtklib.h](#).

### 10.108.2.11 user

```
char ftp_t::user[256]
```

Definition at line 1192 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

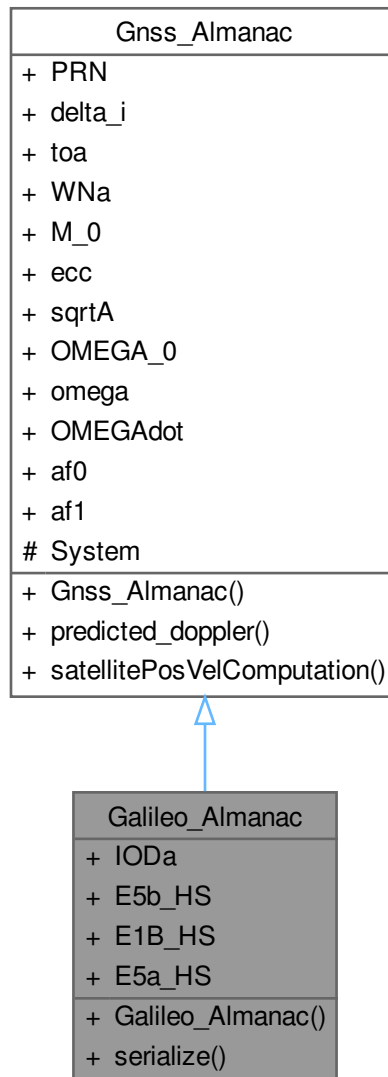
- [rtklib.h](#)

## 10.109 Galileo\_Almanac Class Reference

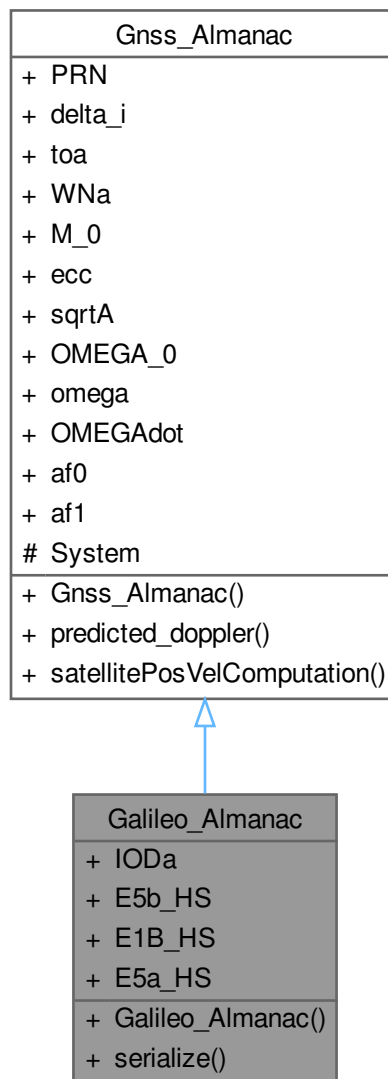
This class is a storage for the Galileo SV ALMANAC data.

```
#include <galileo_almanac.h>
```

Inheritance diagram for Galileo\_Almanac:



Collaboration diagram for Galileo\_Almanac:



### Public Member Functions

- [Galileo\\_Almanac](#) ()
- template<class Archive>  
void [serialize](#) (Archive &ar, const unsigned int version)

### Public Member Functions inherited from [Gnss\\_Almanac](#)

- [Gnss\\_Almanac](#) ()=default
- double [predicted\\_doppler](#) (double rx\_time\_s, double lat, double lon, double h, double ve, double vn, double vu, int band) const  
*Computes prediction of the Doppler shift for a given time and receiver's position and velocity.*
- void [satellitePosVelComputation](#) (double transmitTime, std::array< double, 7 > &pos\_vel\_dtr) const  
*Computes satellite Position and Velocity, in ECEF, for a given time (expressed in seconds of week)*

**Public Attributes**

- int32\_t [IODa](#) {}
- int32\_t [E5b\\_HS](#) {}
- int32\_t [E1B\\_HS](#) {}
- int32\_t [E5a\\_HS](#) {}

**Public Attributes inherited from [Gnss\\_Almanac](#)**

- uint32\_t [PRN](#) {}  
*SV PRN NUMBER.*
- double [delta\\_i](#) {}  
*Inclination Angle at Reference Time (relative to  $i_0 = 0.30$  semi-circles)*
- int32\_t [toa](#) {}  
*Almanac data reference time of week [s].*
- int32\_t [WNa](#) {}  
*Almanac week number.*
- double [M\\_0](#) {}  
*Mean Anomaly at Reference Time [semi-circles].*
- double [ecc](#) {}  
*Eccentricity [dimensionless].*
- double [sqrtA](#) {}  
*Square Root of the Semi-Major Axis [sqrt(m)].*
- double [OMEGA\\_0](#) {}  
*Longitude of Ascending Node of Orbit Plane at Weekly Epoch [semi-circles].*
- double [omega](#) {}  
*Argument of Perigee [semi-circles].*
- double [OMEGAdot](#) {}  
*Rate of Right Ascension [semi-circles/s].*
- double [af0](#) {}  
*Coefficient 0 of code phase offset model [s].*
- double [af1](#) {}  
*Coefficient 1 of code phase offset model [s/s].*

**Additional Inherited Members****Protected Attributes inherited from [Gnss\\_Almanac](#)**

- char [System](#) {}  
*Character ID of the GNSS system. 'G': GPS. 'E': Galileo. 'B': BeiDou.*

**10.109.1 Detailed Description**

This class is a storage for the Galileo SV ALMANAC data.  
Definition at line 33 of file [galileo\\_almanac.h](#).

**10.109.2 Constructor & Destructor Documentation****10.109.2.1 [Galileo\\_Almanac](#)()**

```
Galileo_Almanac::Galileo_Almanac () [inline]
```

Default constructor

Definition at line 39 of file [galileo\\_almanac.h](#).

References [Gnss\\_Almanac::System](#).

### 10.109.3 Member Function Documentation

#### 10.109.3.1 serialize()

```
template<class Archive>
void Galileo_Almanac::serialize (
    Archive & ar,
    const unsigned int version) [inline]
```

Definition at line 51 of file [galileo\\_almanac.h](#).

### 10.109.4 Member Data Documentation

#### 10.109.4.1 E1B\_HS

```
int32_t Galileo_Almanac::E1B_HS {}
```

Definition at line 46 of file [galileo\\_almanac.h](#).

#### 10.109.4.2 E5a\_HS

```
int32_t Galileo_Almanac::E5a_HS {}
```

Definition at line 47 of file [galileo\\_almanac.h](#).

#### 10.109.4.3 E5b\_HS

```
int32_t Galileo_Almanac::E5b_HS {}
```

Definition at line 45 of file [galileo\\_almanac.h](#).

#### 10.109.4.4 IODa

```
int32_t Galileo_Almanac::IODa {}
```

Definition at line 44 of file [galileo\\_almanac.h](#).

The documentation for this class was generated from the following file:

- [galileo\\_almanac.h](#)

## 10.110 Galileo\_Almanac\_Helper Class Reference

This class is a storage for the GALILEO ALMANAC data as described in GALILEO ICD.

```
#include <galileo_almanac_helper.h>
```

Collaboration diagram for Galileo\_Almanac\_Helper:

Galileo_Almanac_Helper
<ul style="list-style-type: none"> <li>+ IOD_a_7</li> <li>+ WN_a_7</li> <li>+ t0a_7</li> <li>+ SVID1_7</li> <li>+ DELTA_A_7</li> <li>+ e_7</li> <li>+ omega_7</li> <li>+ delta_i_7</li> <li>+ Omega0_7</li> <li>+ Omega_dot_7</li> <li>and 37 more...</li> </ul>
<ul style="list-style-type: none"> <li>+ Galileo_Almanac_Helper()</li> <li>+ get_almanac()</li> </ul>

### Public Member Functions

- [Galileo\\_Almanac\\_Helper](#) ()=default  
*Default constructor.*
- [Galileo\\_Almanac](#) **get\_almanac** (int i) const

### Public Attributes

- int32\_t [IOD\\_a\\_7](#) {}
- int32\_t [WN\\_a\\_7](#) {}
- int32\_t [t0a\\_7](#) {}
- int32\_t [SVID1\\_7](#) {}
- double [DELTA\\_A\\_7](#) {}
- double [e\\_7](#) {}
- double [omega\\_7](#) {}
- double [delta\\_i\\_7](#) {}
- double [Omega0\\_7](#) {}
- double [Omega\\_dot\\_7](#) {}
- double [M0\\_7](#) {}
- int32\_t [IOD\\_a\\_8](#) {}
- double [af0\\_8](#) {}
- double [af1\\_8](#) {}
- int32\_t [E5b\\_HS\\_8](#) {}
- int32\_t [E1B\\_HS\\_8](#) {}
- int32\_t [E5a\\_HS\\_8](#) {}
- int32\_t [SVID2\\_8](#) {}
- double [DELTA\\_A\\_8](#) {}
- double [e\\_8](#) {}

- double [omega\\_8](#) {}
- double [delta\\_i\\_8](#) {}
- double [Omega0\\_8](#) {}
- double [Omega\\_dot\\_8](#) {}
- int32\_t [IOD\\_a\\_9](#) {}
- int32\_t [WN\\_a\\_9](#) {}
- int32\_t [t0a\\_9](#) {}
- double [M0\\_9](#) {}
- double [af0\\_9](#) {}
- double [af1\\_9](#) {}
- int32\_t [E5b\\_HS\\_9](#) {}
- int32\_t [E1B\\_HS\\_9](#) {}
- int32\_t [E5a\\_HS\\_9](#) {}
- int32\_t [SVID3\\_9](#) {}
- double [DELTA\\_A\\_9](#) {}
- double [e\\_9](#) {}
- double [omega\\_9](#) {}
- double [delta\\_i\\_9](#) {}
- int32\_t [IOD\\_a\\_10](#) {}
- double [Omega0\\_10](#) {}
- double [Omega\\_dot\\_10](#) {}
- double [M0\\_10](#) {}
- double [af0\\_10](#) {}
- double [af1\\_10](#) {}
- int32\_t [E5b\\_HS\\_10](#) {}
- int32\_t [E1B\\_HS\\_10](#) {}
- int32\_t [E5a\\_HS\\_10](#) {}

### 10.110.1 Detailed Description

This class is a storage for the GALILEO ALMANAC data as described in GALILEO ICD.

See [https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo\\_OS\\_SIS\\_ICD\\_v2.0.pdf](https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.0.pdf) paragraph 5.1.10

Definition at line 34 of file [galileo\\_almanac\\_helper.h](#).

### 10.110.2 Constructor & Destructor Documentation

#### 10.110.2.1 Galileo\_Almanac\_Helper()

`Galileo_Almanac_Helper::Galileo_Almanac_Helper () [default]`

Default constructor.

### 10.110.3 Member Data Documentation

#### 10.110.3.1 af0\_10

`double Galileo_Almanac_Helper::af0_10 {}`

Definition at line 90 of file [galileo\\_almanac\\_helper.h](#).

#### 10.110.3.2 af0\_8

`double Galileo_Almanac_Helper::af0_8 {}`

Definition at line 56 of file [galileo\\_almanac\\_helper.h](#).

#### 10.110.3.3 af0\_9

`double Galileo_Almanac_Helper::af0_9 {}`

Definition at line 74 of file [galileo\\_almanac\\_helper.h](#).

#### 10.110.3.4 af1\_10

double Galileo\_Almanac\_Helper::af1\_10 {}  
Definition at line 91 of file [galileo\\_almanac\\_helper.h](#).

#### 10.110.3.5 af1\_8

double Galileo\_Almanac\_Helper::af1\_8 {}  
Definition at line 57 of file [galileo\\_almanac\\_helper.h](#).

#### 10.110.3.6 af1\_9

double Galileo\_Almanac\_Helper::af1\_9 {}  
Definition at line 75 of file [galileo\\_almanac\\_helper.h](#).

#### 10.110.3.7 DELTA\_A\_7

double Galileo\_Almanac\_Helper::DELTA\_A\_7 {}  
Definition at line 46 of file [galileo\\_almanac\\_helper.h](#).

#### 10.110.3.8 DELTA\_A\_8

double Galileo\_Almanac\_Helper::DELTA\_A\_8 {}  
Definition at line 62 of file [galileo\\_almanac\\_helper.h](#).

#### 10.110.3.9 DELTA\_A\_9

double Galileo\_Almanac\_Helper::DELTA\_A\_9 {}  
Definition at line 80 of file [galileo\\_almanac\\_helper.h](#).

#### 10.110.3.10 delta\_i\_7

double Galileo\_Almanac\_Helper::delta\_i\_7 {}  
Definition at line 49 of file [galileo\\_almanac\\_helper.h](#).

#### 10.110.3.11 delta\_i\_8

double Galileo\_Almanac\_Helper::delta\_i\_8 {}  
Definition at line 65 of file [galileo\\_almanac\\_helper.h](#).

#### 10.110.3.12 delta\_i\_9

double Galileo\_Almanac\_Helper::delta\_i\_9 {}  
Definition at line 83 of file [galileo\\_almanac\\_helper.h](#).

#### 10.110.3.13 E1B\_HS\_10

int32\_t Galileo\_Almanac\_Helper::E1B\_HS\_10 {}  
Definition at line 93 of file [galileo\\_almanac\\_helper.h](#).

#### 10.110.3.14 E1B\_HS\_8

int32\_t Galileo\_Almanac\_Helper::E1B\_HS\_8 {}  
Definition at line 59 of file [galileo\\_almanac\\_helper.h](#).

#### 10.110.3.15 E1B\_HS\_9

int32\_t Galileo\_Almanac\_Helper::E1B\_HS\_9 {}  
Definition at line 77 of file [galileo\\_almanac\\_helper.h](#).



**10.110.3.16 E5a\_HS\_10**

`int32_t Galileo_Almanac_Helper::E5a_HS_10 {}`  
Definition at line 94 of file [galileo\\_almanac\\_helper.h](#).

**10.110.3.17 E5a\_HS\_8**

`int32_t Galileo_Almanac_Helper::E5a_HS_8 {}`  
Definition at line 60 of file [galileo\\_almanac\\_helper.h](#).

**10.110.3.18 E5a\_HS\_9**

`int32_t Galileo_Almanac_Helper::E5a_HS_9 {}`  
Definition at line 78 of file [galileo\\_almanac\\_helper.h](#).

**10.110.3.19 E5b\_HS\_10**

`int32_t Galileo_Almanac_Helper::E5b_HS_10 {}`  
Definition at line 92 of file [galileo\\_almanac\\_helper.h](#).

**10.110.3.20 E5b\_HS\_8**

`int32_t Galileo_Almanac_Helper::E5b_HS_8 {}`  
Definition at line 58 of file [galileo\\_almanac\\_helper.h](#).

**10.110.3.21 E5b\_HS\_9**

`int32_t Galileo_Almanac_Helper::E5b_HS_9 {}`  
Definition at line 76 of file [galileo\\_almanac\\_helper.h](#).

**10.110.3.22 e\_7**

`double Galileo_Almanac_Helper::e_7 {}`  
Definition at line 47 of file [galileo\\_almanac\\_helper.h](#).

**10.110.3.23 e\_8**

`double Galileo_Almanac_Helper::e_8 {}`  
Definition at line 63 of file [galileo\\_almanac\\_helper.h](#).

**10.110.3.24 e\_9**

`double Galileo_Almanac_Helper::e_9 {}`  
Definition at line 81 of file [galileo\\_almanac\\_helper.h](#).

**10.110.3.25 IOD\_a\_10**

`int32_t Galileo_Almanac_Helper::IOD_a_10 {}`  
Definition at line 86 of file [galileo\\_almanac\\_helper.h](#).

**10.110.3.26 IOD\_a\_7**

`int32_t Galileo_Almanac_Helper::IOD_a_7 {}`  
Definition at line 42 of file [galileo\\_almanac\\_helper.h](#).

**10.110.3.27 IOD\_a\_8**

`int32_t Galileo_Almanac_Helper::IOD_a_8 {}`  
Definition at line 55 of file [galileo\\_almanac\\_helper.h](#).

**10.110.3.28 IOD\_a\_9**

```
int32_t Galileo_Almanac_Helper::IOD_a_9 {}
```

Definition at line 70 of file [galileo\\_almanac\\_helper.h](#).

**10.110.3.29 M0\_10**

```
double Galileo_Almanac_Helper::M0_10 {}
```

Definition at line 89 of file [galileo\\_almanac\\_helper.h](#).

**10.110.3.30 M0\_7**

```
double Galileo_Almanac_Helper::M0_7 {}
```

Definition at line 52 of file [galileo\\_almanac\\_helper.h](#).

**10.110.3.31 M0\_9**

```
double Galileo_Almanac_Helper::M0_9 {}
```

Definition at line 73 of file [galileo\\_almanac\\_helper.h](#).

**10.110.3.32 Omega0\_10**

```
double Galileo_Almanac_Helper::Omega0_10 {}
```

Definition at line 87 of file [galileo\\_almanac\\_helper.h](#).

**10.110.3.33 Omega0\_7**

```
double Galileo_Almanac_Helper::Omega0_7 {}
```

Definition at line 50 of file [galileo\\_almanac\\_helper.h](#).

**10.110.3.34 Omega0\_8**

```
double Galileo_Almanac_Helper::Omega0_8 {}
```

Definition at line 66 of file [galileo\\_almanac\\_helper.h](#).

**10.110.3.35 omega\_7**

```
double Galileo_Almanac_Helper::omega_7 {}
```

Definition at line 48 of file [galileo\\_almanac\\_helper.h](#).

**10.110.3.36 omega\_8**

```
double Galileo_Almanac_Helper::omega_8 {}
```

Definition at line 64 of file [galileo\\_almanac\\_helper.h](#).

**10.110.3.37 omega\_9**

```
double Galileo_Almanac_Helper::omega_9 {}
```

Definition at line 82 of file [galileo\\_almanac\\_helper.h](#).

**10.110.3.38 Omega\_dot\_10**

```
double Galileo_Almanac_Helper::Omega_dot_10 {}
```

Definition at line 88 of file [galileo\\_almanac\\_helper.h](#).

**10.110.3.39 Omega\_dot\_7**

```
double Galileo_Almanac_Helper::Omega_dot_7 {}
```

Definition at line 51 of file [galileo\\_almanac\\_helper.h](#).

#### 10.110.3.40 Omega\_dot\_8

`double Galileo_Almanac_Helper::Omega_dot_8 {}`  
Definition at line 67 of file [galileo\\_almanac\\_helper.h](#).

#### 10.110.3.41 SVID1\_7

`int32_t Galileo_Almanac_Helper::SVID1_7 {}`  
Definition at line 45 of file [galileo\\_almanac\\_helper.h](#).

#### 10.110.3.42 SVID2\_8

`int32_t Galileo_Almanac_Helper::SVID2_8 {}`  
Definition at line 61 of file [galileo\\_almanac\\_helper.h](#).

#### 10.110.3.43 SVID3\_9

`int32_t Galileo_Almanac_Helper::SVID3_9 {}`  
Definition at line 79 of file [galileo\\_almanac\\_helper.h](#).

#### 10.110.3.44 t0a\_7

`int32_t Galileo_Almanac_Helper::t0a_7 {}`  
Definition at line 44 of file [galileo\\_almanac\\_helper.h](#).

#### 10.110.3.45 t0a\_9

`int32_t Galileo_Almanac_Helper::t0a_9 {}`  
Definition at line 72 of file [galileo\\_almanac\\_helper.h](#).

#### 10.110.3.46 WN\_a\_7

`int32_t Galileo_Almanac_Helper::WN_a_7 {}`  
Definition at line 43 of file [galileo\\_almanac\\_helper.h](#).

#### 10.110.3.47 WN\_a\_9

`int32_t Galileo_Almanac_Helper::WN_a_9 {}`  
Definition at line 71 of file [galileo\\_almanac\\_helper.h](#).

The documentation for this class was generated from the following file:

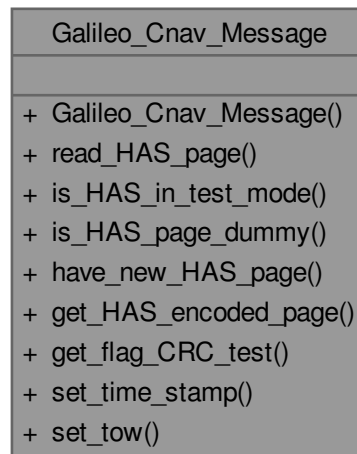
- [galileo\\_almanac\\_helper.h](#)

## 10.111 Galileo\_Cnav\_Message Class Reference

This class handles the Galileo CNAV Data message, as described in the Galileo High Accuracy Service Signal-In-Space Interface Control Document (HAS SIS ICD) Issue 1.0, May 2022.

```
#include <galileo_cnav_message.h>
```

Collaboration diagram for Galileo\_Cnav\_Message:



## Public Member Functions

- void **read\_HAS\_page** (const std::string &page\_string)
- bool [is\\_HAS\\_in\\_test\\_mode](#) () const
- bool [is\\_HAS\\_page\\_dummy](#) () const
- bool [have\\_new\\_HAS\\_page](#) () const
- [Galileo\\_HAS\\_page](#) [get\\_HAS\\_encoded\\_page](#) () const
- bool [get\\_flag\\_CRC\\_test](#) () const
- void [set\\_time\\_stamp](#) (uint64\_t time\_stamp)
- void [set\\_tow](#) (uint32\_t tow)

### 10.111.1 Detailed Description

This class handles the Galileo CNAV Data message, as described in the Galileo High Accuracy Service Signal-In-Space Interface Control Document (HAS SIS ICD) Issue 1.0, May 2022.

Definition at line 40 of file [galileo\\_cnav\\_message.h](#).

### 10.111.2 Member Function Documentation

#### 10.111.2.1 [get\\_flag\\_CRC\\_test\(\)](#)

```
bool Galileo_Cnav_Message::get_flag_CRC_test () const [inline]
```

Definition at line 67 of file [galileo\\_cnav\\_message.h](#).

#### 10.111.2.2 [get\\_HAS\\_encoded\\_page\(\)](#)

```
Galileo\_HAS\_page Galileo_Cnav_Message::get_HAS_encoded_page () const [inline]
```

Definition at line 62 of file [galileo\\_cnav\\_message.h](#).

#### 10.111.2.3 [have\\_new\\_HAS\\_page\(\)](#)

```
bool Galileo_Cnav_Message::have_new_HAS_page () const [inline]
```

Definition at line 57 of file [galileo\\_cnav\\_message.h](#).

**10.111.2.4 is\_HAS\_in\_test\_mode()**

bool Galileo\_Cnav\_Message::is\_HAS\_in\_test\_mode () const [inline]  
 Definition at line 47 of file [galileo\\_cnav\\_message.h](#).

**10.111.2.5 is\_HAS\_page\_dummy()**

bool Galileo\_Cnav\_Message::is\_HAS\_page\_dummy () const [inline]  
 Definition at line 52 of file [galileo\\_cnav\\_message.h](#).

**10.111.2.6 set\_time\_stamp()**

void Galileo\_Cnav\_Message::set\_time\_stamp (   
     uint64\_t time\_stamp) [inline]  
 Definition at line 72 of file [galileo\\_cnav\\_message.h](#).

**10.111.2.7 set\_tow()**

void Galileo\_Cnav\_Message::set\_tow (   
     uint32\_t tow) [inline]  
 Definition at line 77 of file [galileo\\_cnav\\_message.h](#).

The documentation for this class was generated from the following file:

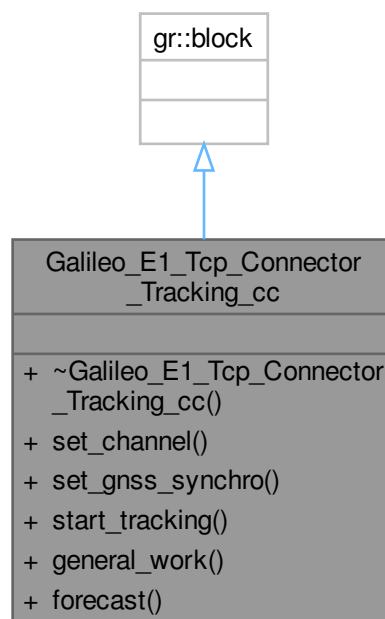
- [galileo\\_cnav\\_message.h](#)

**10.112 Galileo\_E1\_Tcp\_Connector\_Tracking\_cc Class Reference**

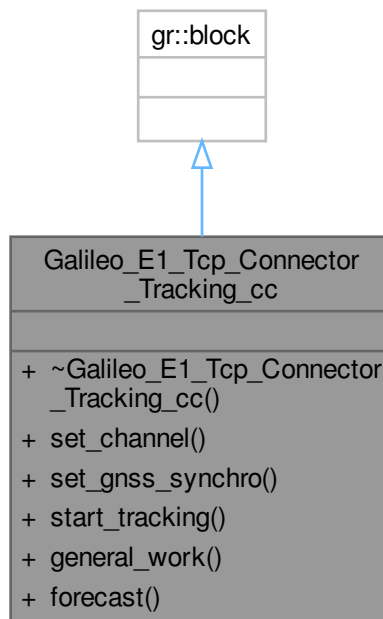
This class implements a code DLL + carrier PLL VEML (Very Early Minus Late) tracking block for Galileo E1 signals.

#include <galileo\_e1\_tcp\_connector\_tracking\_cc.h>

Inheritance diagram for Galileo\_E1\_Tcp\_Connector\_Tracking\_cc:



Collaboration diagram for Galileo\_E1\_Tcp\_Connector\_Tracking\_cc:



### Public Member Functions

- void **set\_channel** (uint32\_t channel)
- void **set\_gnss\_synchro** ([Gnss\\_Synchro](#) \*p\_gnss\_synchro)
- void **start\_tracking** ()
- int **general\_work** (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)
- void **forecast** (int noutput\_items, gr\_vector\_int &ninput\_items\_required)

### Friends

- galileo\_e1\_tcp\_connector\_tracking\_cc\_sptr **galileo\_e1\_tcp\_connector\_make\_tracking\_cc** (int64\_t fs\_in, uint32\_t vector\_length, bool dump, const std::string &dump\_filename, float pll\_bw\_hz, float dll\_bw\_hz, float early\_late\_space\_chips, float very\_early\_late\_space\_chips, size\_t port\_ch0)

### 10.112.1 Detailed Description

This class implements a code DLL + carrier PLL VEML (Very Early Minus Late) tracking block for Galileo E1 signals. Definition at line 63 of file [galileo\\_e1\\_tcp\\_connector\\_tracking\\_cc.h](#).

The documentation for this class was generated from the following file:

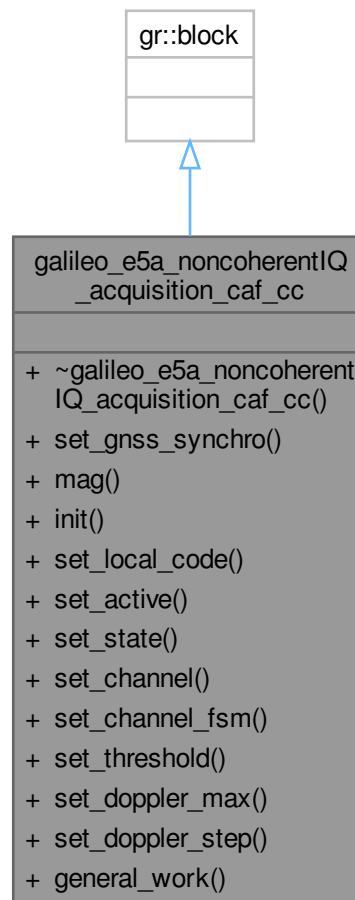
- [galileo\\_e1\\_tcp\\_connector\\_tracking\\_cc.h](#)

## 10.113 galileo\_e5a\_noncoherentIQ\_acquisition\_caf\_cc Class Reference

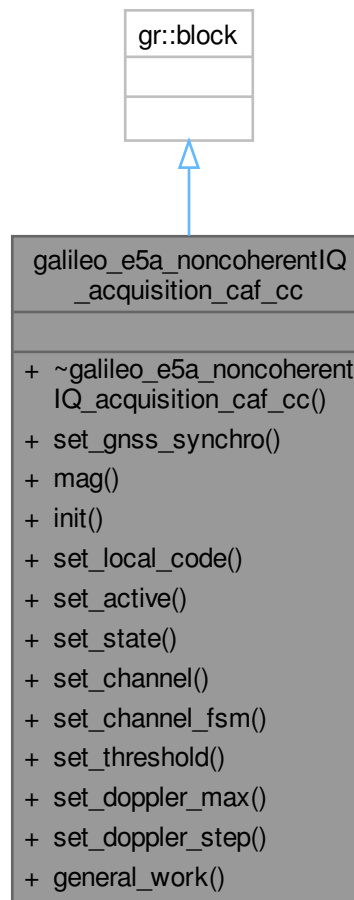
This class implements a Parallel Code Phase Search Acquisition.

```
#include <galileo_e5a_noncoherent_iq_acquisition_caf_cc.h>
```

Inheritance diagram for galileo\_e5a\_noncoherentIQ\_acquisition\_caf\_cc:



Collaboration diagram for `galileo_e5a_noncoherentIQ_acquisition_caf_cc`:



## Public Member Functions

- `~galileo_e5a_noncoherentIQ_acquisition_caf_cc ()`  
*Default destructor.*
- `void set_gnss_synchro (Gnss_Synchro *p_gnss_synchro)`  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to exchange synchronization data between acquisition and tracking blocks.*
- `unsigned int mag () const`  
*Returns the maximum peak of grid search.*
- `void init ()`  
*Initializes acquisition algorithm.*
- `void set_local_code (std::complex< float > *code, std::complex< float > *codeQ)`  
*Sets local code for PCPS acquisition algorithm.*
- `void set_active (bool active)`  
*Starts acquisition algorithm, turning from standby mode to active mode.*
- `void set_state (int state)`  
*If set to 1, ensures that acquisition starts at the first available sample.*
- `void set_channel (unsigned int channel)`



- Set acquisition channel unique ID.*
- void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm)  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold)  
*Set statistics threshold of PCPS algorithm.*
- void [set\\_doppler\\_max](#) (unsigned int doppler\_max)  
*Set maximum Doppler grid search.*
- void [set\\_doppler\\_step](#) (unsigned int doppler\_step)  
*Set Doppler steps for the grid search.*
- int [general\\_work](#) (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)  
*Parallel Code Phase Search Acquisition signal processing.*

## Friends

- galileo\_e5a\_noncoherentIQ\_acquisition\_caf\_cc\_sptr [galileo\\_e5a\\_noncoherentIQ\\_make\\_acquisition\\_caf\\_cc](#) (unsigned int sampled\_ms, unsigned int max\_dwells, unsigned int doppler\_max, int64\_t fs\_in, int samples\_per\_ms, int samples\_per\_code, bool bit\_transition\_flag, bool dump, const std::string &dump\_filename, bool both\_signal\_components\_, int CAF\_window\_hz\_, int Zero\_padding\_, bool enable\_monitor\_output)

## 10.113.1 Detailed Description

This class implements a Parallel Code Phase Search Acquisition.

Check [An Open Source Galileo E1 Software Receiver](#), Algorithm 1, for a pseudocode description of this implementation.

Definition at line 67 of file [galileo\\_e5a\\_noncoherent\\_iq\\_acquisition\\_caf\\_cc.h](#).

## 10.113.2 Constructor & Destructor Documentation

### 10.113.2.1 ~galileo\_e5a\_noncoherentIQ\_acquisition\_caf\_cc()

```
galileo_e5a_noncoherentIQ_acquisition_caf_cc::~galileo_e5a_noncoherentIQ_acquisition_caf_cc ()
```

Default destructor.

## 10.113.3 Member Function Documentation

### 10.113.3.1 general\_work()

```
int galileo_e5a_noncoherentIQ_acquisition_caf_cc::general_work (
    int noutput_items,
    gr_vector_int & ninput_items,
    gr_vector_const_void_star & input_items,
    gr_vector_void_star & output_items)
```

Parallel Code Phase Search Acquisition signal processing.

### 10.113.3.2 init()

```
void galileo_e5a_noncoherentIQ_acquisition_caf_cc::init ()
```

Initializes acquisition algorithm.

### 10.113.3.3 mag()

```
unsigned int galileo_e5a_noncoherentIQ_acquisition_caf_cc::mag () const [inline]
```

Returns the maximum peak of grid search.

Definition at line 88 of file [galileo\\_e5a\\_noncoherent\\_iq\\_acquisition\\_caf\\_cc.h](#).

**10.113.3.4 set\_active()**

```
void galileo_e5a_noncoherentIQ_acquisition_caf_cc::set_active (
    bool active) [inline]
```

Starts acquisition algorithm, turning from standby mode to active mode.

**Parameters**

<i>active</i>	- bool that activates/deactivates the block.
---------------	--

Definition at line 109 of file [galileo\\_e5a\\_noncoherent\\_iq\\_acquisition\\_caf\\_cc.h](#).

**10.113.3.5 set\_channel()**

```
void galileo_e5a_noncoherentIQ_acquisition_caf_cc::set_channel (
    unsigned int channel) [inline]
```

Set acquisition channel unique ID.

**Parameters**

<i>channel</i>	- receiver channel.
----------------	---------------------

Definition at line 125 of file [galileo\\_e5a\\_noncoherent\\_iq\\_acquisition\\_caf\\_cc.h](#).

**10.113.3.6 set\_channel\_fsm()**

```
void galileo_e5a_noncoherentIQ_acquisition_caf_cc::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline]
```

Set channel fsm associated to this acquisition instance.

Definition at line 133 of file [galileo\\_e5a\\_noncoherent\\_iq\\_acquisition\\_caf\\_cc.h](#).

**10.113.3.7 set\_doppler\_max()**

```
void galileo_e5a_noncoherentIQ_acquisition_caf_cc::set_doppler_max (
    unsigned int doppler_max) [inline]
```

Set maximum Doppler grid search.

**Parameters**

<i>doppler_max</i>	- Maximum Doppler shift considered in the grid search [Hz].
--------------------	---

Definition at line 152 of file [galileo\\_e5a\\_noncoherent\\_iq\\_acquisition\\_caf\\_cc.h](#).

**10.113.3.8 set\_doppler\_step()**

```
void galileo_e5a_noncoherentIQ_acquisition_caf_cc::set_doppler_step (
    unsigned int doppler_step) [inline]
```

Set Doppler steps for the grid search.

**Parameters**

<i>doppler_step</i>	- Frequency bin of the search grid [Hz].
---------------------	--

Definition at line 161 of file [galileo\\_e5a\\_noncoherent\\_iq\\_acquisition\\_caf\\_cc.h](#).

#### 10.113.3.9 set\_gnss\_synchro()

```
void galileo_e5a_noncoherentIQ_acquisition_caf_cc::set_gnss_synchro (  
    Gnss_Synchro * p_gnss_synchro) [inline]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to exchange synchronization data between acquisition and tracking blocks.

## Parameters

<i>p_gnss_synchro</i>	Satellite information shared by the processing blocks.
-----------------------	--

Definition at line 80 of file [galileo\\_e5a\\_noncoherent\\_iq\\_acquisition\\_caf\\_cc.h](#).

**10.113.3.10 set\_local\_code()**

```
void galileo_e5a_noncoherentIQ_acquisition_caf_cc::set_local_code (
    std::complex< float > * code,
    std::complex< float > * codeQ)
```

Sets local code for PCPS acquisition algorithm.

## Parameters

<i>code</i>	- Pointer to the PRN code.
-------------	----------------------------

**10.113.3.11 set\_state()**

```
void galileo_e5a_noncoherentIQ_acquisition_caf_cc::set_state (
    int state)
```

If set to 1, ensures that acquisition starts at the first available sample.

## Parameters

<i>state</i>	- int=1 forces start of acquisition
--------------	-------------------------------------

**10.113.3.12 set\_threshold()**

```
void galileo_e5a_noncoherentIQ_acquisition_caf_cc::set_threshold (
    float threshold) [inline]
```

Set statistics threshold of PCPS algorithm.

## Parameters

<i>threshold</i>	- Threshold for signal detection (check <a href="#">Navitec2012</a> , Algorithm 1, for a definition of this threshold).
------------------	---

Definition at line 143 of file [galileo\\_e5a\\_noncoherent\\_iq\\_acquisition\\_caf\\_cc.h](#).

The documentation for this class was generated from the following file:

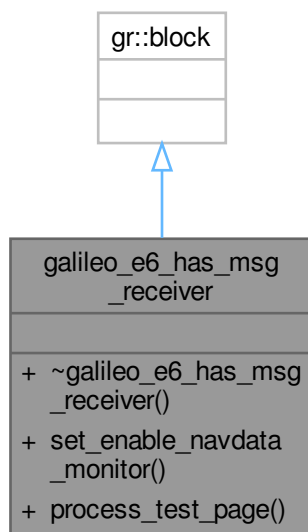
- [galileo\\_e5a\\_noncoherent\\_iq\\_acquisition\\_caf\\_cc.h](#)

**10.114 galileo\_e6\_has\_msg\_receiver Class Reference**

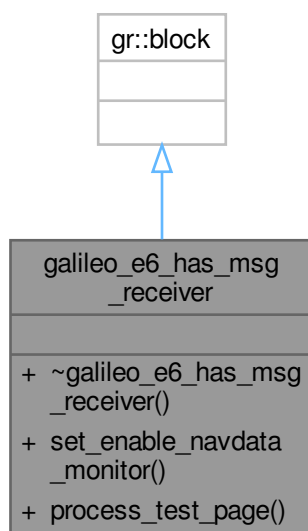
GNU Radio block that receives asynchronous Galileo HAS message pages from the telemetry blocks, stores them in memory, and decodes HAS messages when enough data have been received. The decoded HAS message is sent to the PVT block.

```
#include <galileo_e6_has_msg_receiver.h>
```

Inheritance diagram for galileo\_e6\_has\_msg\_receiver:



Collaboration diagram for galileo\_e6\_has\_msg\_receiver:



### Public Member Functions

- [~galileo\\_e6\\_has\\_msg\\_receiver](#) ()=default  
*Default destructor.*

- void **set\_enable\_navdata\_monitor** (bool enable)
- std::shared\_ptr< [Galileo\\_HAS\\_data](#) > [process\\_test\\_page](#) (const pmt::pmt\_t &msg)

*For testing purposes only.*

## Friends

- galileo\_e6\_has\_msg\_receiver\_sptr **galileo\_e6\_has\_msg\_receiver\_make** ()

### 10.114.1 Detailed Description

GNU Radio block that receives asynchronous Galileo HAS message pages from the telemetry blocks, stores them in memory, and decodes HAS messages when enough data have been received. The decoded HAS message is sent to the PVT block.

Definition at line 56 of file [galileo\\_e6\\_has\\_msg\\_receiver.h](#).

### 10.114.2 Constructor & Destructor Documentation

#### 10.114.2.1 ~galileo\_e6\_has\_msg\_receiver()

```
galileo_e6_has_msg_receiver::~galileo_e6_has_msg_receiver () [default]
```

Default destructor.

### 10.114.3 Member Function Documentation

#### 10.114.3.1 process\_test\_page()

```
std::shared_ptr< Galileo\_HAS\_data > galileo_e6_has_msg_receiver::process_test_page (
    const pmt::pmt_t & msg)
```

For testing purposes only.

The documentation for this class was generated from the following file:

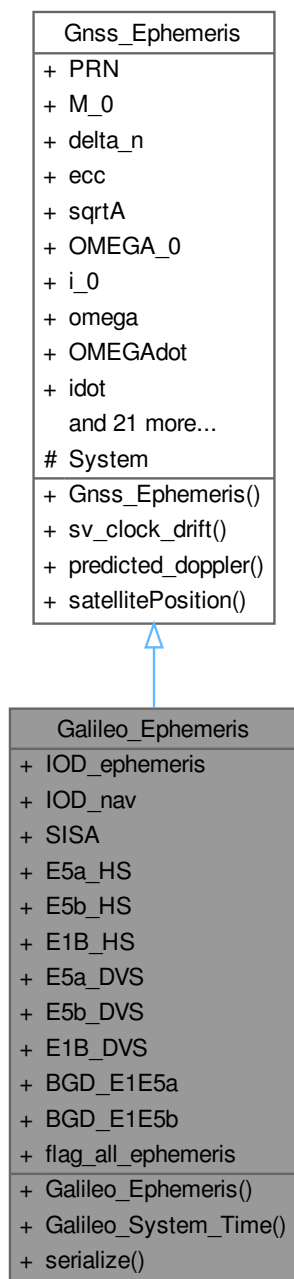
- [galileo\\_e6\\_has\\_msg\\_receiver.h](#)

## 10.115 Galileo\_Ephemeris Class Reference

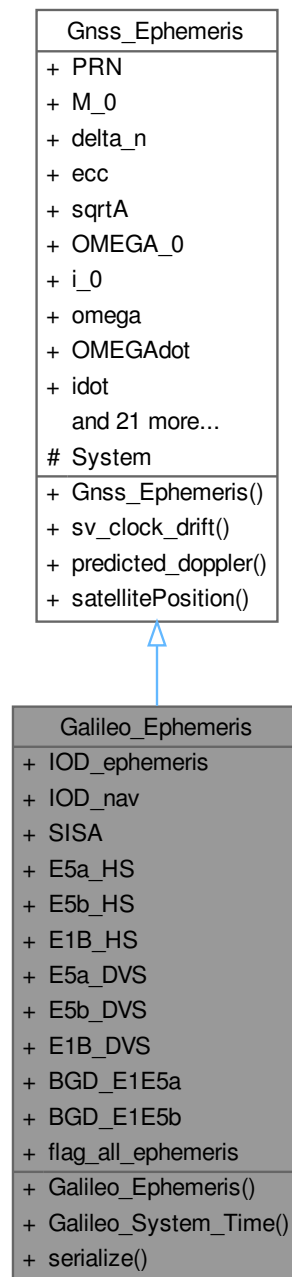
This class is a storage and orbital model functions for the Galileo SV ephemeris data as described in Galileo ICD paragraph 5.1.1.

```
#include <galileo_ephemeris.h>
```

Inheritance diagram for Galileo\_Ephemeris:



Collaboration diagram for Galileo\_Ephemeris:



## Public Member Functions

- double [Galileo\\_System\\_Time](#) (double week\_number, double TOW)

*Galileo System Time (GST), ICD paragraph 5.1.2.*

- template<class Archive>  
void [serialize](#) (Archive &archive, const uint32\_t version)

*Serialize is a boost standard method to be called by the boost XML serialization. Here is used to save the ephemeris data on disk file.*



## Public Member Functions inherited from Gnss\_Ephemeris

- double [sv\\_clock\\_drift](#) (double transmitTime)  
*Sets (satClkDrift) and (dtr), and returns the clock drift in seconds according to the User Algorithm for SV Clock Correction (IS-GPS-200M, 20.3.3.3.1, and Galileo OS SIS ICD, 5.1.4).*
- double [predicted\\_doppler](#) (double rx\_time\_s, double lat, double lon, double h, double ve, double vn, double vu, int band) const  
*Computes prediction of the Doppler shift for a given time and receiver's position and velocity.*
- void [satellitePosition](#) (double transmitTime)  
*Computes the ECEF SV coordinates and ECEF velocity.*

## Public Attributes

- int32\_t [IOD\\_ephemeris](#) {}
- int32\_t [IOD\\_nav](#) {}
- int32\_t [SISA](#) {}  
*Signal in space accuracy index.*
- int32\_t [E5a\\_HS](#) {}  
*E5a Signal Health Status.*
- int32\_t [E5b\\_HS](#) {}  
*E5b Signal Health Status.*
- int32\_t [E1B\\_HS](#) {}  
*E1B Signal Health Status.*
- bool [E5a\\_DVS](#) {}  
*E5a Data Validity Status.*
- bool [E5b\\_DVS](#) {}  
*E5b Data Validity Status.*
- bool [E1B\\_DVS](#) {}  
*E1B Data Validity Status.*
- double [BGD\\_E1E5a](#) {}  
*E1-E5a Broadcast Group Delay [s].*
- double [BGD\\_E1E5b](#) {}  
*E1-E5b Broadcast Group Delay [s].*
- bool [flag\\_all\\_ephemeris](#) {}

## Public Attributes inherited from Gnss\_Ephemeris

- uint32\_t [PRN](#) {}  
*SV ID.*
- double [M\\_0](#) {}  
*Mean anomaly at reference time [rad].*
- double [delta\\_n](#) {}  
*Mean motion difference from computed value [rad/sec].*
- double [ecc](#) {}  
*Eccentricity.*
- double [sqrtA](#) {}  
*Square root of the semi-major axis [meters<sup>1/2</sup>].*
- double [OMEGA\\_0](#) {}  
*Longitude of ascending node of orbital plane at weekly epoch [rad].*
- double [i\\_0](#) {}  
*Inclination angle at reference time [rad].*
- double [omega](#) {}  
*Argument of perigee [rad].*

- double `OMEGAdot` {}  
*Rate of right ascension [rad/sec].*
- double `idot` {}  
*Rate of inclination angle [rad/sec].*
- double `Cuc` {}  
*Amplitude of the cosine harmonic correction term to the argument of latitude [rad].*
- double `Cus` {}  
*Amplitude of the sine harmonic correction term to the argument of latitude [rad].*
- double `Crc` {}  
*Amplitude of the cosine harmonic correction term to the orbit radius [meters].*
- double `Crs` {}  
*Amplitude of the sine harmonic correction term to the orbit radius [meters].*
- double `Cic` {}  
*Amplitude of the cosine harmonic correction term to the angle of inclination [rad].*
- double `Cis` {}  
*Amplitude of the sine harmonic correction term to the angle of inclination [rad].*
- int32\_t `toe` {}  
*Ephemeris reference time [s].*
- int32\_t `toc` {}  
*Clock correction data reference Time of Week [sec].*
- double `af0` {}  
*SV clock bias correction coefficient [s].*
- double `af1` {}  
*SV clock drift correction coefficient [s/s].*
- double `af2` {}  
*SV clock drift rate correction coefficient [s/s<sup>2</sup>].*
- double `satClkDrift` {}  
*SV clock drift.*
- double `dtr` {}  
*Relativistic clock correction term.*
- int32\_t `WN` {}  
*Week number.*
- int32\_t `tow` {}  
*Time of Week.*
- double `satpos_X` {}  
*Earth-fixed coordinate x of the satellite [m]. Intersection of the IERS Reference Meridian (IRM) and the plane passing through the origin and normal to the Z-axis.*
- double `satpos_Y` {}  
*Earth-fixed coordinate y of the satellite [m]. Completes a right-handed, Earth-Centered, Earth-Fixed orthogonal coordinate system.*
- double `satpos_Z` {}  
*Earth-fixed coordinate z of the satellite [m]. The direction of the IERS (International Earth Rotation and Reference Systems Service) Reference Pole (IRP).*
- double `satvel_X` {}  
*Earth-fixed velocity coordinate x of the satellite [m].*
- double `satvel_Y` {}  
*Earth-fixed velocity coordinate y of the satellite [m].*
- double `satvel_Z` {}  
*Earth-fixed velocity coordinate z of the satellite [m].*

## Additional Inherited Members

### Protected Attributes inherited from [Gnss\\_Ephemeris](#)

- char [System](#) {}

*Character ID of the GNSS system. 'G': GPS. 'E': Galileo. 'B': BeiDou.*

### 10.115.1 Detailed Description

This class is a storage and orbital model functions for the Galileo SV ephemeris data as described in Galileo ICD paragraph 5.1.1.

(See [https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo%20OS\\_SIS\\_ICD\\_v2.0.pdf](https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo%20OS_SIS_ICD_v2.0.pdf))

Definition at line 39 of file [galileo\\_ephemeris.h](#).

### 10.115.2 Constructor & Destructor Documentation

#### 10.115.2.1 Galileo\_Ephemeris()

```
Galileo_Ephemeris::Galileo_Ephemeris () [inline]
```

Definition at line 42 of file [galileo\\_ephemeris.h](#).

### 10.115.3 Member Function Documentation

#### 10.115.3.1 Galileo\_System\_Time()

```
double Galileo_Ephemeris::Galileo_System_Time (
    double week_number,
    double TOW)
```

Galileo System Time (GST), ICD paragraph 5.1.2.

#### 10.115.3.2 serialize()

```
template<class Archive>
void Galileo_Ephemeris::serialize (
    Archive & archive,
    const uint32_t version) [inline]
```

Serialize is a boost standard method to be called by the boost XML serialization. Here is used to save the ephemeris data on disk file.

Definition at line 71 of file [galileo\\_ephemeris.h](#).

References [Gnss\\_Ephemeris::af0](#), [Gnss\\_Ephemeris::af1](#), [Gnss\\_Ephemeris::af2](#), [BGD\\_E1E5a](#), [BGD\\_E1E5b](#), [Gnss\\_Ephemeris::Cic](#), [Gnss\\_Ephemeris::Cis](#), [Gnss\\_Ephemeris::Crc](#), [Gnss\\_Ephemeris::Crs](#), [Gnss\\_Ephemeris::Cuc](#), [Gnss\\_Ephemeris::Cus](#), [Gnss\\_Ephemeris::delta\\_n](#), [Gnss\\_Ephemeris::dtr](#), [E1B\\_DVS](#), [E1B\\_HS](#), [E5a\\_DVS](#), [E5a\\_HS](#), [E5b\\_DVS](#), [E5b\\_HS](#), [Gnss\\_Ephemeris::ecc](#), [Gnss\\_Ephemeris::i\\_0](#), [Gnss\\_Ephemeris::idot](#), [Gnss\\_Ephemeris::M\\_0](#), [Gnss\\_Ephemeris::omega](#), [Gnss\\_Ephemeris::OMEGA\\_0](#), [Gnss\\_Ephemeris::OMEGAdot](#), [Gnss\\_Ephemeris::PRN](#), [Gnss\\_Ephemeris::satClkDrift](#), [SISA](#), [Gnss\\_Ephemeris::sqrtA](#), [Gnss\\_Ephemeris::toc](#), [Gnss\\_Ephemeris::toe](#), [Gnss\\_Ephemeris::tow](#), and [Gnss\\_Ephemeris::WN](#).

### 10.115.4 Member Data Documentation

#### 10.115.4.1 BGD\_E1E5a

```
double Galileo_Ephemeris::BGD_E1E5a {}
```

E1-E5a Broadcast Group Delay [s].

Definition at line 60 of file [galileo\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), and [serialize\(\)](#).

#### 10.115.4.2 BGD\_E1E5b

```
double Galileo_Ephemeris::BGD_E1E5b {}
```

E1-E5b Broadcast Group Delay [s].

Definition at line 61 of file [galileo\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), and [serialize\(\)](#).

#### 10.115.4.3 E1B\_DVS

```
bool Galileo_Ephemeris::E1B_DVS {}
```

E1B Data Validity Status.

Definition at line 59 of file [galileo\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), and [serialize\(\)](#).

#### 10.115.4.4 E1B\_HS

```
int32_t Galileo_Ephemeris::E1B_HS {}
```

E1B Signal Health Status.

Definition at line 56 of file [galileo\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), and [serialize\(\)](#).

#### 10.115.4.5 E5a\_DVS

```
bool Galileo_Ephemeris::E5a_DVS {}
```

E5a Data Validity Status.

Definition at line 57 of file [galileo\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), and [serialize\(\)](#).

#### 10.115.4.6 E5a\_HS

```
int32_t Galileo_Ephemeris::E5a_HS {}
```

E5a Signal Health Status.

Definition at line 54 of file [galileo\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), and [serialize\(\)](#).

#### 10.115.4.7 E5b\_DVS

```
bool Galileo_Ephemeris::E5b_DVS {}
```

E5b Data Validity Status.

Definition at line 58 of file [galileo\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), and [serialize\(\)](#).

#### 10.115.4.8 E5b\_HS

```
int32_t Galileo_Ephemeris::E5b_HS {}
```

E5b Signal Health Status.

Definition at line 55 of file [galileo\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), and [serialize\(\)](#).

#### 10.115.4.9 flag\_all\_ephemeris

```
bool Galileo_Ephemeris::flag_all_ephemeris {}
```

Definition at line 63 of file [galileo\\_ephemeris.h](#).

#### 10.115.4.10 IOD\_ephemeris

```
int32_t Galileo_Ephemeris::IOD_ephemeris {}
```

Definition at line 49 of file [galileo\\_ephemeris.h](#).

#### 10.115.4.11 IOD\_nav

```
int32_t Galileo_Ephemeris::IOD_nav {}
```

Definition at line 50 of file [galileo\\_ephemeris.h](#).

#### 10.115.4.12 SISA

```
int32_t Galileo_Ephemeris::SISA {}
```

Signal in space accuracy index.

Definition at line 53 of file [galileo\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), and [serialize\(\)](#).

The documentation for this class was generated from the following file:

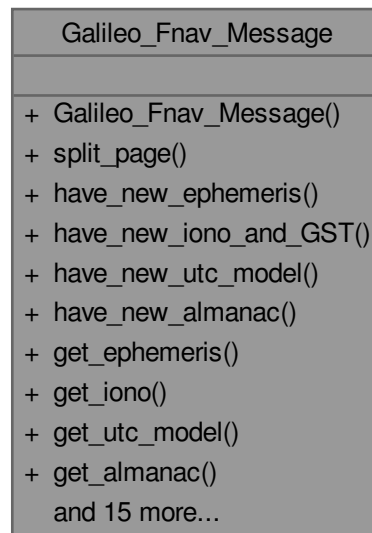
- [galileo\\_ephemeris.h](#)

### 10.116 Galileo\_Fnav\_Message Class Reference

This class handles the Galileo F/NAV Data message, as described in the Galileo Open Service Signal in Space Interface Control Document (OS SIS ICD), Issue 2.0 (Jan. 2021). See [https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo\\_OS\\_SIS\\_ICD\\_v2.0.pdf](https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.0.pdf).

```
#include <galileo_fnav_message.h>
```

Collaboration diagram for Galileo\_Fnav\_Message:



#### Public Member Functions

- void **split\_page** (const std::string &page\_string)
- bool **have\_new\_ephemeris** ()
- bool **have\_new\_iono\_and\_GST** ()
- bool **have\_new\_utc\_model** ()
- bool **have\_new\_almanac** ()
- [Galileo\\_Ephemeris](#) **get\_ephemeris** () const
- [Galileo\\_Iono](#) **get\_iono** () const
- [Galileo\\_Utc\\_Model](#) **get\_utc\_model** () const
- [Galileo\\_Almanac\\_Helper](#) **get\_almanac** () const
- int32\_t **get\_TOW1** () const
- int32\_t **get\_TOW2** () const
- int32\_t **get\_TOW3** () const

- `int32_t get_TOW4 () const`
- `bool get_flag_CRC_test () const`
- `bool get_flag_TOW_set () const`
- `void set_flag_TOW_set (bool flag_tow)`
- `bool is_TOW1_set () const`
- `void set_TOW1_flag (bool flag_tow1)`
- `bool is_TOW2_set () const`
- `void set_TOW2_flag (bool flag_tow2)`
- `bool is_TOW3_set () const`
- `void set_TOW3_flag (bool flag_tow3)`
- `bool is_TOW4_set () const`
- `void set_TOW4_flag (bool flag_tow4)`

### 10.116.1 Detailed Description

This class handles the Galileo F/NAV Data message, as described in the Galileo Open Service Signal in Space Interface Control Document (OS SIS ICD), Issue 2.0 (Jan. 2021). See [https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo\\_OS\\_SIS\\_ICD\\_v2.0.pdf](https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.0.pdf).

Definition at line 50 of file [galileo\\_fnav\\_message.h](#).

### 10.116.2 Member Function Documentation

#### 10.116.2.1 `get_flag_CRC_test()`

```
bool Galileo_Fnav_Message::get_flag_CRC_test () const [inline]
```

Definition at line 85 of file [galileo\\_fnav\\_message.h](#).

#### 10.116.2.2 `get_flag_TOW_set()`

```
bool Galileo_Fnav_Message::get_flag_TOW_set () const [inline]
```

Definition at line 90 of file [galileo\\_fnav\\_message.h](#).

#### 10.116.2.3 `get_TOW1()`

```
int32_t Galileo_Fnav_Message::get_TOW1 () const [inline]
```

Definition at line 65 of file [galileo\\_fnav\\_message.h](#).

#### 10.116.2.4 `get_TOW2()`

```
int32_t Galileo_Fnav_Message::get_TOW2 () const [inline]
```

Definition at line 70 of file [galileo\\_fnav\\_message.h](#).

#### 10.116.2.5 `get_TOW3()`

```
int32_t Galileo_Fnav_Message::get_TOW3 () const [inline]
```

Definition at line 75 of file [galileo\\_fnav\\_message.h](#).

#### 10.116.2.6 `get_TOW4()`

```
int32_t Galileo_Fnav_Message::get_TOW4 () const [inline]
```

Definition at line 80 of file [galileo\\_fnav\\_message.h](#).

#### 10.116.2.7 `is_TOW1_set()`

```
bool Galileo_Fnav_Message::is_TOW1_set () const [inline]
```

Definition at line 100 of file [galileo\\_fnav\\_message.h](#).

**10.116.2.8 is\_TOW2\_set()**

bool Galileo\_Fnav\_Message::is\_TOW2\_set () const [inline]  
 Definition at line 110 of file [galileo\\_fnav\\_message.h](#).

**10.116.2.9 is\_TOW3\_set()**

bool Galileo\_Fnav\_Message::is\_TOW3\_set () const [inline]  
 Definition at line 120 of file [galileo\\_fnav\\_message.h](#).

**10.116.2.10 is\_TOW4\_set()**

bool Galileo\_Fnav\_Message::is\_TOW4\_set () const [inline]  
 Definition at line 130 of file [galileo\\_fnav\\_message.h](#).

**10.116.2.11 set\_flag\_TOW\_set()**

void Galileo\_Fnav\_Message::set\_flag\_TOW\_set (   
     bool *flag\_tow*) [inline]  
 Definition at line 95 of file [galileo\\_fnav\\_message.h](#).

**10.116.2.12 set\_TOW1\_flag()**

void Galileo\_Fnav\_Message::set\_TOW1\_flag (   
     bool *flag\_tow1*) [inline]  
 Definition at line 105 of file [galileo\\_fnav\\_message.h](#).

**10.116.2.13 set\_TOW2\_flag()**

void Galileo\_Fnav\_Message::set\_TOW2\_flag (   
     bool *flag\_tow2*) [inline]  
 Definition at line 115 of file [galileo\\_fnav\\_message.h](#).

**10.116.2.14 set\_TOW3\_flag()**

void Galileo\_Fnav\_Message::set\_TOW3\_flag (   
     bool *flag\_tow3*) [inline]  
 Definition at line 125 of file [galileo\\_fnav\\_message.h](#).

**10.116.2.15 set\_TOW4\_flag()**

void Galileo\_Fnav\_Message::set\_TOW4\_flag (   
     bool *flag\_tow4*) [inline]  
 Definition at line 135 of file [galileo\\_fnav\\_message.h](#).

The documentation for this class was generated from the following file:

- [galileo\\_fnav\\_message.h](#)

**10.117 Galileo\_HAS\_data Class Reference**

This class is a storage for Galileo HAS message type 1, as defined in Galileo High Accuracy Service Signal-In-Space Interface Control Document (HAS SIS ICD) Issue 1.0, May 2022. See [https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo\\_HAS\\_SIS\\_ICD\\_v1.0.pdf](https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_HAS_SIS_ICD_v1.0.pdf).  
 #include <galileo\_has\_data.h>





- `std::vector< float > get_delta_cross_track_m () const`  
*Get Nsat delta cross-track corrections in [m].*
- `std::vector< float > get_delta_cross_track_m (uint8_t nsys) const`  
*Get delta cross-track corrections in [m] for system nsys, with  $0 \leq nsys < Nsys$ .*
- `std::vector< float > get_delta_clock_correction_m () const`  
*Get Nsat delta clock C0 corrections in [m].*
- `std::vector< float > get_delta_clock_correction_m (uint8_t nsys) const`  
*Get delta clock C0 corrections in [m] for system nsys, with  $0 \leq nsys < Nsys$ .*
- `std::vector< float > get_delta_clock_subset_correction_m (uint8_t nsys) const`  
*Get delta clock C0 subset corrections in [m] for system nsys, with  $0 \leq nsys < Nsys$ .*
- `std::vector< int > get_PRNs_in_mask (uint8_t nsys) const`  
*Get PRNs in mask for system nsys, with  $0 \leq nsys < Nsys$ .*
- `std::vector< int > get_PRNs_in_mask (const std::string &system) const`  
*Get PRNs in mask for system ("GPS"/"Galileo")*
- `std::vector< int > get_PRNs_in_submask (uint8_t nsys) const`  
*Get PRNs in submask for system nsys, with  $0 \leq nsys < Nsys$ .*
- `std::vector< uint16_t > get_gnss_iod (uint8_t nsys) const`  
*Get GNSS IODs for for system nsys, with  $0 \leq nsys < Nsys$ .*
- `std::vector< uint8_t > get_num_satellites () const`  
*Get Nsys number of satellites.*
- `std::vector< uint8_t > get_num_subset_satellites () const`  
*Get Nsys\_sub number of satellites.*
- `float get_code_bias_m (const std::string &signal, int PRN) const`  
*Get code bias in [m] for a given signal and PRN satellite.*
- `float get_phase_bias_cycle (const std::string &signal, int PRN) const`  
*Get phase bias in [cycles] for a given signal and PRN satellite.*
- `float get_delta_radial_m (const std::string &system, int prn) const`  
*Get orbital radial correction in [m] for a given system ("GPS"/"Galileo") and PRN.*
- `float get_delta_in_track_m (const std::string &system, int prn) const`  
*Get orbital in\_track correction in [m] for a given system ("GPS"/"Galileo") and PRN.*
- `float get_delta_cross_track_m (const std::string &system, int prn) const`  
*Get orbital cross\_track correction in [m] for a given system ("GPS"/"Galileo") and PRN.*
- `float get_clock_correction_mult_m (const std::string &system, int prn) const`  
*Get clock correction in [m], already multiplied by its Delta Clock Multiplier, for a given system ("GPS"/"Galileo") and PRN.*
- `float get_clock_subset_correction_mult_m (const std::string &system, int prn) const`  
*Get clock correction subset in [m], already multiplied by its Delta Clock Multiplier.*
- `uint16_t get_nsat () const`  
*Get total number of satellites with corrections.*
- `uint16_t get_nsat_sub () const`  
*Get number of satellites in clock subset corrections.*
- `uint16_t get_validity_interval_s (uint8_t validity_interval_index) const`  
*Get validity interval in [s] from the validity\_interval\_index.*
- `uint16_t get_gnss_iod (const std::string &system, int prn) const`  
*Get GNSS IOD from a given system ("GPS"/"Galileo") and PRN.*
- `uint8_t get_gnss_id (int nsat) const`  
*Get GNSS ID from the nsat satellite.*

## Public Attributes

- `std::vector< uint8_t > gnss_id_mask`  
GNSS ID. See HAS SIS ICD 1.0 Section 5.2.1.1.
- `std::vector< uint64_t > satellite_mask`  
SatM - Satellite Mask. See HAS SIS ICD 1.0 Section 5.2.1.2.
- `std::vector< uint16_t > signal_mask`  
SigM - Signal Mask. See HAS SIS ICD 1.0 Section 5.2.1.3.
- `std::vector< bool > cell_mask_availability_flag`  
CMAF - Cell Mask Availability Flag. See HAS SIS ICD 1.0 Section 5.2.1.4.
- `std::vector< std::vector< std::vector< bool > > > cell_mask`  
CM - Cell Mask. See HAS SIS ICD 1.0 Section 5.2.1.5.
- `std::vector< uint8_t > nav_message`  
NM - Navigation Message Index. See HAS SIS ICD 1.0 Section 5.2.1.6.
- `std::vector< uint16_t > gnss_iod`  
IODref - Reference Issue of Data. See HAS SIS ICD 1.0 Table 26.
- `std::vector< int16_t > delta_radial`  
DR - Delta Radial Correction. See HAS SIS ICD 1.0 Table 25.
- `std::vector< int16_t > delta_in_track`  
DIT - Delta In-Track Correction. See HAS SIS ICD 1.0 Table 25.
- `std::vector< int16_t > delta_cross_track`  
DCT - Delta Cross Correction. See HAS SIS ICD 1.0 Table 25.
- `std::vector< uint8_t > delta_clock_multiplier`  
DCM - Delta Clock Multipliers. See HAS SIS ICD 1.0 Section 5.2.3.1.
- `std::vector< int16_t > delta_clock_correction`  
DCC - Delta Clock Corrections. See HAS SIS ICD 1.0 Section 5.2.3.2.
- `std::vector< uint8_t > gnss_id_clock_subset`  
GNSS ID. Specific GNSS to which the corrections refer. See HAS SIS ICD 1.0 Section 5.2.1.1.
- `std::vector< uint8_t > delta_clock_multiplier_clock_subset`  
DCM. Multiplier for all Delta Clock corrections. See HAS SIS ICD 1.0 Section 5.2.3.1.
- `std::vector< uint64_t > satellite_submask`  
SatMsub - Satellite Subset Mask. See HAS SIS ICD 1.0 Section 5.2.4.1.
- `std::vector< std::vector< int16_t > > delta_clock_correction_clock_subset`  
DCCsub - Delta Clock Subset Corrections. See HAS SIS ICD 1.0 Section 5.2.4.1.
- `std::vector< std::vector< int16_t > > code_bias`  
CB - Code bias for the  $m$ -th signal of the  $n$ -th SV. See HAS SIS ICD 1.0 Section 5.2.5.
- `std::vector< std::vector< int16_t > > phase_bias`  
PB - Phase bias for the  $m$ -th signal of the  $n$ -th SV. See HAS SIS ICD 1.0 Section 5.2.6.
- `std::vector< std::vector< uint8_t > > phase_discontinuity_indicator`  
PDI - Phase Discontinuity Indicator. See HAS SIS ICD 1.0 Section 5.2.6.
- `uint32_t tow`  
Time of Week.
- `mt1_header header`  
MT1 Header parameters. See HAS SIS ICD 1.0 Section 5.1.1.
- `uint8_t has_status`  
HASS - HAS Status (from HAS page header). See HAS SIS ICD 1.0 Section 3.1.1.
- `uint8_t message_id`  
MID - Message ID (from HAS page header). See HAS SIS ICD 1.0 Section 3.1.
- `uint8_t Nsys`  
Number of GNSS for which corrections are provided. See HAS SIS ICD 1.0 Section 5.2.1.
- `uint8_t Nsys_sub`

Number of GNSS for which corrections are provided in clock subset corrections. See HAS SIS ICD 1.0 Section 5.2.2.1.

- `uint8_t validity_interval_index_orbit_corrections`  
VI - Validity Interval Index for Orbit corrections. See HAS SIS ICD 1.0 Section 5.2.2.1.
- `uint8_t validity_interval_index_clock_fullset_corrections`  
VI - Validity Interval Index for Clock full-set corrections. See HAS SIS ICD 1.0 Section 5.2.2.1.
- `uint8_t validity_interval_index_clock_subset_corrections`  
VI - Validity Interval Index for Clock subset corrections. See HAS SIS ICD 1.0 Section 5.2.2.1.
- `uint8_t validity_interval_index_code_bias_corrections`  
VI - Validity Interval Index for Code bias. See HAS SIS ICD 1.0 Section 5.2.2.1.
- `uint8_t validity_interval_index_phase_bias_corrections`  
VI - Validity Interval Index for Phase bias. See HAS SIS ICD 1.0 Section 5.2.2.1.

### 10.117.1 Detailed Description

This class is a storage for Galileo HAS message type 1, as defined in Galileo High Accuracy Service Signal-In-Space Interface Control Document (HAS SIS ICD) Issue 1.0, May 2022. See [https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo\\_HAS\\_SIS\\_ICD\\_v1.0.pdf](https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_HAS_SIS_ICD_v1.0.pdf).

Definition at line 50 of file `galileo_has_data.h`.

### 10.117.2 Member Function Documentation

#### 10.117.2.1 `get_clock_correction_mult_m()`

```
float Galileo_HAS_data::get_clock_correction_mult_m (
    const std::string & system,
    int prn) const
```

Get clock correction in [m], already multiplied by its Delta Clock Multiplier, for a given system ("GPS"/"Galileo") and PRN.

#### 10.117.2.2 `get_clock_subset_correction_mult_m()`

```
float Galileo_HAS_data::get_clock_subset_correction_mult_m (
    const std::string & system,
    int prn) const
```

Get clock correction subset in [m], already multiplied by its Delta Clock Multiplier.

#### 10.117.2.3 `get_code_bias_m()` [1/2]

```
std::vector< std::vector< float > > Galileo_HAS_data::get_code_bias_m () const
```

Get Nsat x Ncodes code biases in [m].

#### 10.117.2.4 `get_code_bias_m()` [2/2]

```
float Galileo_HAS_data::get_code_bias_m (
    const std::string & signal,
    int PRN) const
```

Get code bias in [m] for a given signal and PRN satellite.

#### 10.117.2.5 `get_delta_clock_correction_m()` [1/2]

```
std::vector< float > Galileo_HAS_data::get_delta_clock_correction_m () const
```

Get Nsat delta clock C0 corrections in [m].

#### 10.117.2.6 `get_delta_clock_correction_m()` [2/2]

```
std::vector< float > Galileo_HAS_data::get_delta_clock_correction_m (
    uint8_t nsys) const
```

Get delta clock C0 corrections in [m] for system nsys, with  $0 \leq nsys < Nsys$ .

**10.117.2.7 get\_delta\_clock\_subset\_correction\_m() [1/2]**

```
std::vector< std::vector< float > > Galileo_HAS_data::get_delta_clock_subset_correction_m ()
const
```

Get Nsys\_sub vectors with Nsat\_sub delta clock C0 corrections in [m].

**10.117.2.8 get\_delta\_clock\_subset\_correction\_m() [2/2]**

```
std::vector< float > Galileo_HAS_data::get_delta_clock_subset_correction_m (
    uint8_t nsys) const
```

Get delta clock C0 subset corrections in [m] for system nsys, with  $0 \leq nsys < Nsys$ .

**10.117.2.9 get\_delta\_cross\_track\_m() [1/3]**

```
std::vector< float > Galileo_HAS_data::get_delta_cross_track_m () const
```

Get Nsat delta cross-track corrections in [m].

**10.117.2.10 get\_delta\_cross\_track\_m() [2/3]**

```
float Galileo_HAS_data::get_delta_cross_track_m (
    const std::string & system,
    int prn) const
```

Get orbital cross\_track correction in [m] for a given system ("GPS"/"Galileo") and PRN.

**10.117.2.11 get\_delta\_cross\_track\_m() [3/3]**

```
std::vector< float > Galileo_HAS_data::get_delta_cross_track_m (
    uint8_t nsys) const
```

Get delta cross-track corrections in [m] for system nsys, with  $0 \leq nsys < Nsys$ .

**10.117.2.12 get\_delta\_in\_track\_m() [1/3]**

```
std::vector< float > Galileo_HAS_data::get_delta_in_track_m () const
```

Get Nsat delta in-track corrections in [m].

**10.117.2.13 get\_delta\_in\_track\_m() [2/3]**

```
float Galileo_HAS_data::get_delta_in_track_m (
    const std::string & system,
    int prn) const
```

Get orbital in\_track correction in [m] for a given system ("GPS"/"Galileo") and PRN.

**10.117.2.14 get\_delta\_in\_track\_m() [3/3]**

```
std::vector< float > Galileo_HAS_data::get_delta_in_track_m (
    uint8_t nsys) const
```

Get delta in-track corrections in [m] for system nsys, with  $0 \leq nsys < Nsys$ .

**10.117.2.15 get\_delta\_radial\_m() [1/3]**

```
std::vector< float > Galileo_HAS_data::get_delta_radial_m () const
```

Get Nsat delta radial corrections in [m].

**10.117.2.16 get\_delta\_radial\_m() [2/3]**

```
float Galileo_HAS_data::get_delta_radial_m (
    const std::string & system,
    int prn) const
```

Get orbital radial correction in [m] for a given system ("GPS"/"Galileo") and PRN.

**10.117.2.17 get\_delta\_radial\_m()** [3/3]

```
std::vector< float > Galileo_HAS_data::get_delta_radial_m (
    uint8_t nsys) const
```

Get delta radial corrections in [m] for system nsys, with  $0 \leq nsys < Nsys$ .

**10.117.2.18 get\_gnss\_id()**

```
uint8_t Galileo_HAS_data::get_gnss_id (
    int nsat) const
```

Get GNSS ID from the nsat satellite.

**10.117.2.19 get\_gnss\_iod()** [1/2]

```
uint16_t Galileo_HAS_data::get_gnss_iod (
    const std::string & system,
    int prn) const
```

Get GNSS IOD from a given system ("GPS"/"Galileo") and PRN.

**10.117.2.20 get\_gnss\_iod()** [2/2]

```
std::vector< uint16_t > Galileo_HAS_data::get_gnss_iod (
    uint8_t nsys) const
```

Get GNSS IODs for for system nsys, with  $0 \leq nsys < Nsys$ .

**10.117.2.21 get\_nsat()**

```
uint16_t Galileo_HAS_data::get_nsat () const
```

Get total number of satellites with corrections.

**10.117.2.22 get\_nsat\_sub()**

```
uint16_t Galileo_HAS_data::get_nsat_sub () const
```

Get number of satellites in clock subset corrections.

**10.117.2.23 get\_num\_satellites()**

```
std::vector< uint8_t > Galileo_HAS_data::get_num_satellites () const
```

Get Nsys number of satellites.

**10.117.2.24 get\_num\_subset\_satellites()**

```
std::vector< uint8_t > Galileo_HAS_data::get_num_subset_satellites () const
```

Get Nsys\_sub number of satellites.

**10.117.2.25 get\_phase\_bias\_cycle()** [1/2]

```
std::vector< std::vector< float > > Galileo_HAS_data::get_phase_bias_cycle () const
```

Get Nsat x Nphases phase biases in [cycles].

**10.117.2.26 get\_phase\_bias\_cycle()** [2/2]

```
float Galileo_HAS_data::get_phase_bias_cycle (
    const std::string & signal,
    int PRN) const
```

Get phase bias in [cycles] for a given signal and PRN satellite.

**10.117.2.27 get\_PRNs\_in\_mask() [1/2]**

```
std::vector< int > Galileo_HAS_data::get_PRNs_in_mask (
    const std::string & system) const
```

Get PRNs in mask for system ("GPS"/"Galileo")

**10.117.2.28 get\_PRNs\_in\_mask() [2/2]**

```
std::vector< int > Galileo_HAS_data::get_PRNs_in_mask (
    uint8_t nsys) const
```

Get PRNs in mask for system nsys, with 0 <= nsys < Nsys.

**10.117.2.29 get\_PRNs\_in\_submask()**

```
std::vector< int > Galileo_HAS_data::get_PRNs_in_submask (
    uint8_t nsys) const
```

Get PRNs in submask for system nsys, with 0 <= nsys < Nsys.

**10.117.2.30 get\_signals\_in\_mask() [1/2]**

```
std::vector< std::string > Galileo_HAS_data::get_signals_in_mask (
    const std::string & system) const
```

Get a vector of Nsys std::string with signals in mask for system ("GPS"/"Galileo")

**10.117.2.31 get\_signals\_in\_mask() [2/2]**

```
std::vector< std::string > Galileo_HAS_data::get_signals_in_mask (
    uint8_t nsys) const
```

Get a vector of Nsys std::string with signals in mask for system nsys, with 0 <= nsys < Nsys.

**10.117.2.32 get\_systems\_string()**

```
std::vector< std::string > Galileo_HAS_data::get_systems_string () const
```

Get Nsys system name strings.

**10.117.2.33 get\_systems\_subset\_string()**

```
std::vector< std::string > Galileo_HAS_data::get_systems_subset_string () const
```

Get Nsat system name strings present in clock corrections subset.

**10.117.2.34 get\_validity\_interval\_s()**

```
uint16_t Galileo_HAS_data::get_validity_interval_s (
    uint8_t validity_interval_index) const
```

Get validity interval in [s] from the validity\_interval\_index.

**10.117.3 Member Data Documentation****10.117.3.1 cell\_mask**

```
std::vector<std::vector<std::vector<bool> > > Galileo_HAS_data::cell_mask
```

CM - Cell Mask. See HAS SIS ICD 1.0 Section 5.2.1.5.

Definition at line 95 of file [galileo\\_has\\_data.h](#).

**10.117.3.2 cell\_mask\_availability\_flag**

```
std::vector<bool> Galileo_HAS_data::cell_mask_availability_flag
```

CMAF - Cell Mask Availability Flag. See HAS SIS ICD 1.0 Section 5.2.1.4.

Definition at line 94 of file [galileo\\_has\\_data.h](#).

### 10.117.3.3 code\_bias

`std::vector<std::vector<int16_t> > Galileo_HAS_data::code_bias`

CB - Code bias for the m-th signal of the n-th SV. See HAS SIS ICD 1.0 Section 5.2.5.

Definition at line 115 of file [galileo\\_has\\_data.h](#).

### 10.117.3.4 delta\_clock\_correction

`std::vector<int16_t> Galileo_HAS_data::delta_clock_correction`

DCC - Delta Clock Corrections. See HAS SIS ICD 1.0 Section 5.2.3.2.

Definition at line 106 of file [galileo\\_has\\_data.h](#).

### 10.117.3.5 delta\_clock\_correction\_clock\_subset

`std::vector<std::vector<int16_t> > Galileo_HAS_data::delta_clock_correction_clock_subset`

DCCsub - Delta Clock Subset Corrections. See HAS SIS ICD 1.0 Section 5.2.4.1.

Definition at line 112 of file [galileo\\_has\\_data.h](#).

### 10.117.3.6 delta\_clock\_multiplier

`std::vector<uint8_t> Galileo_HAS_data::delta_clock_multiplier`

DCM - Delta Clock Multipliers. See HAS SIS ICD 1.0 Section 5.2.3.1.

Definition at line 105 of file [galileo\\_has\\_data.h](#).

### 10.117.3.7 delta\_clock\_multiplier\_clock\_subset

`std::vector<uint8_t> Galileo_HAS_data::delta_clock_multiplier_clock_subset`

DCM. Multiplier for all Delta Clock corrections. See HAS SIS ICD 1.0 Section 5.2.3.1.

Definition at line 110 of file [galileo\\_has\\_data.h](#).

### 10.117.3.8 delta\_cross\_track

`std::vector<int16_t> Galileo_HAS_data::delta_cross_track`

DCT - Delta Cross Correction. See HAS SIS ICD 1.0 Table 25.

Definition at line 102 of file [galileo\\_has\\_data.h](#).

### 10.117.3.9 delta\_in\_track

`std::vector<int16_t> Galileo_HAS_data::delta_in_track`

DIT - Delta In-Track Correction. See HAS SIS ICD 1.0 Table 25.

Definition at line 101 of file [galileo\\_has\\_data.h](#).

### 10.117.3.10 delta\_radial

`std::vector<int16_t> Galileo_HAS_data::delta_radial`

DR - Delta Radial Correction. See HAS SIS ICD 1.0 Table 25.

Definition at line 100 of file [galileo\\_has\\_data.h](#).

### 10.117.3.11 gnss\_id\_clock\_subset

`std::vector<uint8_t> Galileo_HAS_data::gnss_id_clock_subset`

GNSS ID. Specific GNSS to which the corrections refer. See HAS SIS ICD 1.0 Section 5.2.1.1.

Definition at line 109 of file [galileo\\_has\\_data.h](#).

### 10.117.3.12 gnss\_id\_mask

`std::vector<uint8_t> Galileo_HAS_data::gnss_id_mask`

GNSS ID. See HAS SIS ICD 1.0 Section 5.2.1.1.

Definition at line 91 of file [galileo\\_has\\_data.h](#).

**10.117.3.13 gnss\_iod**

`std::vector<uint16_t> Galileo_HAS_data::gnss_iod`

IODref - Reference Issue of Data. See HAS SIS ICD 1.0 Table 26.

Definition at line 99 of file [galileo\\_has\\_data.h](#).

**10.117.3.14 has\_status**

`uint8_t Galileo_HAS_data::has_status`

HASS - HAS Status (from HAS page header). See HAS SIS ICD 1.0 Section 3.1.1.

Definition at line 124 of file [galileo\\_has\\_data.h](#).

**10.117.3.15 header**

`mt1_header Galileo_HAS_data::header`

MT1 Header parameters. See HAS SIS ICD 1.0 Section 5.1.1.

Definition at line 123 of file [galileo\\_has\\_data.h](#).

**10.117.3.16 message\_id**

`uint8_t Galileo_HAS_data::message_id`

MID - Message ID (from HAS page header). See HAS SIS ICD 1.0 Section 3.1.

Definition at line 125 of file [galileo\\_has\\_data.h](#).

**10.117.3.17 nav\_message**

`std::vector<uint8_t> Galileo_HAS_data::nav_message`

NM - Navigation Message Index. See HAS SIS ICD 1.0 Section 5.2.1.6.

Definition at line 96 of file [galileo\\_has\\_data.h](#).

**10.117.3.18 Nsys**

`uint8_t Galileo_HAS_data::Nsys`

Number of GNSS for which corrections are provided. See HAS SIS ICD 1.0 Section 5.2.1.

Definition at line 127 of file [galileo\\_has\\_data.h](#).

**10.117.3.19 Nsys\_sub**

`uint8_t Galileo_HAS_data::Nsys_sub`

Number of GNSS for which corrections are provided in clock subset corrections. See HAS SIS ICD 1.0 Section 5.2.2.1.

Definition at line 128 of file [galileo\\_has\\_data.h](#).

**10.117.3.20 phase\_bias**

`std::vector<std::vector<int16_t> > Galileo_HAS_data::phase_bias`

PB - Phase bias for the m-th signal of the n-th SV. See HAS SIS ICD 1.0 Section 5.2.6.

Definition at line 118 of file [galileo\\_has\\_data.h](#).

**10.117.3.21 phase\_discontinuity\_indicator**

`std::vector<std::vector<uint8_t> > Galileo_HAS_data::phase_discontinuity_indicator`

PDI - Phase Discontinuity Indicator. See HAS SIS ICD 1.0 Section 5.2.6.

Definition at line 119 of file [galileo\\_has\\_data.h](#).

**10.117.3.22 satellite\_mask**

`std::vector<uint64_t> Galileo_HAS_data::satellite_mask`

SatM - Satellite Mask. See HAS SIS ICD 1.0 Section 5.2.1.2.

Definition at line 92 of file [galileo\\_has\\_data.h](#).



### 10.117.3.23 satellite\_submask

`std::vector<uint64_t> Galileo_HAS_data::satellite_submask`

SatMsub - Satellite Subset Mask. See HAS SIS ICD 1.0 Section 5.2.4.1.

Definition at line 111 of file [galileo\\_has\\_data.h](#).

### 10.117.3.24 signal\_mask

`std::vector<uint16_t> Galileo_HAS_data::signal_mask`

SigM - Signal Mask. See HAS SIS ICD 1.0 Section 5.2.1.3.

Definition at line 93 of file [galileo\\_has\\_data.h](#).

### 10.117.3.25 tow

`uint32_t Galileo_HAS_data::tow`

Time of Week.

Definition at line 121 of file [galileo\\_has\\_data.h](#).

### 10.117.3.26 validity\_interval\_index\_clock\_fullset\_corrections

`uint8_t Galileo_HAS_data::validity_interval_index_clock_fullset_corrections`

VI - Validity Interval Index for Clock full-set corrections. See HAS SIS ICD 1.0 Section 5.2.2.1.

Definition at line 131 of file [galileo\\_has\\_data.h](#).

### 10.117.3.27 validity\_interval\_index\_clock\_subset\_corrections

`uint8_t Galileo_HAS_data::validity_interval_index_clock_subset_corrections`

VI - Validity Interval Index for Clock subset corrections. See HAS SIS ICD 1.0 Section 5.2.2.1.

Definition at line 132 of file [galileo\\_has\\_data.h](#).

### 10.117.3.28 validity\_interval\_index\_code\_bias\_corrections

`uint8_t Galileo_HAS_data::validity_interval_index_code_bias_corrections`

VI - Validity Interval Index for Code bias. See HAS SIS ICD 1.0 Section 5.2.2.1.

Definition at line 133 of file [galileo\\_has\\_data.h](#).

### 10.117.3.29 validity\_interval\_index\_orbit\_corrections

`uint8_t Galileo_HAS_data::validity_interval_index_orbit_corrections`

VI - Validity Interval Index for Orbit corrections. See HAS SIS ICD 1.0 Section 5.2.2.1.

Definition at line 130 of file [galileo\\_has\\_data.h](#).

### 10.117.3.30 validity\_interval\_index\_phase\_bias\_corrections

`uint8_t Galileo_HAS_data::validity_interval_index_phase_bias_corrections`

VI - Validity Interval Index for Phase bias. See HAS SIS ICD 1.0 Section 5.2.2.1.

Definition at line 134 of file [galileo\\_has\\_data.h](#).

The documentation for this class was generated from the following file:

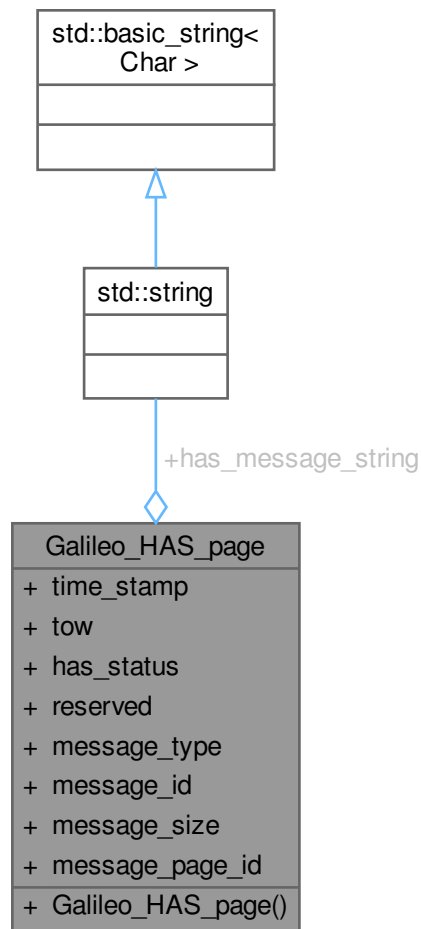
- [galileo\\_has\\_data.h](#)

## 10.118 Galileo\_HAS\_page Class Reference

This class is a storage for Galileo HAS message page, as defined in Galileo High Accuracy Service Signal-In-Space Interface Control Document (HAS SIS ICD) Issue 1.0, May 2022.

```
#include <galileo_has_page.h>
```

Collaboration diagram for Galileo\_HAS\_page:



## Public Attributes

- `std::string` [has\\_message\\_string](#)  
*HAS message content.*
- `uint64_t` [time\\_stamp](#) {}  
*HAS page time stamp, in [s].*
- `uint32_t` [tow](#) {}  
*HAS page time of week, in [s].*
- `uint8_t` [has\\_status](#) {}  
*HAS status.*
- `uint8_t` [reserved](#) {}  
*HAS reserved field.*
- `uint8_t` [message\\_type](#) {}  
*HAS message type (MT)*
- `uint8_t` [message\\_id](#) {}  
*HAS message ID (MID)*
- `uint8_t` [message\\_size](#) {}

- HAS message size (MS)*
- `uint8_t message_page_id {}`
- HAS message page ID (PID)*

### 10.118.1 Detailed Description

This class is a storage for Galileo HAS message page, as defined in Galileo High Accuracy Service Signal-In-Space Interface Control Document (HAS SIS ICD) Issue 1.0, May 2022.

Definition at line 35 of file [galileo\\_has\\_page.h](#).

### 10.118.2 Member Data Documentation

#### 10.118.2.1 has\_message\_string

`std::string Galileo_HAS_page::has_message_string`

HAS message content.

Definition at line 40 of file [galileo\\_has\\_page.h](#).

#### 10.118.2.2 has\_status

`uint8_t Galileo_HAS_page::has_status {}`

HAS status.

Definition at line 45 of file [galileo\\_has\\_page.h](#).

#### 10.118.2.3 message\_id

`uint8_t Galileo_HAS_page::message_id {}`

HAS message ID (MID)

Definition at line 48 of file [galileo\\_has\\_page.h](#).

#### 10.118.2.4 message\_page\_id

`uint8_t Galileo_HAS_page::message_page_id {}`

HAS message page ID (PID)

Definition at line 50 of file [galileo\\_has\\_page.h](#).

#### 10.118.2.5 message\_size

`uint8_t Galileo_HAS_page::message_size {}`

HAS message size (MS)

Definition at line 49 of file [galileo\\_has\\_page.h](#).

#### 10.118.2.6 message\_type

`uint8_t Galileo_HAS_page::message_type {}`

HAS message type (MT)

Definition at line 47 of file [galileo\\_has\\_page.h](#).

#### 10.118.2.7 reserved

`uint8_t Galileo_HAS_page::reserved {}`

HAS reserved field.

Definition at line 46 of file [galileo\\_has\\_page.h](#).

#### 10.118.2.8 time\_stamp

`uint64_t Galileo_HAS_page::time_stamp {}`

HAS page time stamp, in [s].

Definition at line 41 of file [galileo\\_has\\_page.h](#).

### 10.118.2.9 tow

```
uint32_t Galileo_HAS_page::tow {}
```

HAS page time of week, in [s].

Definition at line 42 of file [galileo\\_has\\_page.h](#).

The documentation for this class was generated from the following file:

- [galileo\\_has\\_page.h](#)

## 10.119 Galileo\_Inav\_Message Class Reference

This class handles the Galileo I/NAV Data message, as described in the Galileo Open Service Signal in Space Interface Control Document (OS SIS ICD), Issue 2.0 (Jan. 2021). See [https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo\\_OS\\_SIS\\_ICD\\_v2.0.pdf](https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.0.pdf).

```
#include <galileo_inav_message.h>
```

Collaboration diagram for Galileo\_Inav\_Message:

Galileo_Inav_Message
<ul style="list-style-type: none"> <li>+ Galileo_Inav_Message()</li> <li>+ ~Galileo_Inav_Message()</li> <li>+ split_page()</li> <li>+ have_new_ephemeris()</li> <li>+ have_new_iono_and_GST()</li> <li>+ have_new_utc_model()</li> <li>+ have_new_almanac()</li> <li>+ have_new_reduced_ced()</li> <li>+ have_new_ism()</li> <li>+ have_new_nma()</li> <li>and 31 more...</li> </ul>

### Public Member Functions

- void **split\_page** (std::string page\_string, int32\_t flag\_even\_word)
- bool **have\_new\_ephemeris** ()
- bool **have\_new\_iono\_and\_GST** ()
- bool **have\_new\_utc\_model** ()
- bool **have\_new\_almanac** ()
- bool **have\_new\_reduced\_ced** ()
- bool **have\_new\_ism** ()
- bool **have\_new\_nma** ()
- [Galileo\\_Ephemeris](#) **get\_ephemeris** () const
- [Galileo\\_Iono](#) **get\_iono** () const
- [Galileo\\_Utc\\_Model](#) **get\_utc\_model** () const
- [Galileo\\_Almanac\\_Helper](#) **get\_almanac** () const
- [Galileo\\_Ephemeris](#) **get\_reduced\_ced** () const

- [Galileo\\_ISM](#) `get_galileo_ism ()` const
- [OSNMA\\_msg](#) `get_osnma_msg ()`
- `std::string` `get_osnma_adkd_4_nav_bits ()`
- `void` `reset_osnma_nav_bits_adkd4 ()`
- `std::string` `get_osnma_adkd_0_12_nav_bits ()`
- `void` `reset_osnma_nav_bits_adkd0_12 ()`
- `bool` `get_flag_CRC_test ()` const
- `bool` `get_flag_TOW_set ()` const
- `void` `set_flag_TOW_set (bool flag_tow)`
- `int32_t` `get_Galileo_week ()` const
- `int32_t` `get_TOW5 ()` const
- `int32_t` `get_TOW6 ()` const
- `bool` `is_TOW5_set ()` const
- `void` `set_TOW5_flag (bool flag_tow5)`
- `bool` `is_TOW6_set ()` const
- `void` `set_TOW6_flag (bool flag_tow6)`
- `int32_t` `get_TOW0 ()` const
- `bool` `is_TOW0_set ()` const
- `void` `set_TOW0_flag (bool flag_tow0)`
- `bool` `get_flag_GGTO ()` const
- `double` `get_A0G ()` const
- `double` `get_A1G ()` const
- `double` `get_t0G ()` const
- `double` `get_WN0G ()` const
- `void` `init_PRN (uint32_t prn)`
- `void` `enable_reed_solomon ()`

### 10.119.1 Detailed Description

This class handles the Galileo I/NAV Data message, as described in the Galileo Open Service Signal in Space Interface Control Document (OS SIS ICD), Issue 2.0 (Jan. 2021). See [https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo\\_OS\\_SIS\\_ICD\\_v2.0.pdf](https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.0.pdf).

Definition at line 62 of file [galileo\\_inav\\_message.h](#).

### 10.119.2 Member Function Documentation

#### 10.119.2.1 `enable_reed_solomon()`

```
void Galileo_Inav_Message::enable_reed_solomon () [inline]
```

Definition at line 270 of file [galileo\\_inav\\_message.h](#).

#### 10.119.2.2 `get_A0G()`

```
double Galileo_Inav_Message::get_A0G () const [inline]
```

Definition at line 234 of file [galileo\\_inav\\_message.h](#).

#### 10.119.2.3 `get_A1G()`

```
double Galileo_Inav_Message::get_A1G () const [inline]
```

Definition at line 239 of file [galileo\\_inav\\_message.h](#).

#### 10.119.2.4 `get_flag_CRC_test()`

```
bool Galileo_Inav_Message::get_flag_CRC_test () const [inline]
```

Definition at line 164 of file [galileo\\_inav\\_message.h](#).

#### 10.119.2.5 `get_flag_GGTO()`

`bool Galileo_Inav_Message::get_flag_GGTO () const [inline]`  
Definition at line 229 of file [galileo\\_inav\\_message.h](#).

#### 10.119.2.6 `get_flag_TOW_set()`

`bool Galileo_Inav_Message::get_flag_TOW_set () const [inline]`  
Definition at line 169 of file [galileo\\_inav\\_message.h](#).

#### 10.119.2.7 `get_Galileo_week()`

`int32_t Galileo_Inav_Message::get_Galileo_week () const [inline]`  
Definition at line 179 of file [galileo\\_inav\\_message.h](#).

#### 10.119.2.8 `get_t0G()`

`double Galileo_Inav_Message::get_t0G () const [inline]`  
Definition at line 244 of file [galileo\\_inav\\_message.h](#).

#### 10.119.2.9 `get_TOW0()`

`int32_t Galileo_Inav_Message::get_TOW0 () const [inline]`  
Definition at line 214 of file [galileo\\_inav\\_message.h](#).

#### 10.119.2.10 `get_TOW5()`

`int32_t Galileo_Inav_Message::get_TOW5 () const [inline]`  
Definition at line 184 of file [galileo\\_inav\\_message.h](#).

#### 10.119.2.11 `get_TOW6()`

`int32_t Galileo_Inav_Message::get_TOW6 () const [inline]`  
Definition at line 189 of file [galileo\\_inav\\_message.h](#).

#### 10.119.2.12 `get_WN0G()`

`double Galileo_Inav_Message::get_WN0G () const [inline]`  
Definition at line 249 of file [galileo\\_inav\\_message.h](#).

#### 10.119.2.13 `init_PRN()`

`void Galileo_Inav_Message::init_PRN (`  
    `uint32_t prn) [inline]`  
Definition at line 257 of file [galileo\\_inav\\_message.h](#).

#### 10.119.2.14 `is_TOW0_set()`

`bool Galileo_Inav_Message::is_TOW0_set () const [inline]`  
Definition at line 219 of file [galileo\\_inav\\_message.h](#).

#### 10.119.2.15 `is_TOW5_set()`

`bool Galileo_Inav_Message::is_TOW5_set () const [inline]`  
Definition at line 194 of file [galileo\\_inav\\_message.h](#).

#### 10.119.2.16 `is_TOW6_set()`

`bool Galileo_Inav_Message::is_TOW6_set () const [inline]`  
Definition at line 204 of file [galileo\\_inav\\_message.h](#).

**10.119.2.17 set\_flag\_TOW\_set()**

```
void Galileo_Inav_Message::set_flag_TOW_set (
    bool flag_tow) [inline]
```

Definition at line 174 of file [galileo\\_inav\\_message.h](#).

**10.119.2.18 set\_TOW0\_flag()**

```
void Galileo_Inav_Message::set_TOW0_flag (
    bool flag_tow0) [inline]
```

Definition at line 224 of file [galileo\\_inav\\_message.h](#).

**10.119.2.19 set\_TOW5\_flag()**

```
void Galileo_Inav_Message::set_TOW5_flag (
    bool flag_tow5) [inline]
```

Definition at line 199 of file [galileo\\_inav\\_message.h](#).

**10.119.2.20 set\_TOW6\_flag()**

```
void Galileo_Inav_Message::set_TOW6_flag (
    bool flag_tow6) [inline]
```

Definition at line 209 of file [galileo\\_inav\\_message.h](#).

The documentation for this class was generated from the following file:

- [galileo\\_inav\\_message.h](#)

**10.120 Galileo\_Iono Class Reference**

This class is a storage for the GALILEO IONOSPHERIC data as described in Galileo ICD paragraph 5.1.6.

```
#include <galileo_iono.h>
```

Collaboration diagram for Galileo\_Iono:

Galileo_Iono
+ ai0
+ ai1
+ ai2
+ tow
+ WN
+ Region1_flag
+ Region2_flag
+ Region3_flag
+ Region4_flag
+ Region5_flag
+ Galileo_Iono()
+ serialize()

## Public Member Functions

- [Galileo\\_Iono](#) ()=default
- `template<class Archive>`  
void [serialize](#) (Archive &archive, const unsigned int version)

*Serialize is a boost standard method to be called by the boost XML serialization. Here is used to save the iono data on disk file.*

## Public Attributes

- double [ai0](#) {}  
*Effective Ionisation Level 1st order parameter [sfu].*
- double [ai1](#) {}  
*Effective Ionisation Level 2st order parameter [sfu/degree].*
- double [ai2](#) {}  
*Effective Ionisation Level 3st order parameter [sfu/degree].*
- `int32_t` [tow](#) {}  
*UTC data reference Time of Week [s].*
- `int32_t` [WN](#) {}  
*UTC data reference Week number [week].*
- bool [Region1\\_flag](#) {}  
*Ionospheric Disturbance Flag for region 1.*
- bool [Region2\\_flag](#) {}  
*Ionospheric Disturbance Flag for region 2.*
- bool [Region3\\_flag](#) {}  
*Ionospheric Disturbance Flag for region 3.*
- bool [Region4\\_flag](#) {}  
*Ionospheric Disturbance Flag for region 4.*
- bool [Region5\\_flag](#) {}  
*Ionospheric Disturbance Flag for region 5.*

### 10.120.1 Detailed Description

This class is a storage for the GALILEO IONOSPHERIC data as described in Galileo ICD paragraph 5.1.6.

See [https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo\\_OS\\_SIS\\_ICD\\_v2.0.pdf](https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.0.pdf)

Definition at line 37 of file [galileo\\_iono.h](#).

### 10.120.2 Constructor & Destructor Documentation

#### 10.120.2.1 Galileo\_Iono()

```
Galileo_Iono::Galileo_Iono () [default]
```

Default constructor

### 10.120.3 Member Function Documentation

#### 10.120.3.1 serialize()

```
template<class Archive>
void Galileo_Iono::serialize (
    Archive & archive,
    const unsigned int version) [inline]
```

Serialize is a boost standard method to be called by the boost XML serialization. Here is used to save the iono data on disk file.

Definition at line 67 of file [galileo\\_iono.h](#).

References [ai0](#), [ai1](#), [ai2](#), [Region1\\_flag](#), [Region2\\_flag](#), [Region3\\_flag](#), [Region4\\_flag](#), [Region5\\_flag](#), [tow](#), and [WN](#).



## 10.120.4 Member Data Documentation

### 10.120.4.1 ai0

```
double Galileo_Iono::ai0 {}
```

Effective Ionisation Level 1st order parameter [sfu].

Definition at line 46 of file [galileo\\_iono.h](#).

Referenced by [serialize\(\)](#).

### 10.120.4.2 ai1

```
double Galileo_Iono::ai1 {}
```

Effective Ionisation Level 2st order parameter [sfu/degree].

Definition at line 47 of file [galileo\\_iono.h](#).

Referenced by [serialize\(\)](#).

### 10.120.4.3 ai2

```
double Galileo_Iono::ai2 {}
```

Effective Ionisation Level 3st order parameter [sfu/degree].

Definition at line 48 of file [galileo\\_iono.h](#).

Referenced by [serialize\(\)](#).

### 10.120.4.4 Region1\_flag

```
bool Galileo_Iono::Region1_flag {}
```

Ionospheric Disturbance Flag for region 1.

Definition at line 55 of file [galileo\\_iono.h](#).

Referenced by [serialize\(\)](#).

### 10.120.4.5 Region2\_flag

```
bool Galileo_Iono::Region2_flag {}
```

Ionospheric Disturbance Flag for region 2.

Definition at line 56 of file [galileo\\_iono.h](#).

Referenced by [serialize\(\)](#).

### 10.120.4.6 Region3\_flag

```
bool Galileo_Iono::Region3_flag {}
```

Ionospheric Disturbance Flag for region 3.

Definition at line 57 of file [galileo\\_iono.h](#).

Referenced by [serialize\(\)](#).

### 10.120.4.7 Region4\_flag

```
bool Galileo_Iono::Region4_flag {}
```

Ionospheric Disturbance Flag for region 4.

Definition at line 58 of file [galileo\\_iono.h](#).

Referenced by [serialize\(\)](#).

### 10.120.4.8 Region5\_flag

```
bool Galileo_Iono::Region5_flag {}
```

Ionospheric Disturbance Flag for region 5.

Definition at line 59 of file [galileo\\_iono.h](#).

Referenced by [serialize\(\)](#).

**10.120.4.9 tow**

```
int32_t Galileo_Iono::tow {}
```

UTC data reference Time of Week [s].

Definition at line 51 of file [galileo\\_iono.h](#).

Referenced by [serialize\(\)](#).

**10.120.4.10 WN**

```
int32_t Galileo_Iono::WN {}
```

UTC data reference Week number [week].

Definition at line 52 of file [galileo\\_iono.h](#).

Referenced by [serialize\(\)](#).

The documentation for this class was generated from the following file:

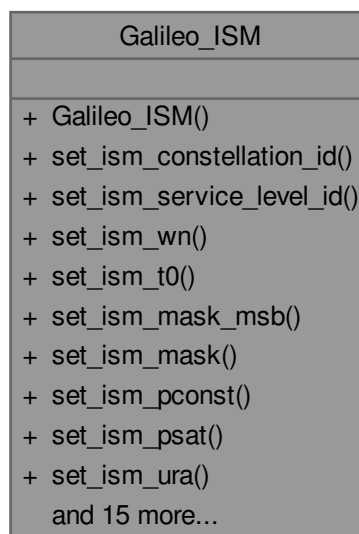
- [galileo\\_iono.h](#)

**10.121 Galileo\_ISM Class Reference**

This class is a storage for the GALILEO Integrity Support Message as described in Galileo ICD paragraph 5.2.

```
#include <galileo_ism.h>
```

Collaboration diagram for Galileo\_ISM:

**Public Member Functions**

- [Galileo\\_ISM](#) ()=default
- void **set\_ism\_constellation\_id** (uint8\_t const\_id)
- void **set\_ism\_service\_level\_id** (uint8\_t sl\_id)
- void **set\_ism\_wn** (uint16\_t wn\_ism)
- void **set\_ism\_t0** (uint16\_t t0)
- void **set\_ism\_mask\_msb** (bool mask\_msb)
- void **set\_ism\_mask** (uint32\_t mask)
- void **set\_ism\_pconst** (uint8\_t pconst)

- void **set\_ism\_psat** (uint8\_t psat)
- void **set\_ism\_ura** (uint8\_t ura)
- void **set\_ism\_ure** (uint8\_t ure)
- void **set\_ism\_bnom** (uint8\_t bnom)
- void **set\_ism\_Tvalidity** (uint8\_t tvalidity)
- bool **check\_ism\_crc** (const std::bitset< GALILEO\_DATA\_JK\_BITS > &bits)
- double **get\_pconst\_value** () const
- double **get\_psat\_value** () const
- float **get\_ura\_m** () const
- float **get\_ure\_m** () const
- float **get\_bnom\_m** () const
- uint32\_t **get\_mask\_ISM** () const
- uint16\_t **get\_WN\_ISM** () const
- uint16\_t **get\_t0\_ISM** () const
- uint16\_t **get\_Tvalidity\_hours** () const
- bool **get\_ism\_mask\_msb** () const
- bool **ism\_parameters\_apply** (uint32\_t prn) const

### 10.121.1 Detailed Description

This class is a storage for the GALILEO Integrity Support Message as described in Galileo ICD paragraph 5.2.

See [https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo\\_OS\\_SIS\\_ICD\\_v2.1.1.pdf](https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.1.1.pdf)

Definition at line 40 of file [galileo\\_ism.h](#).

### 10.121.2 Constructor & Destructor Documentation

#### 10.121.2.1 Galileo\_ISM()

```
Galileo_ISM::Galileo_ISM () [default]
```

Default constructor

The documentation for this class was generated from the following file:

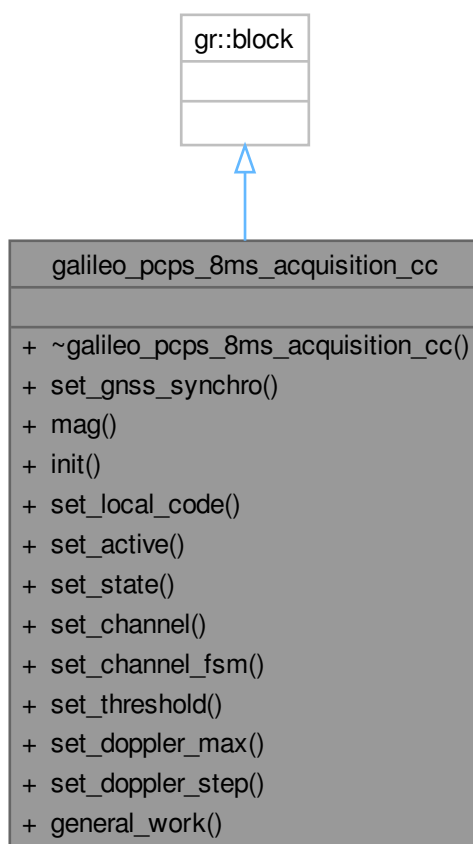
- [galileo\\_ism.h](#)

## 10.122 galileo\_pcps\_8ms\_acquisition\_cc Class Reference

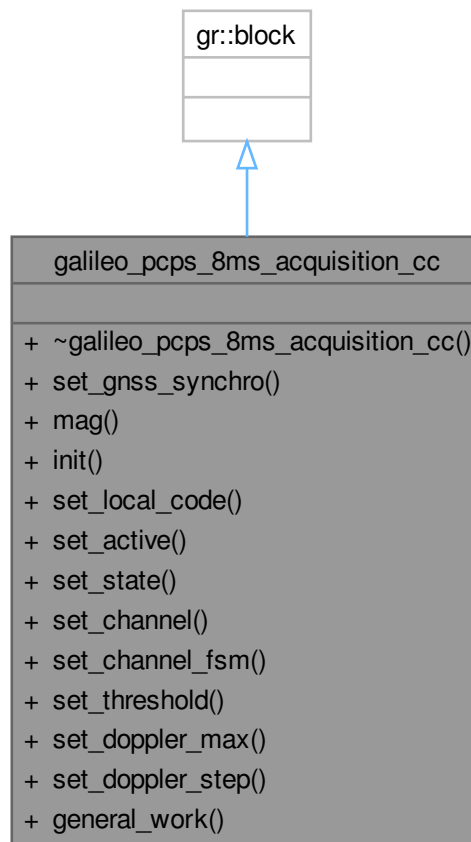
This class implements a Parallel Code Phase Search Acquisition for Galileo E1 signals with coherent integration time = 8 ms (two codes)

```
#include <galileo_pcps_8ms_acquisition_cc.h>
```

Inheritance diagram for galileo\_pcps\_8ms\_acquisition\_cc:



Collaboration diagram for galileo\_pcps\_8ms\_acquisition\_cc:



## Public Member Functions

- `~galileo_pcps_8ms_acquisition_cc ()`  
*Default destructor.*
- void `set_gnss_synchro (Gnss_Synchro *p_gnss_synchro)`  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to exchange synchronization data between acquisition and tracking blocks.*
- `uint32_t mag () const`  
*Returns the maximum peak of grid search.*
- void `init ()`  
*Initializes acquisition algorithm.*
- void `set_local_code (std::complex< float > *code)`  
*Sets local code for PCPS acquisition algorithm.*
- void `set_active (bool active)`  
*Starts acquisition algorithm, turning from standby mode to active mode.*
- void `set_state (int32_t state)`  
*If set to 1, ensures that acquisition starts at the first available sample.*
- void `set_channel (uint32_t channel)`  
*Set acquisition channel unique ID.*

- void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm)  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold)  
*Set statistics threshold of PCPS algorithm.*
- void [set\\_doppler\\_max](#) (uint32\_t doppler\_max)  
*Set maximum Doppler grid search.*
- void [set\\_doppler\\_step](#) (uint32\_t doppler\_step)  
*Set Doppler steps for the grid search.*
- int [general\\_work](#) (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)  
*Parallel Code Phase Search Acquisition signal processing.*

## Friends

- [galileo\\_pcps\\_8ms\\_acquisition\\_cc\\_sptr](#) [galileo\\_pcps\\_8ms\\_make\\_acquisition\\_cc](#) (uint32\_t sampled\_ms, uint32\_t max\_dwells, uint32\_t doppler\_max, int64\_t fs\_in, int32\_t samples\_per\_ms, int32\_t samples\_per\_ms, code, bool dump, const std::string &dump\_filename, bool enable\_monitor\_output)

## 10.122.1 Detailed Description

This class implements a Parallel Code Phase Search Acquisition for Galileo E1 signals with coherent integration time = 8 ms (two codes)

Definition at line 57 of file [galileo\\_pcps\\_8ms\\_acquisition\\_cc.h](#).

## 10.122.2 Constructor & Destructor Documentation

### 10.122.2.1 ~galileo\_pcps\_8ms\_acquisition\_cc()

```
galileo_pcps_8ms_acquisition_cc::~galileo_pcps_8ms_acquisition_cc ()
```

Default destructor.

## 10.122.3 Member Function Documentation

### 10.122.3.1 general\_work()

```
int galileo_pcps_8ms_acquisition_cc::general_work (
    int noutput_items,
    gr_vector_int & ninput_items,
    gr_vector_const_void_star & input_items,
    gr_vector_void_star & output_items)
```

Parallel Code Phase Search Acquisition signal processing.

### 10.122.3.2 init()

```
void galileo_pcps_8ms_acquisition_cc::init ()
```

Initializes acquisition algorithm.

### 10.122.3.3 mag()

```
uint32_t galileo_pcps_8ms_acquisition_cc::mag () const [inline]
```

Returns the maximum peak of grid search.

Definition at line 78 of file [galileo\\_pcps\\_8ms\\_acquisition\\_cc.h](#).

### 10.122.3.4 set\_active()

```
void galileo_pcps_8ms_acquisition_cc::set_active (
    bool active) [inline]
```

Starts acquisition algorithm, turning from standby mode to active mode.

## Parameters

<i>active</i>	- bool that activates/deactivates the block.
---------------	--

Definition at line 99 of file [galileo\\_pcps\\_8ms\\_acquisition\\_cc.h](#).

**10.122.3.5 set\_channel()**

```
void galileo_pcps_8ms_acquisition_cc::set_channel (
    uint32_t channel) [inline]
```

Set acquisition channel unique ID.

## Parameters

<i>channel</i>	- receiver channel.
----------------	---------------------

Definition at line 115 of file [galileo\\_pcps\\_8ms\\_acquisition\\_cc.h](#).

**10.122.3.6 set\_channel\_fsm()**

```
void galileo_pcps_8ms_acquisition_cc::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline]
```

Set channel fsm associated to this acquisition instance.

Definition at line 123 of file [galileo\\_pcps\\_8ms\\_acquisition\\_cc.h](#).

**10.122.3.7 set\_doppler\_max()**

```
void galileo_pcps_8ms_acquisition_cc::set_doppler_max (
    uint32_t doppler_max) [inline]
```

Set maximum Doppler grid search.

## Parameters

<i>doppler_max</i>	- Maximum Doppler shift considered in the grid search [Hz].
--------------------	---

Definition at line 142 of file [galileo\\_pcps\\_8ms\\_acquisition\\_cc.h](#).

**10.122.3.8 set\_doppler\_step()**

```
void galileo_pcps_8ms_acquisition_cc::set_doppler_step (
    uint32_t doppler_step) [inline]
```

Set Doppler steps for the grid search.

## Parameters

<i>doppler_step</i>	- Frequency bin of the search grid [Hz].
---------------------	--

Definition at line 151 of file [galileo\\_pcps\\_8ms\\_acquisition\\_cc.h](#).

**10.122.3.9 set\_gnss\_synchro()**

```
void galileo_pcps_8ms_acquisition_cc::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [inline]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to exchange synchronization data between acquisition and tracking blocks.

## Parameters

<i>p_gnss_synchro</i>	Satellite information shared by the processing blocks.
-----------------------	--

Definition at line 70 of file [galileo\\_pcps\\_8ms\\_acquisition\\_cc.h](#).

**10.122.3.10 set\_local\_code()**

```
void galileo_pcps_8ms_acquisition_cc::set_local_code (
    std::complex< float > * code)
```

Sets local code for PCPS acquisition algorithm.

**Parameters**

<i>code</i>	- Pointer to the PRN code.
-------------	----------------------------

**10.122.3.11 set\_state()**

```
void galileo_pcps_8ms_acquisition_cc::set_state (
    int32_t state)
```

If set to 1, ensures that acquisition starts at the first available sample.

**Parameters**

<i>state</i>	- int=1 forces start of acquisition
--------------	-------------------------------------

**10.122.3.12 set\_threshold()**

```
void galileo_pcps_8ms_acquisition_cc::set_threshold (
    float threshold) [inline]
```

Set statistics threshold of PCPS algorithm.

**Parameters**

<i>threshold</i>	- Threshold for signal detection (check <a href="#">Navitec2012</a> , Algorithm 1, for a definition of this threshold).
------------------	---

Definition at line 133 of file [galileo\\_pcps\\_8ms\\_acquisition\\_cc.h](#).

The documentation for this class was generated from the following file:

- [galileo\\_pcps\\_8ms\\_acquisition\\_cc.h](#)

**10.123 Galileo\_Reduced\_CED Class Reference**

This class manages the Galileo Reduced Clock and Ephemeris Data.

```
#include <galileo_reduced_ced.h>
```



Collaboration diagram for Galileo\_Reduced\_CED:

Galileo_Reduced_CED
+ PRN
+ TOTRedCED
+ IODnav
+ DeltaAred
+ exred
+ eyred
+ Deltai0red
+ Omega0red
+ lambda0red
+ af0red
+ af1red
+ Galileo_Reduced_CED()
+ compute_eph()

### Public Member Functions

- [Galileo\\_Reduced\\_CED](#) ()=default
- [Galileo\\_Ephemeris compute\\_eph](#) () const

### Public Attributes

- uint32\_t [PRN](#) {}  
*Satellite ID.*
- int32\_t [TOTRedCED](#) {}  
*Start time of transmission of the Reduced CED word in GST.*
- int32\_t [IODnav](#) {}  
*Issue of Data.*
- double [DeltaAred](#) {}  
*Difference between the Reduced CED semi-major axis and the nominal semi-major axis [meters].*
- double [exred](#) {}  
*Reduced CED eccentricity vector component x.*
- double [eyred](#) {}  
*Reduced CED eccentricity vector component y.*
- double [Deltai0red](#) {}  
*Difference between the Reduced CED inclination angle at reference time and the nominal inclination [rad].*
- double [Omega0red](#) {}  
*Reduced CED longitude of ascending node at weekly epoch [rad].*
- double [lambda0red](#) {}  
*Reduced CED mean argument of latitude [rad].*
- double [af0red](#) {}  
*Reduced CED satellite clock bias correction coefficient [seconds].*

- double [af1red](#) {}

*Reduced CED satellite clock drift correction coefficient [seconds/seconds].*

### 10.123.1 Detailed Description

This class manages the Galileo Reduced Clock and Ephemeris Data.

Definition at line 33 of file [galileo\\_reduced\\_ced.h](#).

### 10.123.2 Constructor & Destructor Documentation

#### 10.123.2.1 Galileo\_Reduced\_CED()

```
Galileo_Reduced_CED::Galileo_Reduced_CED () [default]
```

Default constructor

### 10.123.3 Member Function Documentation

#### 10.123.3.1 compute\_eph()

```
Galileo_Ephemeris Galileo_Reduced_CED::compute_eph () const
```

Convert to [Galileo\\_Ephemeris](#)

### 10.123.4 Member Data Documentation

#### 10.123.4.1 af0red

```
double Galileo_Reduced_CED::af0red {}
```

Reduced CED satellite clock bias correction coefficient [seconds].

Definition at line 55 of file [galileo\\_reduced\\_ced.h](#).

#### 10.123.4.2 af1red

```
double Galileo_Reduced_CED::af1red {}
```

Reduced CED satellite clock drift correction coefficient [seconds/seconds].

Definition at line 56 of file [galileo\\_reduced\\_ced.h](#).

#### 10.123.4.3 DeltaAred

```
double Galileo_Reduced_CED::DeltaAred {}
```

Difference between the Reduced CED semi-major axis and the nominal semi-major axis [meters].

Definition at line 49 of file [galileo\\_reduced\\_ced.h](#).

#### 10.123.4.4 Deltai0red

```
double Galileo_Reduced_CED::Deltai0red {}
```

Difference between the Reduced CED inclination angle at reference time and the nominal inclination [rad].

Definition at line 52 of file [galileo\\_reduced\\_ced.h](#).

#### 10.123.4.5 exred

```
double Galileo_Reduced_CED::exred {}
```

Reduced CED eccentricity vector component x.

Definition at line 50 of file [galileo\\_reduced\\_ced.h](#).

#### 10.123.4.6 eyred

```
double Galileo_Reduced_CED::eyred {}
```

Reduced CED eccentricity vector component y.

Definition at line 51 of file [galileo\\_reduced\\_ced.h](#).

#### 10.123.4.7 IODnav

```
int32_t Galileo_Reduced_CED::IODnav {}
```

Issue of Data.

Definition at line 48 of file [galileo\\_reduced\\_ced.h](#).

#### 10.123.4.8 lambda0red

```
double Galileo_Reduced_CED::lambda0red {}
```

Reduced CED mean argument of latitude [rad].

Definition at line 54 of file [galileo\\_reduced\\_ced.h](#).

#### 10.123.4.9 Omega0red

```
double Galileo_Reduced_CED::Omega0red {}
```

Reduced CED longitude of ascending node at weekly epoch [rad].

Definition at line 53 of file [galileo\\_reduced\\_ced.h](#).

#### 10.123.4.10 PRN

```
uint32_t Galileo_Reduced_CED::PRN {}
```

Satellite ID.

Definition at line 46 of file [galileo\\_reduced\\_ced.h](#).

#### 10.123.4.11 TOTRedCED

```
int32_t Galileo_Reduced_CED::TOTRedCED {}
```

Start time of transmission of the Reduced CED word in GST.

Definition at line 47 of file [galileo\\_reduced\\_ced.h](#).

The documentation for this class was generated from the following file:

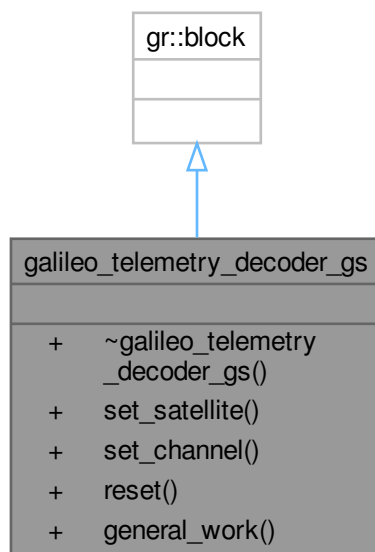
- [galileo\\_reduced\\_ced.h](#)

## 10.124 galileo\_telemetry\_decoder\_gs Class Reference

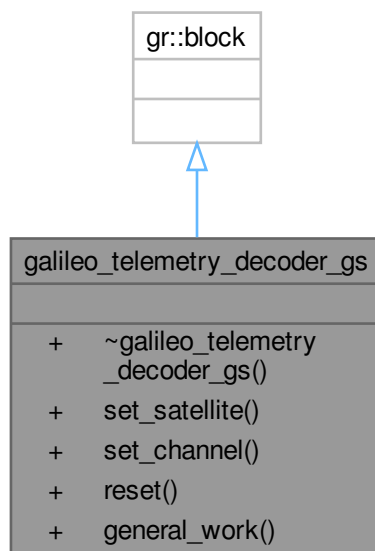
This class implements a block that decodes the INAV and FNAV data defined in Galileo ICD.

```
#include <galileo_telemetry_decoder_gs.h>
```

Inheritance diagram for `galileo_telemetry_decoder_gs`:



Collaboration diagram for `galileo_telemetry_decoder_gs`:



### Public Member Functions

- void `set_satellite` (const `Gnss_Satellite` &satellite)

*Set satellite PRN.*

- void [set\\_channel](#) (int32\_t channel)

*Set receiver's channel.*

- void [reset](#) ()
- int [general\\_work](#) (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items) override

*This is where all signal processing takes place.*

## Friends

- galileo\_telemetry\_decoder\_gs\_sptr [galileo\\_make\\_telemetry\\_decoder\\_gs](#) (const [Gnss\\_Satellite](#) &satellite, const [Tlm\\_Conf](#) &conf, int frame\_type)

## 10.124.1 Detailed Description

This class implements a block that decodes the INAV and FNAV data defined in Galileo ICD. Definition at line 61 of file [galileo\\_telemetry\\_decoder\\_gs.h](#).

## 10.124.2 Member Function Documentation

### 10.124.2.1 [general\\_work\(\)](#)

```
int galileo_telemetry_decoder_gs::general_work (
    int noutput_items,
    gr_vector_int & ninput_items,
    gr_vector_const_void_star & input_items,
    gr_vector_void_star & output_items) [override]
```

This is where all signal processing takes place.

### 10.124.2.2 [set\\_channel\(\)](#)

```
void galileo_telemetry_decoder_gs::set_channel (
    int32_t channel)
```

Set receiver's channel.

### 10.124.2.3 [set\\_satellite\(\)](#)

```
void galileo_telemetry_decoder_gs::set_satellite (
    const Gnss\_Satellite & satellite)
```

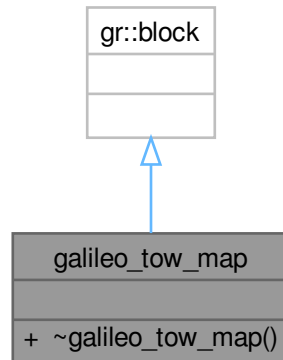
Set satellite PRN.

The documentation for this class was generated from the following file:

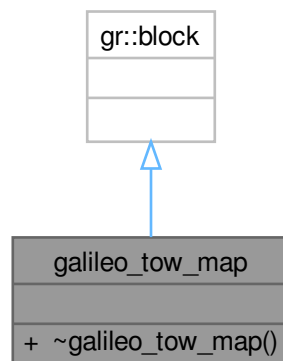
- [galileo\\_telemetry\\_decoder\\_gs.h](#)

## 10.125 galileo\_tow\_map Class Reference

Inheritance diagram for galileo\_tow\_map:



Collaboration diagram for galileo\_tow\_map:



### Public Member Functions

- [`~galileo\_tow\_map\(\)`](#)=default  
*Default destructor.*

### Friends

- `galileo_tow_map_sptr` `galileo_tow_map_make()`

### 10.125.1 Detailed Description

Definition at line 38 of file [galileo\\_tow\\_map.h](#).

## 10.125.2 Constructor & Destructor Documentation

### 10.125.2.1 ~galileo\_tow\_map()

galileo\_tow\_map::~galileo\_tow\_map () [default]

Default destructor.

The documentation for this class was generated from the following file:

- [galileo\\_tow\\_map.h](#)

## 10.126 Galileo\_Utc\_Model Class Reference

This class is a storage for the GALILEO UTC MODEL data as described in Galileo ICD [https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo\\_OS\\_SIS\\_ICD\\_v2.0.pdf](https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.0.pdf) paragraph 5.1.7.

#include <galileo\_utc\_model.h>

Collaboration diagram for Galileo\_Utc\_Model:

Galileo_Utc_Model
+ A0
+ A1
+ Delta_tLS
+ tot
+ WNot
+ WN_LSF
+ DN
+ Delta_tLSF
+ A_0G
+ A_1G
+ t_0G
+ WN_0G
+ flag_utc_model
+ Galileo_Utc_Model()
+ GST_to_UTC_time()
+ serialize()

### Public Member Functions

- [Galileo\\_Utc\\_Model](#) ()=default
- double [GST\\_to\\_UTC\\_time](#) (double t\_e, int32\_t WN) const

*GST-UTC Conversion Algorithm and Parameters.*

- template<class Archive>  
void [serialize](#) (Archive &archive, const unsigned int version)

*Serialize is a boost standard method to be called by the boost XML serialization. Here is used to save the UTC data on disk file.*

## Public Attributes

- double [A0](#) {}
- double [A1](#) {}
- int32\_t [Delta\\_tLS](#) {}
- int32\_t [tot](#) {}
  - UTC data reference Time of Week [s].*
- int32\_t [WNNot](#) {}
  - UTC data reference Week number [week].*
- int32\_t [WN\\_LSF](#) {}
- int32\_t [DN](#) {}
- int32\_t [Delta\\_tLSF](#) {}
- double [A\\_0G](#) {}
- double [A\\_1G](#) {}
- int32\_t [t\\_0G](#) {}
- int32\_t [WN\\_0G](#) {}
- bool [flag\\_utc\\_model](#) {}

### 10.126.1 Detailed Description

This class is a storage for the GALILEO UTC MODEL data as described in Galileo ICD [https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo\\_OS\\_SIS\\_ICD\\_v2.0.pdf](https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.0.pdf) paragraph 5.1.7.

Definition at line 36 of file [galileo\\_utc\\_model.h](#).

### 10.126.2 Constructor & Destructor Documentation

#### 10.126.2.1 Galileo\_Utc\_Model()

```
Galileo_Utc_Model::Galileo_Utc_Model () [default]
```

Default constructor

### 10.126.3 Member Function Documentation

#### 10.126.3.1 GST\_to\_UTC\_time()

```
double Galileo_Utc_Model::GST_to_UTC_time (
    double t_e,
    int32_t WN) const
```

GST-UTC Conversion Algorithm and Parameters.

#### 10.126.3.2 serialize()

```
template<class Archive>
void Galileo_Utc_Model::serialize (
    Archive & archive,
    const unsigned int version) [inline]
```

Serialize is a boost standard method to be called by the boost XML serialization. Here is used to save the UTC data on disk file.

Definition at line 71 of file [galileo\\_utc\\_model.h](#).

References [tot](#), and [WNNot](#).

### 10.126.4 Member Data Documentation

#### 10.126.4.1 A0

```
double Galileo_Utc_Model::A0 {}
```

Definition at line 48 of file [galileo\\_utc\\_model.h](#).



#### 10.126.4.2 A1

```
double Galileo_Utc_Model::A1 {}
```

Definition at line 49 of file [galileo\\_utc\\_model.h](#).

#### 10.126.4.3 A\_0G

```
double Galileo_Utc_Model::A_0G {}
```

Definition at line 58 of file [galileo\\_utc\\_model.h](#).

#### 10.126.4.4 A\_1G

```
double Galileo_Utc_Model::A_1G {}
```

Definition at line 59 of file [galileo\\_utc\\_model.h](#).

#### 10.126.4.5 Delta\_tLS

```
int32_t Galileo_Utc_Model::Delta_tLS {}
```

Definition at line 50 of file [galileo\\_utc\\_model.h](#).

#### 10.126.4.6 Delta\_tLSF

```
int32_t Galileo_Utc_Model::Delta_tLSF {}
```

Definition at line 55 of file [galileo\\_utc\\_model.h](#).

#### 10.126.4.7 DN

```
int32_t Galileo_Utc_Model::DN {}
```

Definition at line 54 of file [galileo\\_utc\\_model.h](#).

#### 10.126.4.8 flag\_utc\_model

```
bool Galileo_Utc_Model::flag_utc_model {}
```

Definition at line 63 of file [galileo\\_utc\\_model.h](#).

#### 10.126.4.9 t\_0G

```
int32_t Galileo_Utc_Model::t_0G {}
```

Definition at line 60 of file [galileo\\_utc\\_model.h](#).

#### 10.126.4.10 tot

```
int32_t Galileo_Utc_Model::tot {}
```

UTC data reference Time of Week [s].

Definition at line 51 of file [galileo\\_utc\\_model.h](#).

Referenced by [serialize\(\)](#).

#### 10.126.4.11 WN\_0G

```
int32_t Galileo_Utc_Model::WN_0G {}
```

Definition at line 61 of file [galileo\\_utc\\_model.h](#).

#### 10.126.4.12 WN\_LSF

```
int32_t Galileo_Utc_Model::WN_LSF {}
```

Definition at line 53 of file [galileo\\_utc\\_model.h](#).

#### 10.126.4.13 WNot

```
int32_t Galileo_Utc_Model::WNot {}
```

UTC data reference Week number [week].

Definition at line 52 of file [galileo\\_utc\\_model.h](#).

Referenced by [serialize\(\)](#).

The documentation for this class was generated from the following file:

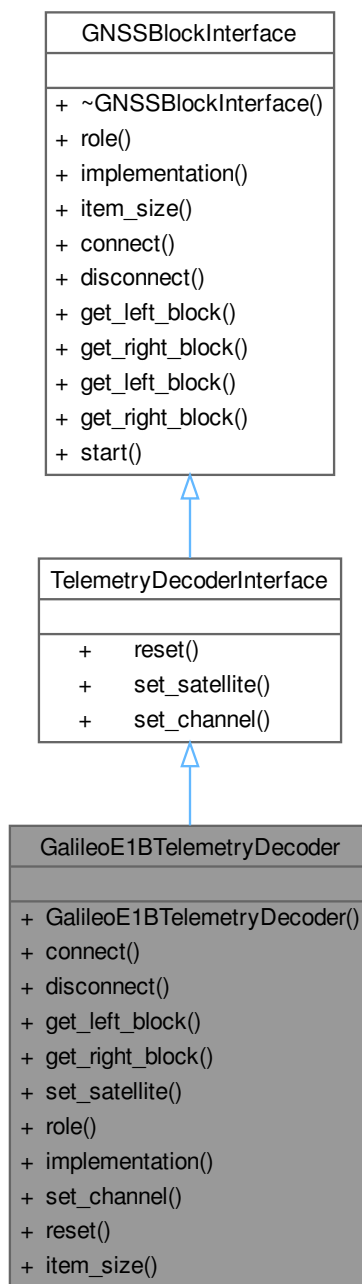
- [galileo\\_utc\\_model.h](#)

### 10.127 GalileoE1BTelemetryDecoder Class Reference

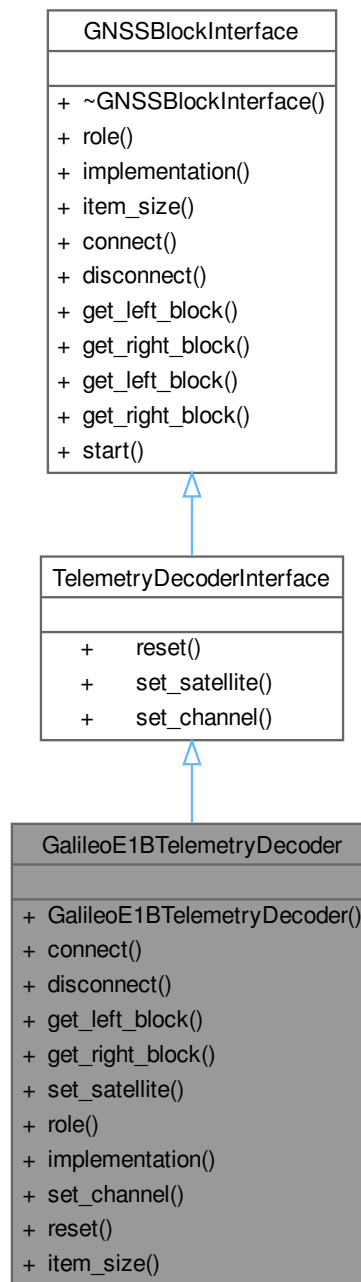
This class implements a NAV data decoder for Galileo INAV frames in E1B radio link.

```
#include <galileo_e1b_telemetry_decoder.h>
```

Inheritance diagram for GalileoE1BTelemetryDecoder:



Collaboration diagram for GalileoE1BTelemetryDecoder:



### Public Member Functions

- **GalileoE1BTelemetryDecoder** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override

- void [set\\_satellite](#) (const [Gnss\\_Satellite](#) &satellite) override
- std::string [role](#) () override
- std::string [implementation](#) () override  
*Returns "Galileo\_E1B\_Telemetry\_Decoder".*
- void [set\\_channel](#) (int channel) override
- void [reset](#) () override
- size\_t [item\\_size](#) () override

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

## 10.127.1 Detailed Description

This class implements a NAV data decoder for Galileo INAV frames in E1B radio link.  
Definition at line 44 of file [galileo\\_e1b\\_telemetry\\_decoder.h](#).

## 10.127.2 Member Function Documentation

### 10.127.2.1 connect()

```
void GalileoE1BTelemetryDecoder::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.127.2.2 disconnect()

```
void GalileoE1BTelemetryDecoder::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.127.2.3 get\_left\_block()

```
gr::basic_block_sptr GalileoE1BTelemetryDecoder::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.127.2.4 get\_right\_block()

```
gr::basic_block_sptr GalileoE1BTelemetryDecoder::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.127.2.5 implementation()

```
std::string GalileoE1BTelemetryDecoder::implementation () [inline], [override], [virtual]
```

Returns "Galileo\_E1B\_Telemetry\_Decoder".

Implements [GNSSBlockInterface](#).

Definition at line 68 of file [galileo\\_e1b\\_telemetry\\_decoder.h](#).

### 10.127.2.6 item\_size()

```
size_t GalileoE1BTelemetryDecoder::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 80 of file [galileo\\_e1b\\_telemetry\\_decoder.h](#).

#### 10.127.2.7 reset()

```
void GalileoE1BTelemetryDecoder::reset () [inline], [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

Definition at line 75 of file [galileo\\_e1b\\_telemetry\\_decoder.h](#).

#### 10.127.2.8 role()

```
std::string GalileoE1BTelemetryDecoder::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 60 of file [galileo\\_e1b\\_telemetry\\_decoder.h](#).

#### 10.127.2.9 set\_channel()

```
void GalileoE1BTelemetryDecoder::set_channel (  
    int channel) [inline], [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

Definition at line 73 of file [galileo\\_e1b\\_telemetry\\_decoder.h](#).

#### 10.127.2.10 set\_satellite()

```
void GalileoE1BTelemetryDecoder::set_satellite (  
    const Gnss_Satellite & satellite) [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

The documentation for this class was generated from the following file:

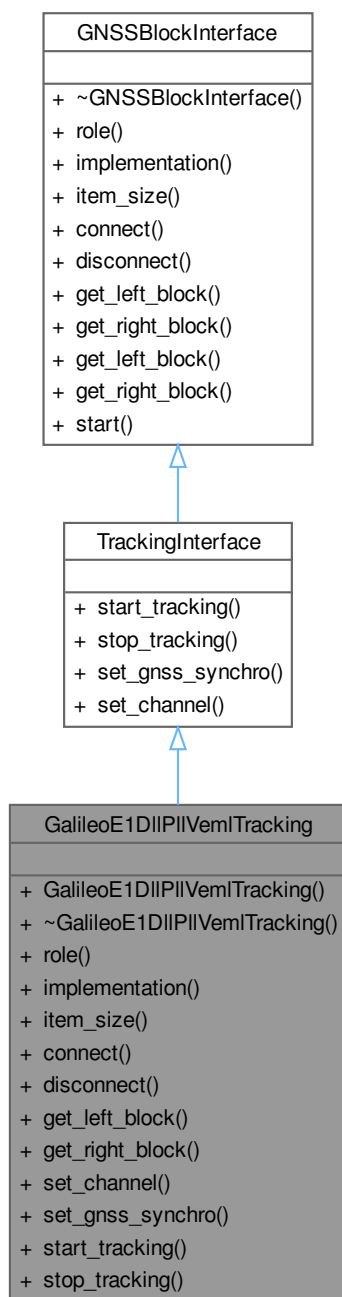
- [galileo\\_e1b\\_telemetry\\_decoder.h](#)

## 10.128 GalileoE1DIPIIVemlTracking Class Reference

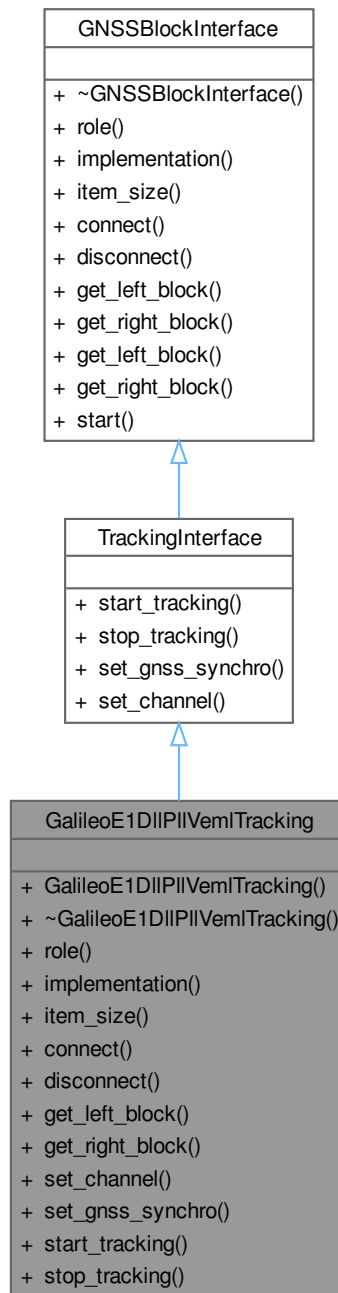
This class Adapts a DLL+PLL VEML (Very Early Minus Late) tracking loop block to a [TrackingInterface](#) for Galileo E1 signals.

```
#include <galileo_e1_dll_pll_veml_tracking.h>
```

Inheritance diagram for GalileoE1DIIPIIVemlTracking:



Collaboration diagram for GalileoE1DIIPiIVemlTracking:



## Public Member Functions

- **GalileoE1DIIPiIVemlTracking** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "Galileo\_E1\_DLL\_PLL\_VEML\_Tracking".
- size\_t [item\\_size](#) () override



- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_channel](#) (unsigned int channel) override  
*Set tracking channel unique ID.*
- void [set\\_gnss\\_synchro](#) ([Gnss\\_Synchro](#) \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [start\\_tracking](#) () override
- void [stop\\_tracking](#) () override  
*Stop running tracking.*

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()  
*Start the flow of samples if needed.*

### 10.128.1 Detailed Description

This class Adapts a DLL+PLL VEML (Very Early Minus Late) tracking loop block to a [TrackingInterface](#) for Galileo E1 signals.

Definition at line 42 of file [galileo\\_e1\\_dll\\_pll\\_veml\\_tracking.h](#).

### 10.128.2 Member Function Documentation

#### 10.128.2.1 connect()

```
void GalileoE1DllPllVemlTracking::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.128.2.2 disconnect()

```
void GalileoE1DllPllVemlTracking::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.128.2.3 get\_left\_block()

```
gr::basic_block_sptr GalileoE1DllPllVemlTracking::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.128.2.4 get\_right\_block()

```
gr::basic_block_sptr GalileoE1DllPllVemlTracking::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.128.2.5 implementation()

```
std::string GalileoE1DllPllVemlTracking::implementation () [inline], [override], [virtual]
```

Returns "Galileo\_E1\_DLL\_PLL\_VEML\_Tracking".

Implements [GNSSBlockInterface](#).

Definition at line 59 of file [galileo\\_e1\\_dll\\_pll\\_veml\\_tracking.h](#).

**10.128.2.6 item\_size()**

```
size_t GalileoE1DllPllVemlTracking::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 64 of file [galileo\\_e1\\_dll\\_pll\\_veml\\_tracking.h](#).

**10.128.2.7 role()**

```
std::string GalileoE1DllPllVemlTracking::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 53 of file [galileo\\_e1\\_dll\\_pll\\_veml\\_tracking.h](#).

**10.128.2.8 set\_channel()**

```
void GalileoE1DllPllVemlTracking::set_channel (
    unsigned int channel) [override], [virtual]
```

Set tracking channel unique ID.

Implements [TrackingInterface](#).

**10.128.2.9 set\_gnss\_synchro()**

```
void GalileoE1DllPllVemlTracking::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [TrackingInterface](#).

**10.128.2.10 start\_tracking()**

```
void GalileoE1DllPllVemlTracking::start_tracking () [override], [virtual]
```

Implements [TrackingInterface](#).

**10.128.2.11 stop\_tracking()**

```
void GalileoE1DllPllVemlTracking::stop_tracking () [override], [virtual]
```

Stop running tracking.

Implements [TrackingInterface](#).

The documentation for this class was generated from the following file:

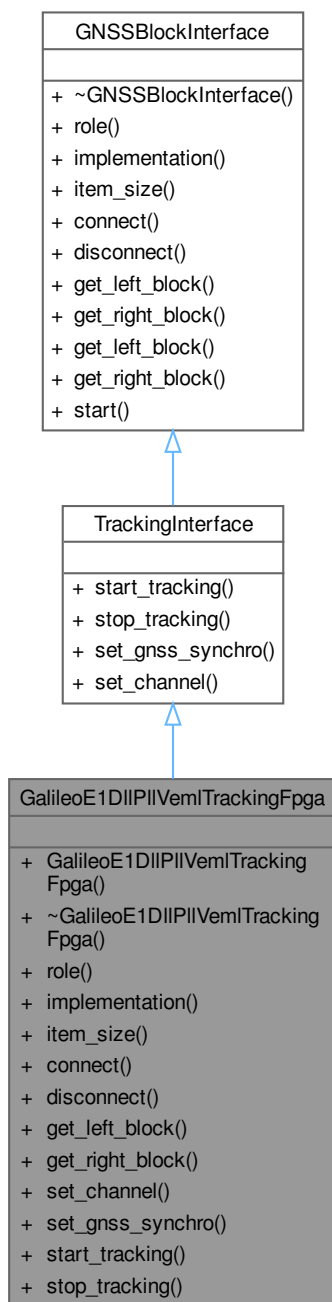
- [galileo\\_e1\\_dll\\_pll\\_veml\\_tracking.h](#)

**10.129 GalileoE1DllPllVemlTrackingFpga Class Reference**

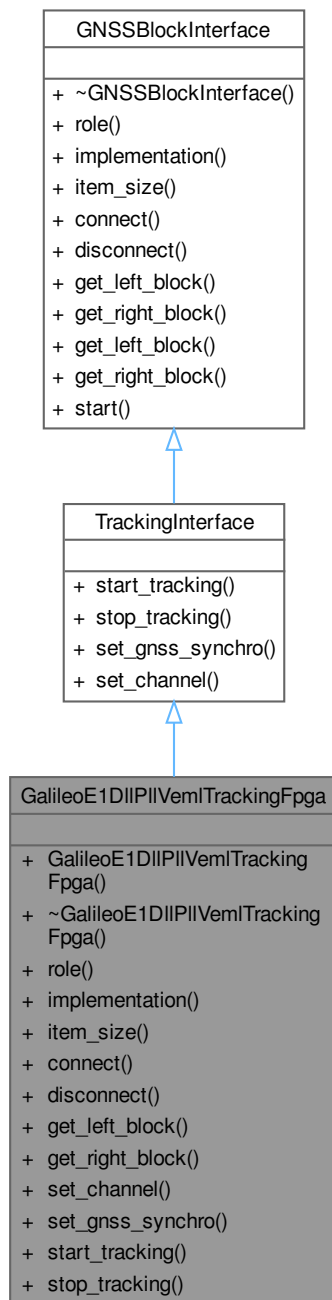
This class Adapts a DLL+PLL VEML (Very Early Minus Late) tracking loop block to a [TrackingInterface](#) for Galileo E1 signals.

```
#include <galileo_e1_dll_pll_veml_tracking_fpga.h>
```

Inheritance diagram for GalileoE1DIIPIIVemITrackingFpga:



Collaboration diagram for GalileoE1DIIPiIVemITrackingFpga:



## Public Member Functions

- [GalileoE1DIIPiIVemITrackingFpga](#) (const [ConfigurationInterface](#) \*configuration, const std::string &[role](#), unsigned int in\_streams, unsigned int out\_streams)

*Constructor.*

- virtual [~GalileoE1DIIPiIVemITrackingFpga](#) ()

*Destructor.*

- std::string [role](#) () override

- Role.*
- `std::string implementation ()` override  
*Returns "Galileo\_E1\_DLL\_PLL\_VEML\_Tracking\_FPGA".*
- `size_t item_size ()` override  
*Returns size of `lv_16sc_t`.*
- `void connect (gr::top_block_sptr top_block)` override  
*Connect.*
- `void disconnect (gr::top_block_sptr top_block)` override  
*Disconnect.*
- `gr::basic_block_sptr get_left_block ()` override  
*Get left block.*
- `gr::basic_block_sptr get_right_block ()` override  
*Get right block.*
- `void set_channel (unsigned int channel)` override  
*Set tracking channel unique ID.*
- `void set_gnss_synchro (Gnss_Synchro *p_gnss_synchro)` override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- `void start_tracking ()` override  
*Start the tracking process in the FPGA.*
- `void stop_tracking ()` override  
*Stop the tracking process in the FPGA.*

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual `gr::basic_block_sptr get_left_block (int RF_channel)`
- virtual `gr::basic_block_sptr get_right_block (int RF_channel)`
- virtual `void start ()`  
*Start the flow of samples if needed.*

## 10.129.1 Detailed Description

This class Adapts a DLL+PLL VEML (Very Early Minus Late) tracking loop block to a [TrackingInterface](#) for Galileo E1 signals.

Definition at line 42 of file [galileo\\_e1\\_dll\\_pll\\_veml\\_tracking\\_fpga.h](#).

## 10.129.2 Constructor & Destructor Documentation

### 10.129.2.1 GalileoE1DIIPIIVemlTrackingFpga()

```
GalileoE1DllPllVemlTrackingFpga::GalileoE1DllPllVemlTrackingFpga (
    const ConfigurationInterface * configuration,
    const std::string & role,
    unsigned int in_streams,
    unsigned int out_streams)
```

Constructor.

References [role\(\)](#).

Here is the call graph for this function:



### 10.129.2.2 ~GalileoE1DllPllVemlTrackingFpga()

virtual GalileoE1DllPllVemlTrackingFpga::~~GalileoE1DllPllVemlTrackingFpga () [virtual]  
Destructor.

## 10.129.3 Member Function Documentation

### 10.129.3.1 connect()

void GalileoE1DllPllVemlTrackingFpga::connect (   
 gr::top\_block\_sptr top\_block) [override], [virtual]

Connect.

Implements [GNSSBlockInterface](#).

### 10.129.3.2 disconnect()

void GalileoE1DllPllVemlTrackingFpga::disconnect (   
 gr::top\_block\_sptr top\_block) [override], [virtual]

Disconnect.

Implements [GNSSBlockInterface](#).

### 10.129.3.3 get\_left\_block()

gr::basic\_block\_sptr GalileoE1DllPllVemlTrackingFpga::get\_left\_block () [override], [virtual]

Get left block.

Implements [GNSSBlockInterface](#).

### 10.129.3.4 get\_right\_block()

gr::basic\_block\_sptr GalileoE1DllPllVemlTrackingFpga::get\_right\_block () [override], [virtual]

Get right block.

Implements [GNSSBlockInterface](#).

### 10.129.3.5 implementation()

std::string GalileoE1DllPllVemlTrackingFpga::implementation () [inline], [override], [virtual]

Returns "Galileo\_E1\_DLL\_PLL\_VEML\_Tracking\_FPGA".

Implements [GNSSBlockInterface](#).

Definition at line 70 of file [galileo\\_e1\\_dll\\_pll\\_veml\\_tracking\\_fpga.h](#).

### 10.129.3.6 item\_size()

size\_t GalileoE1DllPllVemlTrackingFpga::item\_size () [inline], [override], [virtual]

Returns size of lv\_16sc\_t.

Implements [GNSSBlockInterface](#).

Definition at line 78 of file [galileo\\_e1\\_dll\\_pll\\_veml\\_tracking\\_fpga.h](#).

### 10.129.3.7 role()

std::string GalileoE1DllPllVemlTrackingFpga::role () [inline], [override], [virtual]

Role.

Implements [GNSSBlockInterface](#).

Definition at line 62 of file [galileo\\_e1\\_dll\\_pll\\_veml\\_tracking\\_fpga.h](#).

Referenced by [GalileoE1DllPllVemlTrackingFpga\(\)](#).

### 10.129.3.8 set\_channel()

void GalileoE1DllPllVemlTrackingFpga::set\_channel (   
 unsigned int channel) [override], [virtual]

Set tracking channel unique ID.

Implements [TrackingInterface](#).

### 10.129.3.9 set\_gnss\_synchro()

```
void GalileoE1D1lP1lVemlTrackingFpga::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [TrackingInterface](#).

### 10.129.3.10 start\_tracking()

```
void GalileoE1D1lP1lVemlTrackingFpga::start_tracking () [override], [virtual]
```

Start the tracking process in the FPGA.

Implements [TrackingInterface](#).

### 10.129.3.11 stop\_tracking()

```
void GalileoE1D1lP1lVemlTrackingFpga::stop_tracking () [override], [virtual]
```

Stop the tracking process in the FPGA.

Implements [TrackingInterface](#).

The documentation for this class was generated from the following file:

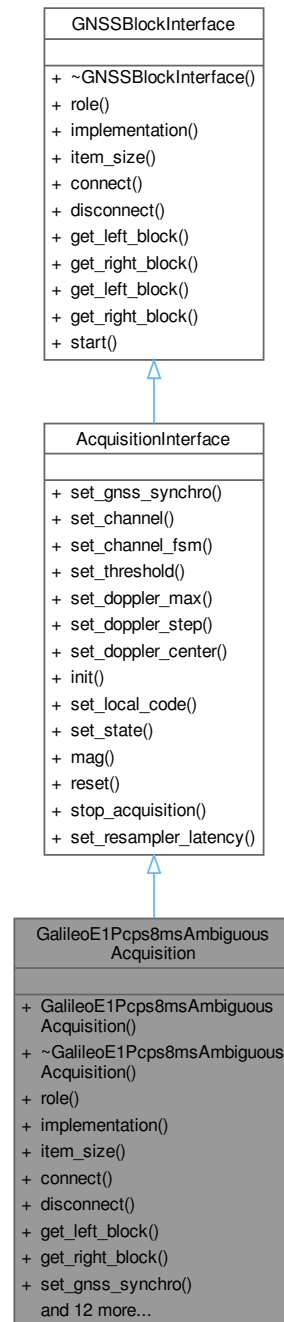
- [galileo\\_e1\\_dll\\_pll\\_veml\\_tracking\\_fpga.h](#)

## 10.130 GalileoE1Pcps8msAmbiguousAcquisition Class Reference

Adapts a PCPS 8ms acquisition block to an [AcquisitionInterface](#) for Galileo E1 Signals.

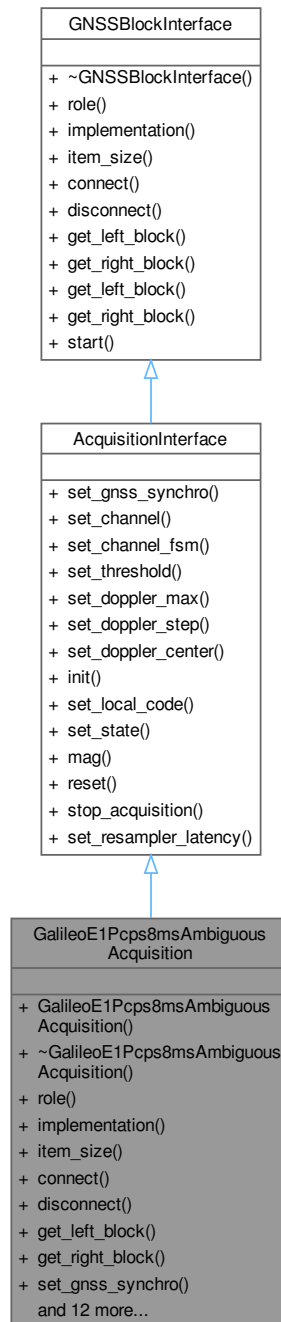
```
#include <galileo_e1_pcps_8ms_ambiguous_acquisition.h>
```

Inheritance diagram for GalileoE1Pcps8msAmbiguousAcquisition:





Collaboration diagram for GalileoE1Pcps8msAmbiguousAcquisition:



## Public Member Functions

- **GalileoE1Pcps8msAmbiguousAcquisition** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "Galileo\_E1\_PCPS\_8ms\_Ambiguous\_Acquisition".
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [set\\_channel](#) (unsigned int channel) override  
*Set acquisition channel unique ID.*
- void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm) override  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold) override  
*Set statistics threshold of PCPS algorithm.*
- void [set\\_doppler\\_max](#) (unsigned int doppler\_max) override  
*Set maximum Doppler off grid search.*
- void [set\\_doppler\\_step](#) (unsigned int doppler\_step) override  
*Set Doppler steps for the grid search.*
- void [init](#) () override  
*Initializes acquisition algorithm.*
- void [set\\_local\\_code](#) () override  
*Sets local code for Galileo E1 PCPS acquisition algorithm.*
- signed int [mag](#) () override  
*Returns the maximum peak of grid search.*
- void [reset](#) () override  
*Restart acquisition algorithm.*
- void [stop\\_acquisition](#) () override  
*Stop running acquisition.*
- void [set\\_state](#) (int state \_\_attribute\_\_((unused))) override
- void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples \_\_attribute\_\_((unused))) override

### Public Member Functions inherited from [AcquisitionInterface](#)

- virtual void [set\\_doppler\\_center](#) (int doppler\_center \_\_attribute\_\_((unused)))
- virtual void [set\\_state](#) (int state)=0
- virtual void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples)=0

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

## 10.130.1 Detailed Description

Adapts a PCPS 8ms acquisition block to an [AcquisitionInterface](#) for Galileo E1 Signals.  
Definition at line 42 of file [galileo\\_e1\\_pcps\\_8ms\\_ambiguous\\_acquisition.h](#).

## 10.130.2 Member Function Documentation

### 10.130.2.1 [connect\(\)](#)

```
void GalileoE1Pcps8msAmbiguousAcquisition::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.130.2.2 disconnect()

```
void GalileoE1Pcps8msAmbiguousAcquisition::disconnect (  
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.130.2.3 get\_left\_block()

```
gr::basic_block_sptr GalileoE1Pcps8msAmbiguousAcquisition::get_left_block () [override],  
[virtual]
```

Implements [GNSSBlockInterface](#).

### 10.130.2.4 get\_right\_block()

```
gr::basic_block_sptr GalileoE1Pcps8msAmbiguousAcquisition::get_right_block () [override],  
[virtual]
```

Implements [GNSSBlockInterface](#).

### 10.130.2.5 implementation()

```
std::string GalileoE1Pcps8msAmbiguousAcquisition::implementation () [inline], [override],  
[virtual]
```

Returns "Galileo\_E1\_PCPS\_8ms\_Ambiguous\_Acquisition".

Implements [GNSSBlockInterface](#).

Definition at line 60 of file [galileo\\_e1\\_pcps\\_8ms\\_ambiguous\\_acquisition.h](#).

### 10.130.2.6 init()

```
void GalileoE1Pcps8msAmbiguousAcquisition::init () [override], [virtual]
```

Initializes acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.130.2.7 item\_size()

```
size_t GalileoE1Pcps8msAmbiguousAcquisition::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 65 of file [galileo\\_e1\\_pcps\\_8ms\\_ambiguous\\_acquisition.h](#).

### 10.130.2.8 mag()

```
signed int GalileoE1Pcps8msAmbiguousAcquisition::mag () [override], [virtual]
```

Returns the maximum peak of grid search.

Implements [AcquisitionInterface](#).

### 10.130.2.9 reset()

```
void GalileoE1Pcps8msAmbiguousAcquisition::reset () [override], [virtual]
```

Restart acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.130.2.10 role()

```
std::string GalileoE1Pcps8msAmbiguousAcquisition::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 52 of file [galileo\\_e1\\_pcps\\_8ms\\_ambiguous\\_acquisition.h](#).

**10.130.2.11 set\_channel()**

```
void GalileoE1Pcps8msAmbiguousAcquisition::set_channel (
    unsigned int channel) [inline], [override], [virtual]
```

Set acquisition channel unique ID.

Implements [AcquisitionInterface](#).

Definition at line 85 of file [galileo\\_e1\\_pcps\\_8ms\\_ambiguous\\_acquisition.h](#).

**10.130.2.12 set\_channel\_fsm()**

```
void GalileoE1Pcps8msAmbiguousAcquisition::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline], [override], [virtual]
```

Set channel fsm associated to this acquisition instance.

Implements [AcquisitionInterface](#).

Definition at line 94 of file [galileo\\_e1\\_pcps\\_8ms\\_ambiguous\\_acquisition.h](#).

**10.130.2.13 set\_doppler\_max()**

```
void GalileoE1Pcps8msAmbiguousAcquisition::set_doppler_max (
    unsigned int doppler_max) [override], [virtual]
```

Set maximum Doppler off grid search.

Implements [AcquisitionInterface](#).

**10.130.2.14 set\_doppler\_step()**

```
void GalileoE1Pcps8msAmbiguousAcquisition::set_doppler_step (
    unsigned int doppler_step) [override], [virtual]
```

Set Doppler steps for the grid search.

Implements [AcquisitionInterface](#).

**10.130.2.15 set\_gnss\_synchro()**

```
void GalileoE1Pcps8msAmbiguousAcquisition::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [AcquisitionInterface](#).

**10.130.2.16 set\_local\_code()**

```
void GalileoE1Pcps8msAmbiguousAcquisition::set_local_code () [override], [virtual]
```

Sets local code for Galileo E1 PCPS acquisition algorithm.

Implements [AcquisitionInterface](#).

**10.130.2.17 set\_resampler\_latency()**

```
void GalileoE1Pcps8msAmbiguousAcquisition::set_resampler_latency (
    uint32_t latency_samples __attribute__((unused))) [inline], [override]
```

Definition at line 142 of file [galileo\\_e1\\_pcps\\_8ms\\_ambiguous\\_acquisition.h](#).

**10.130.2.18 set\_state()**

```
void GalileoE1Pcps8msAmbiguousAcquisition::set_state (
    int state __attribute__((unused))) [inline], [override]
```

Definition at line 140 of file [galileo\\_e1\\_pcps\\_8ms\\_ambiguous\\_acquisition.h](#).

**10.130.2.19 set\_threshold()**

```
void GalileoE1Pcps8msAmbiguousAcquisition::set_threshold (
    float threshold) [override], [virtual]
```

Set statistics threshold of PCPS algorithm.  
Implements [AcquisitionInterface](#).

#### 10.130.2.20 stop\_acquisition()

```
void GalileoE1Pcps8msAmbiguousAcquisition::stop_acquisition () [override], [virtual]
```

Stop running acquisition.

Implements [AcquisitionInterface](#).

The documentation for this class was generated from the following file:

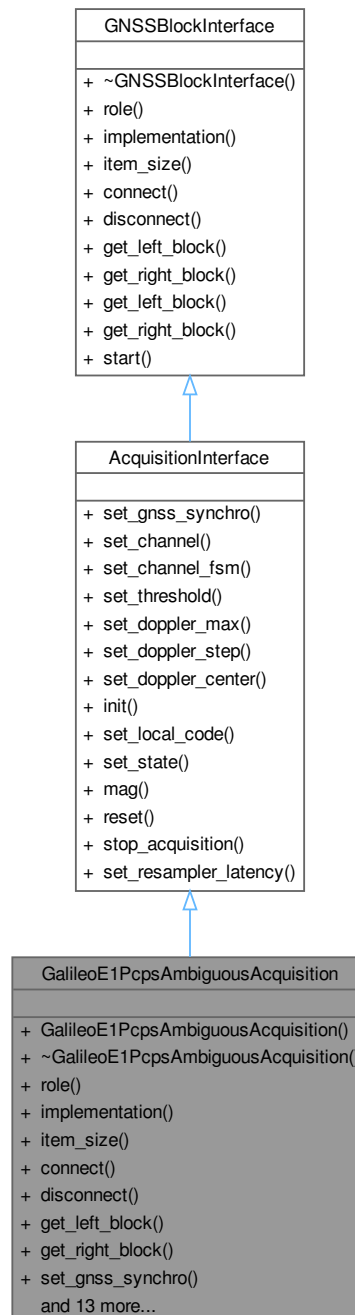
- [galileo\\_e1\\_pcps\\_8ms\\_ambiguous\\_acquisition.h](#)

## 10.131 GalileoE1PcpsAmbiguousAcquisition Class Reference

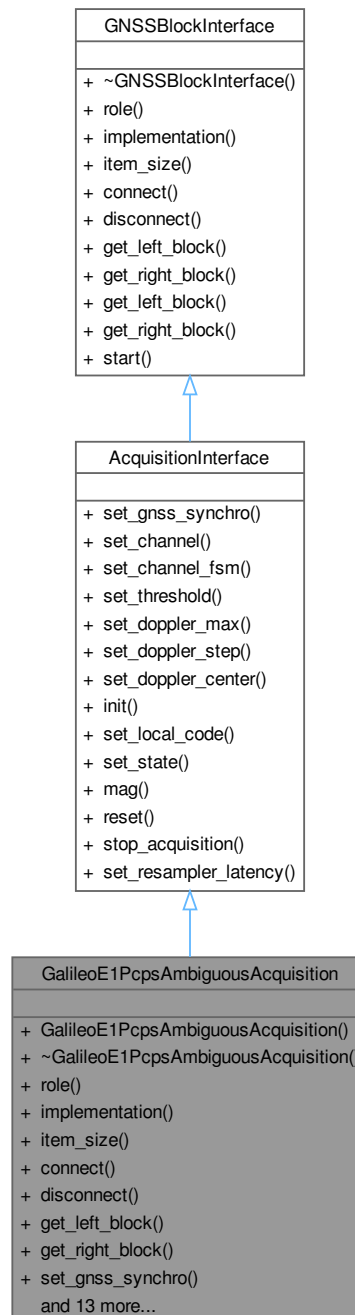
This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Galileo E1 Signals.

```
#include <galileo_e1_pcps_ambiguous_acquisition.h>
```

Inheritance diagram for GalileoE1PcpsAmbiguousAcquisition:



Collaboration diagram for GalileoE1PcpsAmbiguousAcquisition:



## Public Member Functions

- **GalileoE1PcpsAmbiguousAcquisition** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override
 

*Returns "Galileo\_E1\_PCPS\_Ambiguous\_Acquisition".*
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [set\\_channel](#) (unsigned int channel) override  
*Set acquisition channel unique ID.*
- void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm) override  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold) override  
*Set statistics threshold of PCPS algorithm.*
- void [set\\_doppler\\_max](#) (unsigned int doppler\_max) override  
*Set maximum Doppler off grid search.*
- void [set\\_doppler\\_step](#) (unsigned int doppler\_step) override  
*Set Doppler steps for the grid search.*
- void [set\\_doppler\\_center](#) (int doppler\_center) override  
*Set Doppler center for the grid search.*
- void [init](#) () override  
*Initializes acquisition algorithm.*
- void [set\\_local\\_code](#) () override  
*Sets local code for Galileo E1 PCPS acquisition algorithm.*
- signed int [mag](#) () override  
*Returns the maximum peak of grid search.*
- void [reset](#) () override  
*Restart acquisition algorithm.*
- void [set\\_state](#) (int state) override  
*If state = 1, it forces the block to start acquiring from the first sample.*
- void [stop\\_acquisition](#) () override  
*Stop running acquisition.*
- void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples) override  
*Sets the resampler latency to account it in the acquisition code delay estimation.*

### Public Member Functions inherited from [AcquisitionInterface](#)

- virtual void [set\\_doppler\\_center](#) (int doppler\_center \_\_attribute\_\_((unused)))

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

#### 10.131.1 Detailed Description

This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Galileo E1 Signals.  
Definition at line 44 of file [galileo\\_e1\\_pcps\\_ambiguous\\_acquisition.h](#).



## 10.131.2 Member Function Documentation

### 10.131.2.1 connect()

```
void GalileoE1PcpsAmbiguousAcquisition::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.131.2.2 disconnect()

```
void GalileoE1PcpsAmbiguousAcquisition::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.131.2.3 get\_left\_block()

```
gr::basic_block_sptr GalileoE1PcpsAmbiguousAcquisition::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.131.2.4 get\_right\_block()

```
gr::basic_block_sptr GalileoE1PcpsAmbiguousAcquisition::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.131.2.5 implementation()

```
std::string GalileoE1PcpsAmbiguousAcquisition::implementation () [inline], [override], [virtual]
```

Returns "Galileo\_E1\_PCPS\_Ambiguous\_Acquisition".

Implements [GNSSBlockInterface](#).

Definition at line 63 of file [galileo\\_e1\\_pcps\\_ambiguous\\_acquisition.h](#).

### 10.131.2.6 init()

```
void GalileoE1PcpsAmbiguousAcquisition::init () [override], [virtual]
```

Initializes acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.131.2.7 item\_size()

```
size_t GalileoE1PcpsAmbiguousAcquisition::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 68 of file [galileo\\_e1\\_pcps\\_ambiguous\\_acquisition.h](#).

### 10.131.2.8 mag()

```
signed int GalileoE1PcpsAmbiguousAcquisition::mag () [override], [virtual]
```

Returns the maximum peak of grid search.

Implements [AcquisitionInterface](#).

### 10.131.2.9 reset()

```
void GalileoE1PcpsAmbiguousAcquisition::reset () [override], [virtual]
```

Restart acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.131.2.10 role()

```
std::string GalileoE1PcpsAmbiguousAcquisition::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 55 of file [galileo\\_e1\\_pcps\\_ambiguous\\_acquisition.h](#).

**10.131.2.11 set\_channel()**

```
void GalileoE1PcpsAmbiguousAcquisition::set_channel (
    unsigned int channel) [inline], [override], [virtual]
```

Set acquisition channel unique ID.

Implements [AcquisitionInterface](#).

Definition at line 88 of file [galileo\\_e1\\_pcps\\_ambiguous\\_acquisition.h](#).

**10.131.2.12 set\_channel\_fsm()**

```
void GalileoE1PcpsAmbiguousAcquisition::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline], [override], [virtual]
```

Set channel fsm associated to this acquisition instance.

Implements [AcquisitionInterface](#).

Definition at line 97 of file [galileo\\_e1\\_pcps\\_ambiguous\\_acquisition.h](#).

**10.131.2.13 set\_doppler\_center()**

```
void GalileoE1PcpsAmbiguousAcquisition::set_doppler_center (
    int doppler_center) [override]
```

Set Doppler center for the grid search.

**10.131.2.14 set\_doppler\_max()**

```
void GalileoE1PcpsAmbiguousAcquisition::set_doppler_max (
    unsigned int doppler_max) [override], [virtual]
```

Set maximum Doppler off grid search.

Implements [AcquisitionInterface](#).

**10.131.2.15 set\_doppler\_step()**

```
void GalileoE1PcpsAmbiguousAcquisition::set_doppler_step (
    unsigned int doppler_step) [override], [virtual]
```

Set Doppler steps for the grid search.

Implements [AcquisitionInterface](#).

**10.131.2.16 set\_gnss\_synchro()**

```
void GalileoE1PcpsAmbiguousAcquisition::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [AcquisitionInterface](#).

**10.131.2.17 set\_local\_code()**

```
void GalileoE1PcpsAmbiguousAcquisition::set_local_code () [override], [virtual]
```

Sets local code for Galileo E1 PCPS acquisition algorithm.

Implements [AcquisitionInterface](#).

**10.131.2.18 set\_resampler\_latency()**

```
void GalileoE1PcpsAmbiguousAcquisition::set_resampler_latency (
    uint32_t latency_samples) [override], [virtual]
```

Sets the resampler latency to account it in the acquisition code delay estimation.

Implements [AcquisitionInterface](#).

**10.131.2.19 set\_state()**

```
void GalileoE1PcpsAmbiguousAcquisition::set_state (  
    int state) [override], [virtual]
```

If state = 1, it forces the block to start acquiring from the first sample.

Implements [AcquisitionInterface](#).

**10.131.2.20 set\_threshold()**

```
void GalileoE1PcpsAmbiguousAcquisition::set_threshold (  
    float threshold) [override], [virtual]
```

Set statistics threshold of PCPS algorithm.

Implements [AcquisitionInterface](#).

**10.131.2.21 stop\_acquisition()**

```
void GalileoE1PcpsAmbiguousAcquisition::stop_acquisition () [override], [virtual]
```

Stop running acquisition.

Implements [AcquisitionInterface](#).

The documentation for this class was generated from the following file:

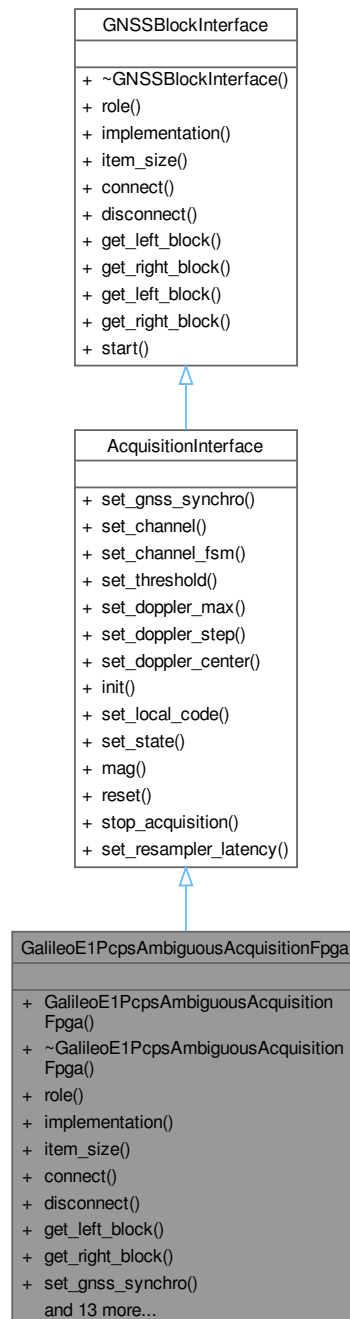
- [galileo\\_e1\\_pcps\\_ambiguous\\_acquisition.h](#)

**10.132 GalileoE1PcpsAmbiguousAcquisitionFpga Class Reference**

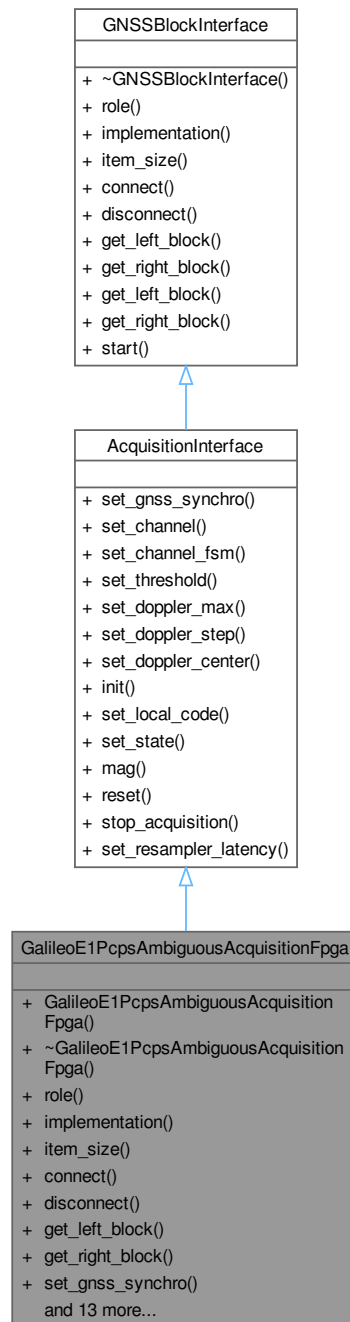
This class adapts a PCPS acquisition block off-loaded on an FPGA to an [AcquisitionInterface](#) for Galileo E1 Signals.

```
#include <galileo_e1_pcps_ambiguous_acquisition_fpga.h>
```

Inheritance diagram for GalileoE1PcpsAmbiguousAcquisitionFpga:



Collaboration diagram for GalileoE1PcpsAmbiguousAcquisitionFpga:



## Public Member Functions

- [GalileoE1PcpsAmbiguousAcquisitionFpga](#) (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)  
*Constructor.*
- [~GalileoE1PcpsAmbiguousAcquisitionFpga](#) ()=default  
*Destructor.*
- std::string [role](#) () override

- Role.*
- std::string [implementation](#) () override  
*Returns "Galileo\_E1\_PCPS\_Ambiguous\_Acquisition\_FPGA".*
- size\_t [item\\_size](#) () override  
*Returns size of lv\_16sc\_t.*
- void [connect](#) (gr::top\_block\_sptr top\_block) override  
*Connect.*
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override  
*Disconnect.*
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override  
*Get left block.*
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override  
*Get right block.*
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common Gnss\_Synchro object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [set\\_channel](#) (unsigned int channel) override  
*Set acquisition channel unique ID.*
- void [set\\_channel\\_fsm](#) (std::weak\_ptr< ChannelFsm > channel\_fsm) override  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold) override  
*Set statistics threshold of PCPS algorithm.*
- void [set\\_doppler\\_max](#) (unsigned int doppler\_max) override  
*Set maximum Doppler off grid search.*
- void [set\\_doppler\\_step](#) (unsigned int doppler\_step) override  
*Set Doppler steps for the grid search.*
- void [set\\_doppler\\_center](#) (int doppler\_center) override  
*Set Doppler center for the grid search.*
- void [init](#) () override  
*Initializes acquisition algorithm.*
- void [set\\_local\\_code](#) () override  
*Sets local code for Galileo E1 PCPS acquisition algorithm.*
- signed int [mag](#) () override  
*Returns the maximum peak of grid search.*
- void [reset](#) () override  
*Restart acquisition algorithm.*
- void [set\\_state](#) (int state) override  
*If state = 1, it forces the block to start acquiring from the first sample.*
- void [stop\\_acquisition](#) () override  
*Stop running acquisition.*
- void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples \_\_attribute\_\_((unused))) override  
*Set resampler latency.*

### Public Member Functions inherited from [AcquisitionInterface](#)

- virtual void [set\\_doppler\\_center](#) (int doppler\_center \_\_attribute\_\_((unused)))
- virtual void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples)=0

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()  
*Start the flow of samples if needed.*

### 10.132.1 Detailed Description

This class adapts a PCPS acquisition block off-loaded on an FPGA to an [AcquisitionInterface](#) for Galileo E1 Signals. Definition at line 43 of file [galileo\\_e1\\_pcps\\_ambiguous\\_acquisition\\_fpga.h](#).

### 10.132.2 Constructor & Destructor Documentation

#### 10.132.2.1 GalileoE1PcpsAmbiguousAcquisitionFpga()

```
GalileoE1PcpsAmbiguousAcquisitionFpga::GalileoE1PcpsAmbiguousAcquisitionFpga (
    const ConfigurationInterface * configuration,
    const std::string & role,
    unsigned int in_streams,
    unsigned int out_streams)
```

Constructor.

References [role\(\)](#).

Here is the call graph for this function:



#### 10.132.2.2 ~GalileoE1PcpsAmbiguousAcquisitionFpga()

```
GalileoE1PcpsAmbiguousAcquisitionFpga::~~GalileoE1PcpsAmbiguousAcquisitionFpga () [default]
```

Destructor.

### 10.132.3 Member Function Documentation

#### 10.132.3.1 connect()

```
void GalileoE1PcpsAmbiguousAcquisitionFpga::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Connect.

Implements [GNSSBlockInterface](#).

#### 10.132.3.2 disconnect()

```
void GalileoE1PcpsAmbiguousAcquisitionFpga::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Disconnect.

Implements [GNSSBlockInterface](#).

#### 10.132.3.3 get\_left\_block()

```
gr::basic_block_sptr GalileoE1PcpsAmbiguousAcquisitionFpga::get_left_block () [override], [virtual]
```

Get left block.

Implements [GNSSBlockInterface](#).

#### 10.132.3.4 get\_right\_block()

```
gr::basic_block_sptr GalileoE1PcpsAmbiguousAcquisitionFpga::get_right_block () [override], [virtual]
```

Get right block.

Implements [GNSSBlockInterface](#).

**10.132.3.5 implementation()**

```
std::string GalileoE1PcpsAmbiguousAcquisitionFpga::implementation () [inline], [override], [virtual]
```

Returns "Galileo\_E1\_PCPS\_Ambiguous\_Acquisition\_FPGA".

Implements [GNSSBlockInterface](#).

Definition at line 71 of file [galileo\\_e1\\_pcps\\_ambiguous\\_acquisition\\_fpga.h](#).

**10.132.3.6 init()**

```
void GalileoE1PcpsAmbiguousAcquisitionFpga::init () [override], [virtual]
```

Initializes acquisition algorithm.

Implements [AcquisitionInterface](#).

**10.132.3.7 item\_size()**

```
size_t GalileoE1PcpsAmbiguousAcquisitionFpga::item_size () [inline], [override], [virtual]
```

Returns size of `lv_16sc_t`.

Implements [GNSSBlockInterface](#).

Definition at line 79 of file [galileo\\_e1\\_pcps\\_ambiguous\\_acquisition\\_fpga.h](#).

**10.132.3.8 mag()**

```
signed int GalileoE1PcpsAmbiguousAcquisitionFpga::mag () [override], [virtual]
```

Returns the maximum peak of grid search.

Implements [AcquisitionInterface](#).

**10.132.3.9 reset()**

```
void GalileoE1PcpsAmbiguousAcquisitionFpga::reset () [override], [virtual]
```

Restart acquisition algorithm.

Implements [AcquisitionInterface](#).

**10.132.3.10 role()**

```
std::string GalileoE1PcpsAmbiguousAcquisitionFpga::role () [inline], [override], [virtual]
```

Role.

Implements [GNSSBlockInterface](#).

Definition at line 63 of file [galileo\\_e1\\_pcps\\_ambiguous\\_acquisition\\_fpga.h](#).

Referenced by [GalileoE1PcpsAmbiguousAcquisitionFpga\(\)](#).

**10.132.3.11 set\_channel()**

```
void GalileoE1PcpsAmbiguousAcquisitionFpga::set_channel (
    unsigned int channel) [inline], [override], [virtual]
```

Set acquisition channel unique ID.

Implements [AcquisitionInterface](#).

Definition at line 114 of file [galileo\\_e1\\_pcps\\_ambiguous\\_acquisition\\_fpga.h](#).

**10.132.3.12 set\_channel\_fsm()**

```
void GalileoE1PcpsAmbiguousAcquisitionFpga::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline], [override], [virtual]
```

Set channel fsm associated to this acquisition instance.

Implements [AcquisitionInterface](#).

Definition at line 123 of file [galileo\\_e1\\_pcps\\_ambiguous\\_acquisition\\_fpga.h](#).



**10.132.3.13 set\_doppler\_center()**

```
void GalileoE1PcpsAmbiguousAcquisitionFpga::set_doppler_center (
    int doppler_center) [override]
```

Set Doppler center for the grid search.

**10.132.3.14 set\_doppler\_max()**

```
void GalileoE1PcpsAmbiguousAcquisitionFpga::set_doppler_max (
    unsigned int doppler_max) [override], [virtual]
```

Set maximum Doppler off grid search.

Implements [AcquisitionInterface](#).

**10.132.3.15 set\_doppler\_step()**

```
void GalileoE1PcpsAmbiguousAcquisitionFpga::set_doppler_step (
    unsigned int doppler_step) [override], [virtual]
```

Set Doppler steps for the grid search.

Implements [AcquisitionInterface](#).

**10.132.3.16 set\_gnss\_synchro()**

```
void GalileoE1PcpsAmbiguousAcquisitionFpga::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [AcquisitionInterface](#).

**10.132.3.17 set\_local\_code()**

```
void GalileoE1PcpsAmbiguousAcquisitionFpga::set_local_code () [override], [virtual]
```

Sets local code for Galileo E1 PCPS acquisition algorithm.

Implements [AcquisitionInterface](#).

**10.132.3.18 set\_resampler\_latency()**

```
void GalileoE1PcpsAmbiguousAcquisitionFpga::set_resampler_latency (
    uint32_t latency_samples __attribute__((unused))) [inline], [override]
```

Set resampler latency.

Definition at line 182 of file [galileo\\_e1\\_pcps\\_ambiguous\\_acquisition\\_fpga.h](#).

**10.132.3.19 set\_state()**

```
void GalileoE1PcpsAmbiguousAcquisitionFpga::set_state (
    int state) [override], [virtual]
```

If state = 1, it forces the block to start acquiring from the first sample.

Implements [AcquisitionInterface](#).

**10.132.3.20 set\_threshold()**

```
void GalileoE1PcpsAmbiguousAcquisitionFpga::set_threshold (
    float threshold) [override], [virtual]
```

Set statistics threshold of PCPS algorithm.

Implements [AcquisitionInterface](#).

**10.132.3.21 stop\_acquisition()**

```
void GalileoE1PcpsAmbiguousAcquisitionFpga::stop_acquisition () [override], [virtual]
```

Stop running acquisition.

Implements [AcquisitionInterface](#).

The documentation for this class was generated from the following file:

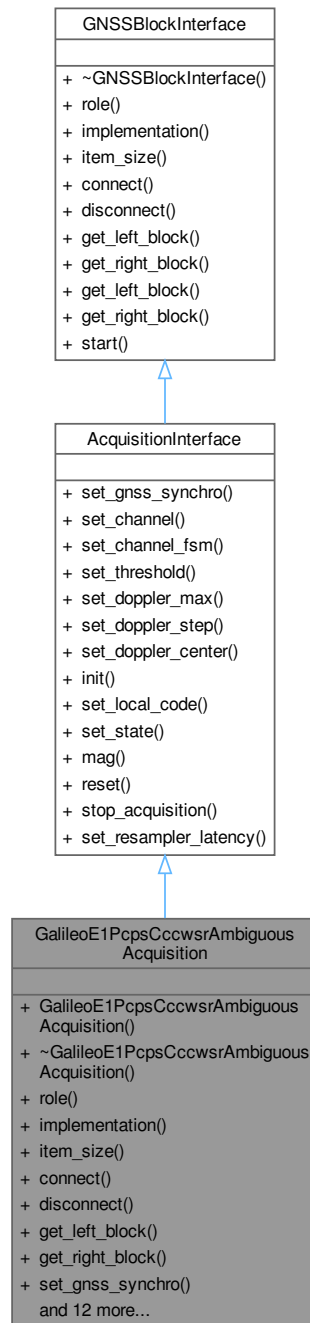
- [galileo\\_e1\\_pcps\\_ambiguous\\_acquisition\\_fpga.h](#)

## 10.133 GalileoE1PcpsCccwsrAmbiguousAcquisition Class Reference

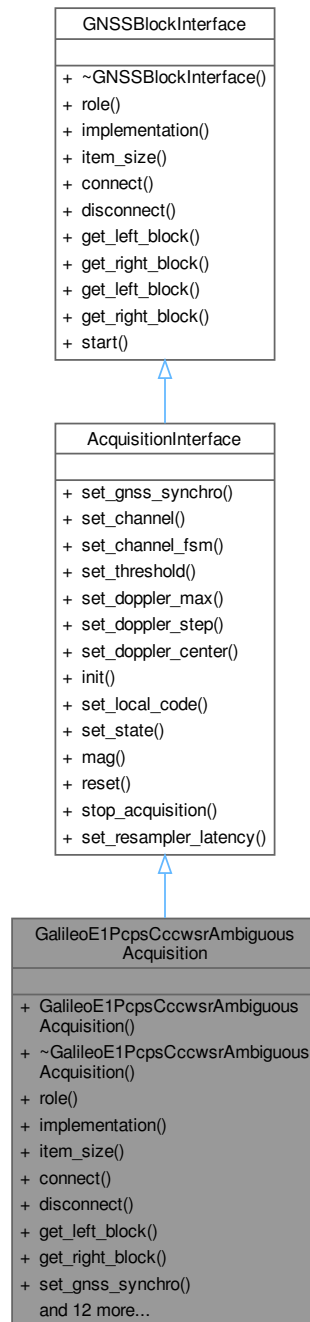
Adapts a PCPS CCCWSR acquisition block to an [AcquisitionInterface](#) for Galileo E1 Signals.

```
#include <galileo_e1_pcps_cccwsr_ambiguous_acquisition.h>
```

Inheritance diagram for GalileoE1PcpsCccwsrAmbiguousAcquisition:



Collaboration diagram for GalileoE1PcpsCccwsrAmbiguousAcquisition:



## Public Member Functions

- **GalileoE1PcpsCccwsrAmbiguousAcquisition** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override
- Returns "Galileo\_E1\_PCPS\_CCCWSR\_Ambiguous\_Acquisition".
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [set\\_channel](#) (unsigned int channel) override  
*Set acquisition channel unique ID.*
- void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm) override  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold) override  
*Set statistics threshold of CCCWSR algorithm.*
- void [set\\_doppler\\_max](#) (unsigned int doppler\_max) override  
*Set maximum Doppler off grid search.*
- void [set\\_doppler\\_step](#) (unsigned int doppler\_step) override  
*Set Doppler steps for the grid search.*
- void [init](#) () override  
*Initializes acquisition algorithm.*
- void [set\\_local\\_code](#) () override
- signed int [mag](#) () override  
*Returns the maximum peak of grid search.*
- void [reset](#) () override  
*Restart acquisition algorithm.*
- void [set\\_state](#) (int state) override  
*If state = 1, it forces the block to start acquiring from the first sample.*
- void [stop\\_acquisition](#) () override  
*Stop running acquisition.*
- void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples \_\_attribute\_\_((unused))) override

### Public Member Functions inherited from [AcquisitionInterface](#)

- virtual void [set\\_doppler\\_center](#) (int doppler\_center \_\_attribute\_\_((unused)))
- virtual void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples)=0

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

#### 10.133.1 Detailed Description

Adapts a PCPS CCCWSR acquisition block to an [AcquisitionInterface](#) for Galileo E1 Signals.  
 Definition at line 42 of file [galileo\\_e1\\_pcps\\_cccwsr\\_ambiguous\\_acquisition.h](#).

#### 10.133.2 Member Function Documentation

##### 10.133.2.1 [connect](#)()

```
void GalileoE1PcpsCccwsrAmbiguousAcquisition::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.133.2.2 disconnect()

```
void GalileoE1PcpsCccwsrAmbiguousAcquisition::disconnect (  
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.133.2.3 get\_left\_block()

```
gr::basic_block_sptr GalileoE1PcpsCccwsrAmbiguousAcquisition::get_left_block () [override],  
[virtual]
```

Implements [GNSSBlockInterface](#).

### 10.133.2.4 get\_right\_block()

```
gr::basic_block_sptr GalileoE1PcpsCccwsrAmbiguousAcquisition::get_right_block () [override],  
[virtual]
```

Implements [GNSSBlockInterface](#).

### 10.133.2.5 implementation()

```
std::string GalileoE1PcpsCccwsrAmbiguousAcquisition::implementation () [inline], [override],  
[virtual]
```

Returns "Galileo\_E1\_PCPS\_CCCWSR\_Ambiguous\_Acquisition".

Implements [GNSSBlockInterface](#).

Definition at line 61 of file [galileo\\_e1\\_pcps\\_cccwsr\\_ambiguous\\_acquisition.h](#).

### 10.133.2.6 init()

```
void GalileoE1PcpsCccwsrAmbiguousAcquisition::init () [override], [virtual]
```

Initializes acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.133.2.7 item\_size()

```
size_t GalileoE1PcpsCccwsrAmbiguousAcquisition::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 66 of file [galileo\\_e1\\_pcps\\_cccwsr\\_ambiguous\\_acquisition.h](#).

### 10.133.2.8 mag()

```
signed int GalileoE1PcpsCccwsrAmbiguousAcquisition::mag () [override], [virtual]
```

Returns the maximum peak of grid search.

Implements [AcquisitionInterface](#).

### 10.133.2.9 reset()

```
void GalileoE1PcpsCccwsrAmbiguousAcquisition::reset () [override], [virtual]
```

Restart acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.133.2.10 role()

```
std::string GalileoE1PcpsCccwsrAmbiguousAcquisition::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 53 of file [galileo\\_e1\\_pcps\\_cccwsr\\_ambiguous\\_acquisition.h](#).

**10.133.2.11 set\_channel()**

```
void GalileoElPcpsCccwsrAmbiguousAcquisition::set_channel (
    unsigned int channel) [inline], [override], [virtual]
```

Set acquisition channel unique ID.

Implements [AcquisitionInterface](#).

Definition at line 86 of file [galileo\\_e1\\_pcps\\_cccwsr\\_ambiguous\\_acquisition.h](#).

**10.133.2.12 set\_channel\_fsm()**

```
void GalileoElPcpsCccwsrAmbiguousAcquisition::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline], [override], [virtual]
```

Set channel fsm associated to this acquisition instance.

Implements [AcquisitionInterface](#).

Definition at line 95 of file [galileo\\_e1\\_pcps\\_cccwsr\\_ambiguous\\_acquisition.h](#).

**10.133.2.13 set\_doppler\_max()**

```
void GalileoElPcpsCccwsrAmbiguousAcquisition::set_doppler_max (
    unsigned int doppler_max) [override], [virtual]
```

Set maximum Doppler off grid search.

Implements [AcquisitionInterface](#).

**10.133.2.14 set\_doppler\_step()**

```
void GalileoElPcpsCccwsrAmbiguousAcquisition::set_doppler_step (
    unsigned int doppler_step) [override], [virtual]
```

Set Doppler steps for the grid search.

Implements [AcquisitionInterface](#).

**10.133.2.15 set\_gnss\_synchro()**

```
void GalileoElPcpsCccwsrAmbiguousAcquisition::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [AcquisitionInterface](#).

**10.133.2.16 set\_local\_code()**

```
void GalileoElPcpsCccwsrAmbiguousAcquisition::set_local_code () [override], [virtual]
```

Implements [AcquisitionInterface](#).

**10.133.2.17 set\_resampler\_latency()**

```
void GalileoElPcpsCccwsrAmbiguousAcquisition::set_resampler_latency (
    uint32_t latency_samples __attribute__((unused))) [inline], [override]
```

Definition at line 143 of file [galileo\\_e1\\_pcps\\_cccwsr\\_ambiguous\\_acquisition.h](#).

**10.133.2.18 set\_state()**

```
void GalileoElPcpsCccwsrAmbiguousAcquisition::set_state (
    int state) [override], [virtual]
```

If state = 1, it forces the block to start acquiring from the first sample.

Implements [AcquisitionInterface](#).

**10.133.2.19 set\_threshold()**

```
void GalileoElPcpsCccwsrAmbiguousAcquisition::set_threshold (
    float threshold) [override], [virtual]
```

Set statistics threshold of CCCWSR algorithm.  
Implements [AcquisitionInterface](#).

#### 10.133.2.20 stop\_acquisition()

```
void GalileoE1PcpsCccwsrAmbiguousAcquisition::stop_acquisition () [override], [virtual]
```

Stop running acquisition.

Implements [AcquisitionInterface](#).

The documentation for this class was generated from the following file:

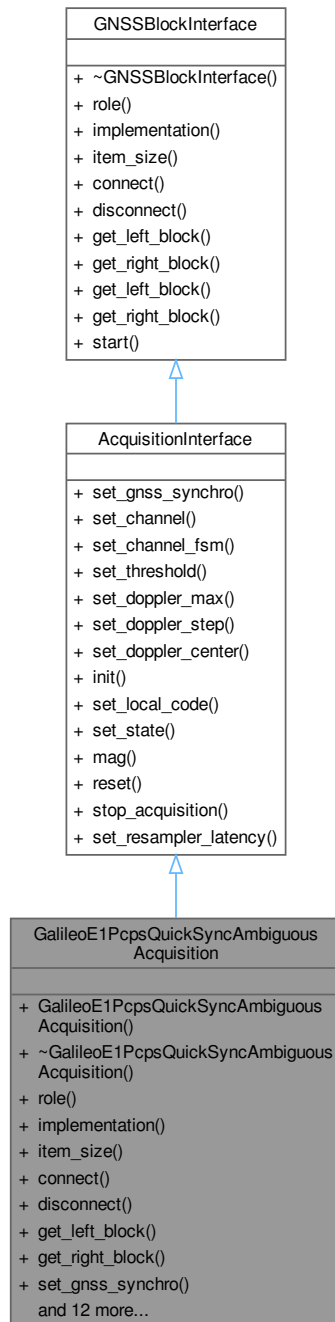
- [galileo\\_e1\\_pcps\\_cccwsr\\_ambiguous\\_acquisition.h](#)

## 10.134 GalileoE1PcpsQuickSyncAmbiguousAcquisition Class Reference

This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Galileo E1 Signals.

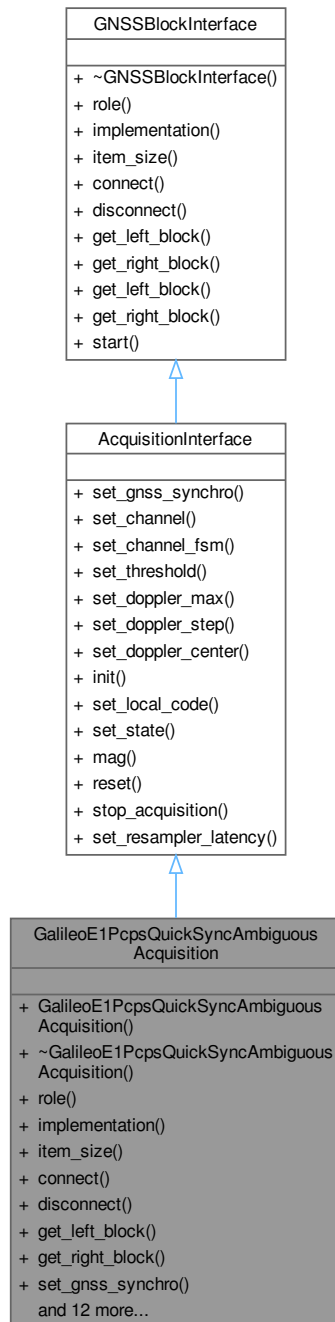
```
#include <galileo_e1_pcps_quicksync_ambiguous_acquisition.h>
```

Inheritance diagram for GalileoE1PcpsQuickSyncAmbiguousAcquisition:





Collaboration diagram for GalileoE1PcpsQuickSyncAmbiguousAcquisition:



## Public Member Functions

- **GalileoE1PcpsQuickSyncAmbiguousAcquisition** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "Galileo\_E1\_PCPS\_Ambiguous\_Acquisition".
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [set\\_channel](#) (unsigned int channel) override  
*Set acquisition channel unique ID.*
- void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm) override  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold) override  
*Set statistics threshold of PCPS algorithm.*
- void [set\\_doppler\\_max](#) (unsigned int doppler\_max) override  
*Set maximum Doppler off grid search.*
- void [set\\_doppler\\_step](#) (unsigned int doppler\_step) override  
*Set Doppler steps for the grid search.*
- void [init](#) () override  
*Initializes acquisition algorithm.*
- void [set\\_local\\_code](#) () override  
*Sets local code for Galileo E1 PCPS acquisition algorithm.*
- signed int [mag](#) () override  
*Returns the maximum peak of grid search.*
- void [reset](#) () override  
*Restart acquisition algorithm.*
- void [set\\_state](#) (int state) override  
*If state = 1, it forces the block to start acquiring from the first sample.*
- void [stop\\_acquisition](#) () override  
*Stop running acquisition.*
- void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples \_\_attribute\_\_((unused))) override

## Public Member Functions inherited from [AcquisitionInterface](#)

- virtual void [set\\_doppler\\_center](#) (int doppler\_center \_\_attribute\_\_((unused)))
- virtual void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples)=0

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.134.1 Detailed Description

This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Galileo E1 Signals.  
Definition at line 42 of file [galileo\\_e1\\_pcps\\_quicksync\\_ambiguous\\_acquisition.h](#).

### 10.134.2 Member Function Documentation

#### 10.134.2.1 [connect\(\)](#)

```
void GalileoE1PcpsQuickSyncAmbiguousAcquisition::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.134.2.2 disconnect()

```
void GalileoE1PcpsQuickSyncAmbiguousAcquisition::disconnect (  
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.134.2.3 get\_left\_block()

```
gr::basic_block_sptr GalileoE1PcpsQuickSyncAmbiguousAcquisition::get_left_block () [override],  
[virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.134.2.4 get\_right\_block()

```
gr::basic_block_sptr GalileoE1PcpsQuickSyncAmbiguousAcquisition::get_right_block () [override],  
[virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.134.2.5 implementation()

```
std::string GalileoE1PcpsQuickSyncAmbiguousAcquisition::implementation () [inline], [override],  
[virtual]
```

Returns "Galileo\_E1\_PCPS\_Ambiguous\_Acquisition".

Implements [GNSSBlockInterface](#).

Definition at line 61 of file [galileo\\_e1\\_pcps\\_quicksync\\_ambiguous\\_acquisition.h](#).

#### 10.134.2.6 init()

```
void GalileoE1PcpsQuickSyncAmbiguousAcquisition::init () [override], [virtual]
```

Initializes acquisition algorithm.

Implements [AcquisitionInterface](#).

#### 10.134.2.7 item\_size()

```
size_t GalileoE1PcpsQuickSyncAmbiguousAcquisition::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 66 of file [galileo\\_e1\\_pcps\\_quicksync\\_ambiguous\\_acquisition.h](#).

#### 10.134.2.8 mag()

```
signed int GalileoE1PcpsQuickSyncAmbiguousAcquisition::mag () [override], [virtual]
```

Returns the maximum peak of grid search.

Implements [AcquisitionInterface](#).

#### 10.134.2.9 reset()

```
void GalileoE1PcpsQuickSyncAmbiguousAcquisition::reset () [override], [virtual]
```

Restart acquisition algorithm.

Implements [AcquisitionInterface](#).

#### 10.134.2.10 role()

```
std::string GalileoE1PcpsQuickSyncAmbiguousAcquisition::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 53 of file [galileo\\_e1\\_pcps\\_quicksync\\_ambiguous\\_acquisition.h](#).

**10.134.2.11 set\_channel()**

```
void GalileoE1PcpsQuickSyncAmbiguousAcquisition::set_channel (
    unsigned int channel) [inline], [override], [virtual]
```

Set acquisition channel unique ID.

Implements [AcquisitionInterface](#).

Definition at line 86 of file [galileo\\_e1\\_pcps\\_quicksync\\_ambiguous\\_acquisition.h](#).

**10.134.2.12 set\_channel\_fsm()**

```
void GalileoE1PcpsQuickSyncAmbiguousAcquisition::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline], [override], [virtual]
```

Set channel fsm associated to this acquisition instance.

Implements [AcquisitionInterface](#).

Definition at line 95 of file [galileo\\_e1\\_pcps\\_quicksync\\_ambiguous\\_acquisition.h](#).

**10.134.2.13 set\_doppler\_max()**

```
void GalileoE1PcpsQuickSyncAmbiguousAcquisition::set_doppler_max (
    unsigned int doppler_max) [override], [virtual]
```

Set maximum Doppler off grid search.

Implements [AcquisitionInterface](#).

**10.134.2.14 set\_doppler\_step()**

```
void GalileoE1PcpsQuickSyncAmbiguousAcquisition::set_doppler_step (
    unsigned int doppler_step) [override], [virtual]
```

Set Doppler steps for the grid search.

Implements [AcquisitionInterface](#).

**10.134.2.15 set\_gnss\_synchro()**

```
void GalileoE1PcpsQuickSyncAmbiguousAcquisition::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [AcquisitionInterface](#).

**10.134.2.16 set\_local\_code()**

```
void GalileoE1PcpsQuickSyncAmbiguousAcquisition::set_local_code () [override], [virtual]
```

Sets local code for Galileo E1 PCPS acquisition algorithm.

Implements [AcquisitionInterface](#).

**10.134.2.17 set\_resampler\_latency()**

```
void GalileoE1PcpsQuickSyncAmbiguousAcquisition::set_resampler_latency (
    uint32_t latency_samples __attribute__((unused))) [inline], [override]
```

Definition at line 146 of file [galileo\\_e1\\_pcps\\_quicksync\\_ambiguous\\_acquisition.h](#).

**10.134.2.18 set\_state()**

```
void GalileoE1PcpsQuickSyncAmbiguousAcquisition::set_state (
    int state) [override], [virtual]
```

If state = 1, it forces the block to start acquiring from the first sample.

Implements [AcquisitionInterface](#).

**10.134.2.19 set\_threshold()**

```
void GalileoE1PcpsQuickSyncAmbiguousAcquisition::set_threshold (
    float threshold) [override], [virtual]
```

Set statistics threshold of PCPS algorithm.

Implements [AcquisitionInterface](#).

**10.134.2.20 stop\_acquisition()**

```
void GalileoE1PcpsQuickSyncAmbiguousAcquisition::stop_acquisition () [override], [virtual]
```

Stop running acquisition.

Implements [AcquisitionInterface](#).

The documentation for this class was generated from the following file:

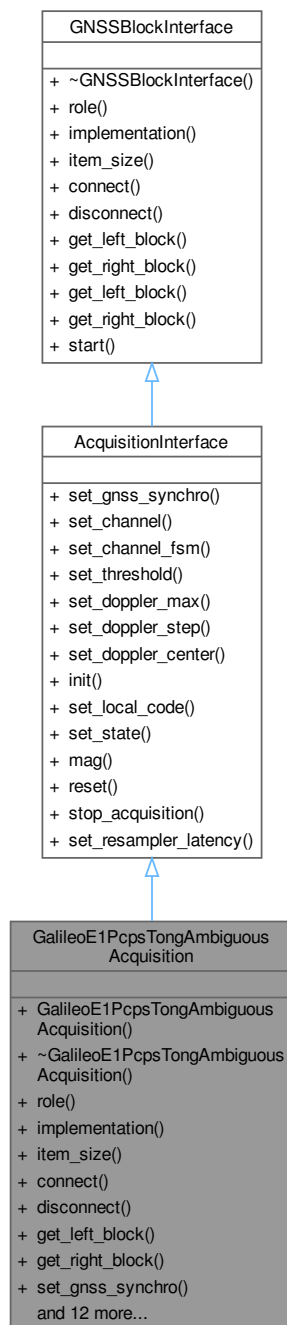
- [galileo\\_e1\\_pcps\\_quicksync\\_ambiguous\\_acquisition.h](#)

**10.135 GalileoE1PcpsTongAmbiguousAcquisition Class Reference**

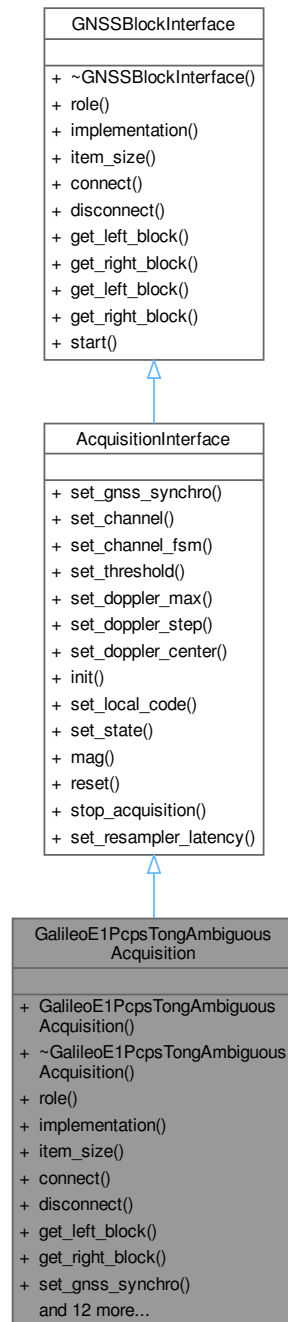
Adapts a PCPS Tong acquisition block to an [AcquisitionInterface](#) for Galileo E1 Signals.

```
#include <galileo_e1_pcps_tong_ambiguous_acquisition.h>
```

Inheritance diagram for GalileoE1PcpsTongAmbiguousAcquisition:



Collaboration diagram for GalileoE1PcpsTongAmbiguousAcquisition:



## Public Member Functions

- **GalileoE1PcpsTongAmbiguousAcquisition** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "Galileo\_E1\_PCPS\_Tong\_Ambiguous\_Acquisition".
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [set\\_channel](#) (unsigned int channel) override  
*Set acquisition channel unique ID.*
- void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm) override  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold) override  
*Set statistics threshold of TONG algorithm.*
- void [set\\_doppler\\_max](#) (unsigned int doppler\_max) override  
*Set maximum Doppler off grid search.*
- void [set\\_doppler\\_step](#) (unsigned int doppler\_step) override  
*Set Doppler steps for the grid search.*
- void [init](#) () override  
*Initializes acquisition algorithm.*
- void [set\\_local\\_code](#) () override  
*Sets local code for Galileo E1 TONG acquisition algorithm.*
- signed int [mag](#) () override  
*Returns the maximum peak of grid search.*
- void [reset](#) () override  
*Restart acquisition algorithm.*
- void [set\\_state](#) (int state) override  
*If state = 1, it forces the block to start acquiring from the first sample.*
- void [stop\\_acquisition](#) () override  
*Stop running acquisition.*
- void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples \_\_attribute\_\_((unused))) override

## Public Member Functions inherited from [AcquisitionInterface](#)

- virtual void [set\\_doppler\\_center](#) (int doppler\_center \_\_attribute\_\_((unused)))
- virtual void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples)=0

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.135.1 Detailed Description

Adapts a PCPS Tong acquisition block to an [AcquisitionInterface](#) for Galileo E1 Signals.  
Definition at line 42 of file [galileo\\_e1\\_pcps\\_tong\\_ambiguous\\_acquisition.h](#).

### 10.135.2 Member Function Documentation

#### 10.135.2.1 [connect\(\)](#)

```
void GalileoE1PcpsTongAmbiguousAcquisition::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).



### 10.135.2.2 disconnect()

```
void GalileoE1PcpsTongAmbiguousAcquisition::disconnect (  
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.135.2.3 get\_left\_block()

```
gr::basic_block_sptr GalileoE1PcpsTongAmbiguousAcquisition::get_left_block () [override],  
[virtual]
```

Implements [GNSSBlockInterface](#).

### 10.135.2.4 get\_right\_block()

```
gr::basic_block_sptr GalileoE1PcpsTongAmbiguousAcquisition::get_right_block () [override],  
[virtual]
```

Implements [GNSSBlockInterface](#).

### 10.135.2.5 implementation()

```
std::string GalileoE1PcpsTongAmbiguousAcquisition::implementation () [inline], [override],  
[virtual]
```

Returns "Galileo\_E1\_PCPS\_Tong\_Ambiguous\_Acquisition".

Implements [GNSSBlockInterface](#).

Definition at line 61 of file [galileo\\_e1\\_pcps\\_tong\\_ambiguous\\_acquisition.h](#).

### 10.135.2.6 init()

```
void GalileoE1PcpsTongAmbiguousAcquisition::init () [override], [virtual]
```

Initializes acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.135.2.7 item\_size()

```
size_t GalileoE1PcpsTongAmbiguousAcquisition::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 66 of file [galileo\\_e1\\_pcps\\_tong\\_ambiguous\\_acquisition.h](#).

### 10.135.2.8 mag()

```
signed int GalileoE1PcpsTongAmbiguousAcquisition::mag () [override], [virtual]
```

Returns the maximum peak of grid search.

Implements [AcquisitionInterface](#).

### 10.135.2.9 reset()

```
void GalileoE1PcpsTongAmbiguousAcquisition::reset () [override], [virtual]
```

Restart acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.135.2.10 role()

```
std::string GalileoE1PcpsTongAmbiguousAcquisition::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 53 of file [galileo\\_e1\\_pcps\\_tong\\_ambiguous\\_acquisition.h](#).

**10.135.2.11 set\_channel()**

```
void GalileoE1PcpsTongAmbiguousAcquisition::set_channel (
    unsigned int channel) [inline], [override], [virtual]
```

Set acquisition channel unique ID.

Implements [AcquisitionInterface](#).

Definition at line 86 of file [galileo\\_e1\\_pcps\\_tong\\_ambiguous\\_acquisition.h](#).

**10.135.2.12 set\_channel\_fsm()**

```
void GalileoE1PcpsTongAmbiguousAcquisition::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline], [override], [virtual]
```

Set channel fsm associated to this acquisition instance.

Implements [AcquisitionInterface](#).

Definition at line 95 of file [galileo\\_e1\\_pcps\\_tong\\_ambiguous\\_acquisition.h](#).

**10.135.2.13 set\_doppler\_max()**

```
void GalileoE1PcpsTongAmbiguousAcquisition::set_doppler_max (
    unsigned int doppler_max) [override], [virtual]
```

Set maximum Doppler off grid search.

Implements [AcquisitionInterface](#).

**10.135.2.14 set\_doppler\_step()**

```
void GalileoE1PcpsTongAmbiguousAcquisition::set_doppler_step (
    unsigned int doppler_step) [override], [virtual]
```

Set Doppler steps for the grid search.

Implements [AcquisitionInterface](#).

**10.135.2.15 set\_gnss\_synchro()**

```
void GalileoE1PcpsTongAmbiguousAcquisition::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [AcquisitionInterface](#).

**10.135.2.16 set\_local\_code()**

```
void GalileoE1PcpsTongAmbiguousAcquisition::set_local_code () [override], [virtual]
```

Sets local code for Galileo E1 TONG acquisition algorithm.

Implements [AcquisitionInterface](#).

**10.135.2.17 set\_resampler\_latency()**

```
void GalileoE1PcpsTongAmbiguousAcquisition::set_resampler_latency (
    uint32_t latency_samples __attribute__((unused))) [inline], [override]
```

Definition at line 146 of file [galileo\\_e1\\_pcps\\_tong\\_ambiguous\\_acquisition.h](#).

**10.135.2.18 set\_state()**

```
void GalileoE1PcpsTongAmbiguousAcquisition::set_state (
    int state) [override], [virtual]
```

If state = 1, it forces the block to start acquiring from the first sample.

Implements [AcquisitionInterface](#).

**10.135.2.19 set\_threshold()**

```
void GalileoE1PcpsTongAmbiguousAcquisition::set_threshold (
    float threshold) [override], [virtual]
```

Set statistics threshold of TONG algorithm.

Implements [AcquisitionInterface](#).

**10.135.2.20 stop\_acquisition()**

```
void GalileoE1PcpsTongAmbiguousAcquisition::stop_acquisition () [override], [virtual]
```

Stop running acquisition.

Implements [AcquisitionInterface](#).

The documentation for this class was generated from the following file:

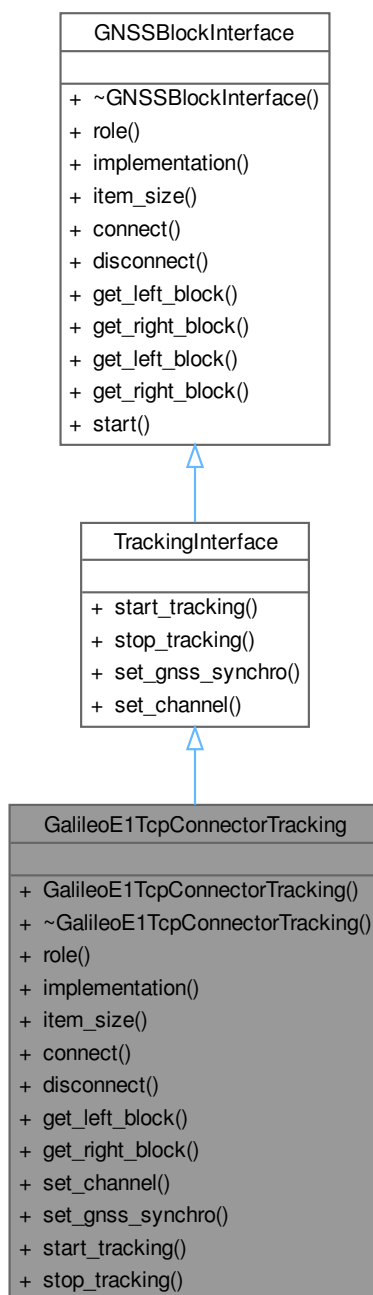
- [galileo\\_e1\\_pcps\\_tong\\_ambiguous\\_acquisition.h](#)

## 10.136 GalileoE1TcpConnectorTracking Class Reference

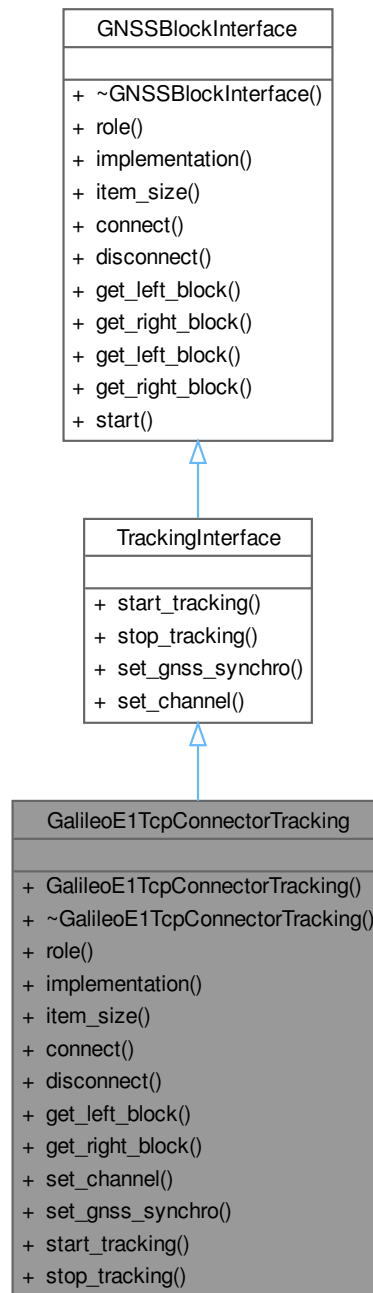
This class implements a code DLL + carrier PLL tracking loop.

```
#include <galileo_e1_tcp_connector_tracking.h>
```

Inheritance diagram for GalileoE1TcpConnectorTracking:



Collaboration diagram for GalileoE1TcpConnectorTracking:



### Public Member Functions

- **GalileoE1TcpConnectorTracking** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "Galileo\_E1\_TCP\_CONNECTOR\_Tracking".
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_channel](#) (unsigned int channel) override
 

*Set tracking channel unique ID.*
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override
 

*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [start\\_tracking](#) () override
- void [stop\\_tracking](#) () override
 

*Stop running tracking.*

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()
 

*Start the flow of samples if needed.*

### 10.136.1 Detailed Description

This class implements a code DLL + carrier PLL tracking loop.  
 Definition at line 43 of file [galileo\\_e1\\_tcp\\_connector\\_tracking.h](#).

### 10.136.2 Member Function Documentation

#### 10.136.2.1 connect()

```
void GalileoE1TcpConnectorTracking::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.136.2.2 disconnect()

```
void GalileoE1TcpConnectorTracking::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.136.2.3 get\_left\_block()

```
gr::basic_block_sptr GalileoE1TcpConnectorTracking::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.136.2.4 get\_right\_block()

```
gr::basic_block_sptr GalileoE1TcpConnectorTracking::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.136.2.5 implementation()

```
std::string GalileoE1TcpConnectorTracking::implementation () [inline], [override], [virtual]
```

Returns "Galileo\_E1\_TCP\_CONNECTOR\_Tracking".  
 Implements [GNSSBlockInterface](#).  
 Definition at line 60 of file [galileo\\_e1\\_tcp\\_connector\\_tracking.h](#).

**10.136.2.6 item\_size()**

```
size_t GalileoE1TcpConnectorTracking::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 65 of file [galileo\\_e1\\_tcp\\_connector\\_tracking.h](#).

**10.136.2.7 role()**

```
std::string GalileoE1TcpConnectorTracking::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 54 of file [galileo\\_e1\\_tcp\\_connector\\_tracking.h](#).

**10.136.2.8 set\_channel()**

```
void GalileoE1TcpConnectorTracking::set_channel (
    unsigned int channel) [override], [virtual]
```

Set tracking channel unique ID.

Implements [TrackingInterface](#).

**10.136.2.9 set\_gnss\_synchro()**

```
void GalileoE1TcpConnectorTracking::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [TrackingInterface](#).

**10.136.2.10 start\_tracking()**

```
void GalileoE1TcpConnectorTracking::start_tracking () [override], [virtual]
```

Implements [TrackingInterface](#).

**10.136.2.11 stop\_tracking()**

```
void GalileoE1TcpConnectorTracking::stop_tracking () [override], [virtual]
```

Stop running tracking.

Implements [TrackingInterface](#).

The documentation for this class was generated from the following file:

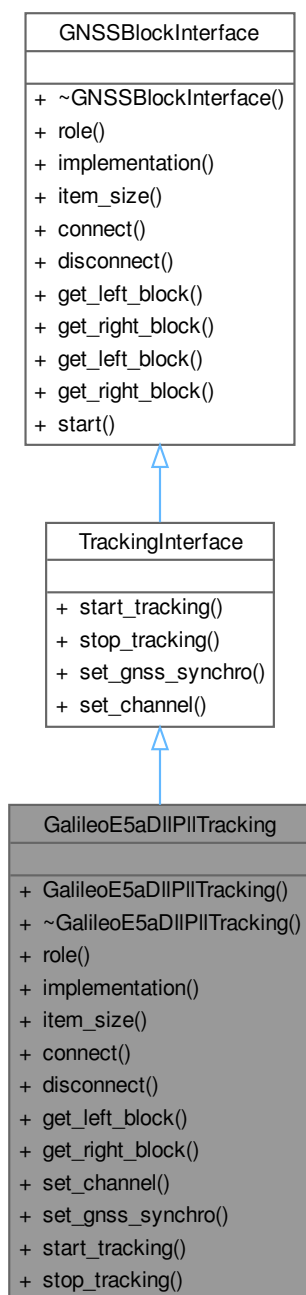
- [galileo\\_e1\\_tcp\\_connector\\_tracking.h](#)

**10.137 GalileoE5aDIPIITracking Class Reference**

This class implements a code DLL + carrier PLL tracking loop.

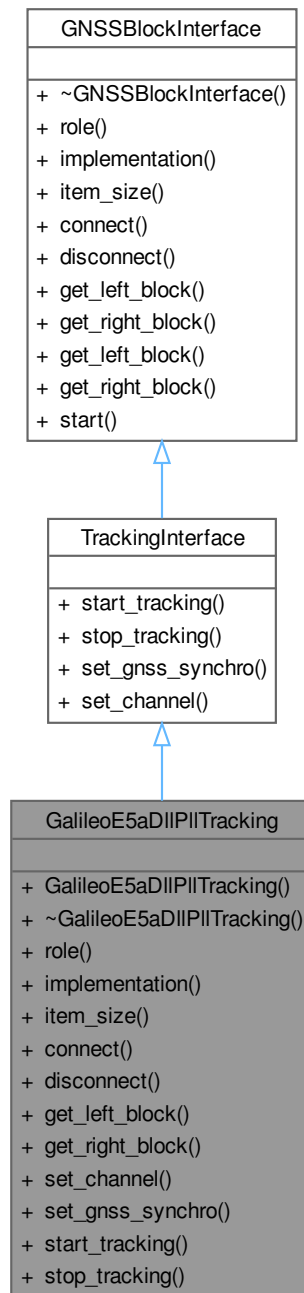
```
#include <galileo_e5a_dll_pll_tracking.h>
```

Inheritance diagram for GalileoE5aDIIPIITracking:





Collaboration diagram for GalileoE5aDIIPIITracking:



### Public Member Functions

- **GalileoE5aDIIPIITracking** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "Galileo\_E5a\_DLL\_PLL\_Tracking".
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_channel](#) (unsigned int channel) override  
*Set tracking channel unique ID.*
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [start\\_tracking](#) () override
- void [stop\\_tracking](#) () override  
*Stop running tracking.*

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()  
*Start the flow of samples if needed.*

### 10.137.1 Detailed Description

This class implements a code DLL + carrier PLL tracking loop.  
Definition at line 41 of file [galileo\\_e5a\\_dll\\_pll\\_tracking.h](#).

### 10.137.2 Member Function Documentation

#### 10.137.2.1 connect()

```
void GalileoE5aDllPllTracking::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.137.2.2 disconnect()

```
void GalileoE5aDllPllTracking::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.137.2.3 get\_left\_block()

```
gr::basic_block_sptr GalileoE5aDllPllTracking::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.137.2.4 get\_right\_block()

```
gr::basic_block_sptr GalileoE5aDllPllTracking::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.137.2.5 implementation()

```
std::string GalileoE5aDllPllTracking::implementation () [inline], [override], [virtual]
```

Returns "Galileo\_E5a\_DLL\_PLL\_Tracking".  
Implements [GNSSBlockInterface](#).  
Definition at line 58 of file [galileo\\_e5a\\_dll\\_pll\\_tracking.h](#).

**10.137.2.6 item\_size()**

```
size_t GalileoE5aDlIPllTracking::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 63 of file [galileo\\_e5a\\_dll\\_pll\\_tracking.h](#).

**10.137.2.7 role()**

```
std::string GalileoE5aDlIPllTracking::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 52 of file [galileo\\_e5a\\_dll\\_pll\\_tracking.h](#).

**10.137.2.8 set\_channel()**

```
void GalileoE5aDlIPllTracking::set_channel (
    unsigned int channel) [override], [virtual]
```

Set tracking channel unique ID.

Implements [TrackingInterface](#).

**10.137.2.9 set\_gnss\_synchro()**

```
void GalileoE5aDlIPllTracking::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [TrackingInterface](#).

**10.137.2.10 start\_tracking()**

```
void GalileoE5aDlIPllTracking::start_tracking () [override], [virtual]
```

Implements [TrackingInterface](#).

**10.137.2.11 stop\_tracking()**

```
void GalileoE5aDlIPllTracking::stop_tracking () [override], [virtual]
```

Stop running tracking.

Implements [TrackingInterface](#).

The documentation for this class was generated from the following file:

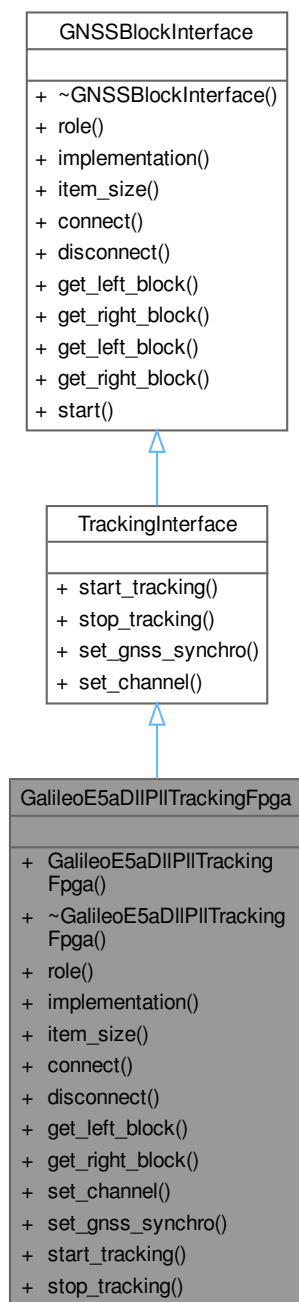
- [galileo\\_e5a\\_dll\\_pll\\_tracking.h](#)

**10.138 GalileoE5aDlIPllTrackingFpga Class Reference**

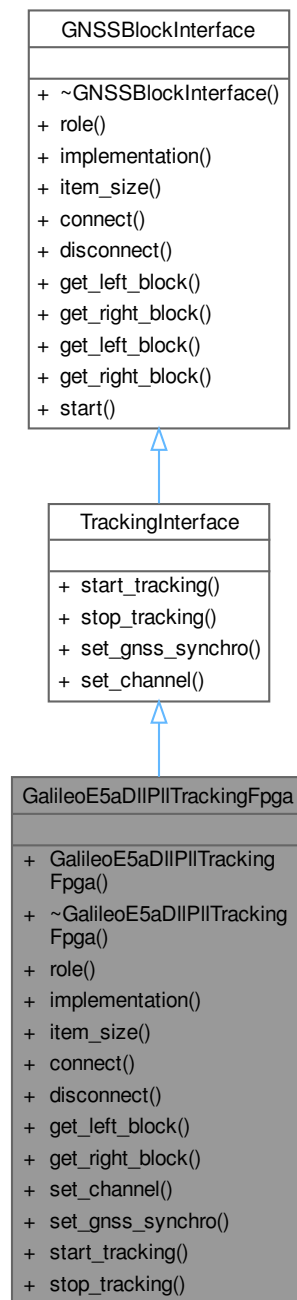
This class implements a code DLL + carrier PLL tracking loop.

```
#include <galileo_e5a_dll_pll_tracking_fpga.h>
```

Inheritance diagram for GalileoE5aDIIPIITrackingFpga:



Collaboration diagram for GalileoE5aDIIPIITrackingFpga:



## Public Member Functions

- [GalileoE5aDIIPIITrackingFpga](#) (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)  
*Constructor.*
- virtual [~GalileoE5aDIIPIITrackingFpga](#) ()  
*Destructor.*
- std::string [role](#) () override

*Role.*

- `std::string implementation ()` override  
*Returns "Galileo\_E5a\_DLL\_PLL\_Tracking\_FPGA".*
- `size_t item_size ()` override  
*Returns size of `lv_16sc_t`.*
- `void connect (gr::top_block_sptr top_block)` override  
*Connect.*
- `void disconnect (gr::top_block_sptr top_block)` override  
*Disconnect.*
- `gr::basic_block_sptr get_left_block ()` override  
*Get left block.*
- `gr::basic_block_sptr get_right_block ()` override  
*Get right block.*
- `void set_channel (unsigned int channel)` override  
*Set tracking channel unique ID.*
- `void set_gnss_synchro (Gnss_Synchro *p_gnss_synchro)` override  
*Set acquisition/tracking common `Gnss_Synchro` object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- `void start_tracking ()` override  
*Start the tracking process in the FPGA.*
- `void stop_tracking ()` override  
*Stop the tracking process in the FPGA.*

## Public Member Functions inherited from GNSSBlockInterface

- virtual `gr::basic_block_sptr get_left_block (int RF_channel)`
- virtual `gr::basic_block_sptr get_right_block (int RF_channel)`
- virtual `void start ()`  
*Start the flow of samples if needed.*

### 10.138.1 Detailed Description

This class implements a code DLL + carrier PLL tracking loop.  
Definition at line 36 of file [galileo\\_e5a\\_dll\\_pll\\_tracking\\_fpga.h](#).

### 10.138.2 Constructor & Destructor Documentation

#### 10.138.2.1 GalileoE5aDllPllTrackingFpga()

```
GalileoE5aDllPllTrackingFpga::GalileoE5aDllPllTrackingFpga (
    const ConfigurationInterface * configuration,
    const std::string & role,
    unsigned int in_streams,
    unsigned int out_streams)
```

Constructor.

References [role\(\)](#).

Here is the call graph for this function:



**10.138.2.2 ~GalileoE5aDlIPllTrackingFpga()**

virtual GalileoE5aDlIPllTrackingFpga::~~GalileoE5aDlIPllTrackingFpga () [virtual]  
Destructor.

**10.138.3 Member Function Documentation****10.138.3.1 connect()**

void GalileoE5aDlIPllTrackingFpga::connect (   
gr::top\_block\_sptr top\_block) [override], [virtual]  
Connect.  
Implements [GNSSBlockInterface](#).

**10.138.3.2 disconnect()**

void GalileoE5aDlIPllTrackingFpga::disconnect (   
gr::top\_block\_sptr top\_block) [override], [virtual]  
Disconnect.  
Implements [GNSSBlockInterface](#).

**10.138.3.3 get\_left\_block()**

gr::basic\_block\_sptr GalileoE5aDlIPllTrackingFpga::get\_left\_block () [override], [virtual]  
Get left block.  
Implements [GNSSBlockInterface](#).

**10.138.3.4 get\_right\_block()**

gr::basic\_block\_sptr GalileoE5aDlIPllTrackingFpga::get\_right\_block () [override], [virtual]  
Get right block.  
Implements [GNSSBlockInterface](#).

**10.138.3.5 implementation()**

std::string GalileoE5aDlIPllTrackingFpga::implementation () [inline], [override], [virtual]  
Returns "Galileo\_E5a\_DLL\_PLL\_Tracking\_FPGA".  
Implements [GNSSBlockInterface](#).  
Definition at line 64 of file [galileo\\_e5a\\_dll\\_pll\\_tracking\\_fpga.h](#).

**10.138.3.6 item\_size()**

size\_t GalileoE5aDlIPllTrackingFpga::item\_size () [inline], [override], [virtual]  
Returns size of lv\_16sc\_t.  
Implements [GNSSBlockInterface](#).  
Definition at line 72 of file [galileo\\_e5a\\_dll\\_pll\\_tracking\\_fpga.h](#).

**10.138.3.7 role()**

std::string GalileoE5aDlIPllTrackingFpga::role () [inline], [override], [virtual]  
Role.  
Implements [GNSSBlockInterface](#).  
Definition at line 56 of file [galileo\\_e5a\\_dll\\_pll\\_tracking\\_fpga.h](#).  
Referenced by [GalileoE5aDlIPllTrackingFpga\(\)](#).

**10.138.3.8 set\_channel()**

void GalileoE5aDlIPllTrackingFpga::set\_channel (   
unsigned int channel) [override], [virtual]  
Set tracking channel unique ID.  
Implements [TrackingInterface](#).

#### 10.138.3.9 set\_gnss\_synchro()

```
void GalileoE5aDllPllTrackingFpga::set_gnss_synchro (  
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [TrackingInterface](#).

#### 10.138.3.10 start\_tracking()

```
void GalileoE5aDllPllTrackingFpga::start_tracking () [override], [virtual]
```

Start the tracking process in the FPGA.

Implements [TrackingInterface](#).

#### 10.138.3.11 stop\_tracking()

```
void GalileoE5aDllPllTrackingFpga::stop_tracking () [override], [virtual]
```

Stop the tracking process in the FPGA.

Implements [TrackingInterface](#).

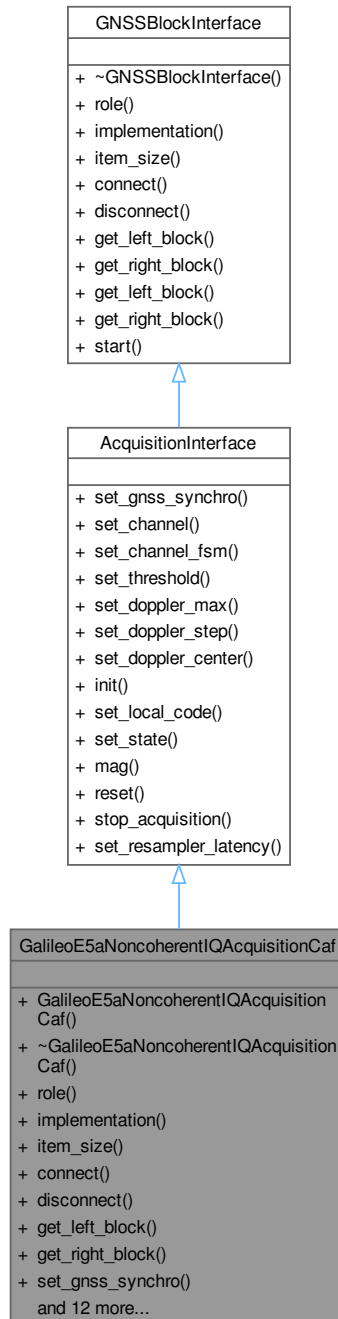
The documentation for this class was generated from the following file:

- [galileo\\_e5a\\_dll\\_pll\\_tracking\\_fpga.h](#)

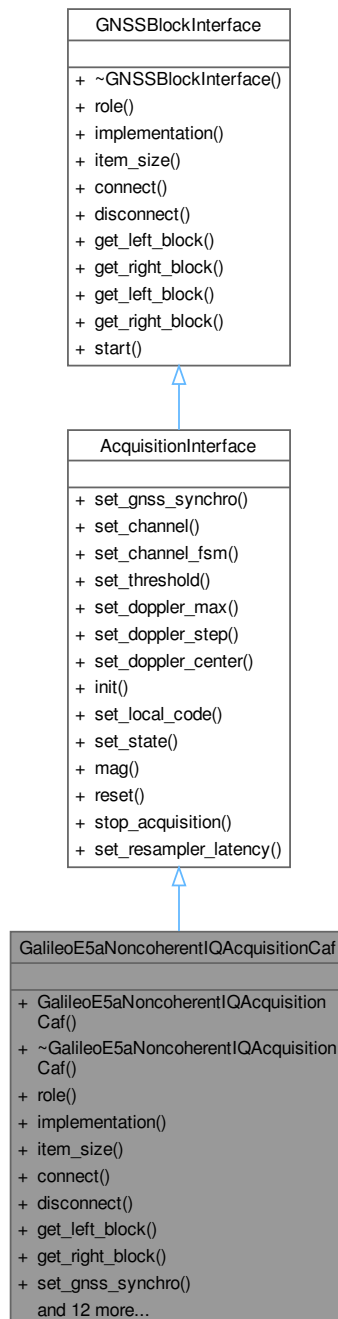


## 10.139 GalileoE5aNoncoherentIQAcquisitionCaf Class Reference

Inheritance diagram for GalileoE5aNoncoherentIQAcquisitionCaf:



Collaboration diagram for GalileoE5aNoncoherentIQAcquisitionCaf:



## Public Member Functions

- **GalileoE5aNoncoherentIQAcquisitionCaf** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "Galileo\_E5a\_Noncoherent\_IQ\_Acquisition\_CAF".
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [set\\_channel](#) (unsigned int channel) override  
*Set acquisition channel unique ID.*
- void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm) override  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold) override  
*Set statistics threshold of PCPS algorithm.*
- void [set\\_doppler\\_max](#) (unsigned int doppler\_max) override  
*Set maximum Doppler off grid search.*
- void [set\\_doppler\\_step](#) (unsigned int doppler\_step) override  
*Set Doppler steps for the grid search.*
- void [init](#) () override  
*Initializes acquisition algorithm.*
- void [set\\_local\\_code](#) () override  
*Sets local Galileo E5a code for PCPS acquisition algorithm.*
- signed int [mag](#) () override  
*Returns the maximum peak of grid search.*
- void [reset](#) () override  
*Restart acquisition algorithm.*
- void [set\\_state](#) (int state) override  
*If set to 1, ensures that acquisition starts at the first available sample.*
- void [stop\\_acquisition](#) () override  
*Stop running acquisition.*
- void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples \_\_attribute\_\_((unused))) override

### Public Member Functions inherited from [AcquisitionInterface](#)

- virtual void [set\\_doppler\\_center](#) (int doppler\_center \_\_attribute\_\_((unused)))
- virtual void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples)=0

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()  
*Start the flow of samples if needed.*

## 10.139.1 Detailed Description

Definition at line 43 of file [galileo\\_e5a\\_noncoherent\\_iq\\_acquisition\\_caf.h](#).

## 10.139.2 Member Function Documentation

### 10.139.2.1 [connect\(\)](#)

```
void GalileoE5aNoncoherentIQAcquisitionCaf::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.139.2.2 disconnect()

```
void GalileoE5aNoncoherentIQAcquisitionCaf::disconnect (  
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.139.2.3 get\_left\_block()

```
gr::basic_block_sptr GalileoE5aNoncoherentIQAcquisitionCaf::get_left_block () [override],  
[virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.139.2.4 get\_right\_block()

```
gr::basic_block_sptr GalileoE5aNoncoherentIQAcquisitionCaf::get_right_block () [override],  
[virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.139.2.5 implementation()

```
std::string GalileoE5aNoncoherentIQAcquisitionCaf::implementation () [inline], [override],  
[virtual]
```

Returns "Galileo\_E5a\_Noncoherent\_IQ\_Acquisition\_CAF".

Implements [GNSSBlockInterface](#).

Definition at line 61 of file [galileo\\_e5a\\_noncoherent\\_iq\\_acquisition\\_caf.h](#).

#### 10.139.2.6 init()

```
void GalileoE5aNoncoherentIQAcquisitionCaf::init () [override], [virtual]
```

Initializes acquisition algorithm.

Implements [AcquisitionInterface](#).

#### 10.139.2.7 item\_size()

```
size_t GalileoE5aNoncoherentIQAcquisitionCaf::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 66 of file [galileo\\_e5a\\_noncoherent\\_iq\\_acquisition\\_caf.h](#).

#### 10.139.2.8 mag()

```
signed int GalileoE5aNoncoherentIQAcquisitionCaf::mag () [override], [virtual]
```

Returns the maximum peak of grid search.

Implements [AcquisitionInterface](#).

#### 10.139.2.9 reset()

```
void GalileoE5aNoncoherentIQAcquisitionCaf::reset () [override], [virtual]
```

Restart acquisition algorithm.

Implements [AcquisitionInterface](#).

#### 10.139.2.10 role()

```
std::string GalileoE5aNoncoherentIQAcquisitionCaf::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 53 of file [galileo\\_e5a\\_noncoherent\\_iq\\_acquisition\\_caf.h](#).

**10.139.2.11 set\_channel()**

```
void GalileoE5aNoncoherentIQAcquisitionCaf::set_channel (
    unsigned int channel) [inline], [override], [virtual]
```

Set acquisition channel unique ID.

Implements [AcquisitionInterface](#).

Definition at line 86 of file [galileo\\_e5a\\_noncoherent\\_iq\\_acquisition\\_caf.h](#).

**10.139.2.12 set\_channel\_fsm()**

```
void GalileoE5aNoncoherentIQAcquisitionCaf::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline], [override], [virtual]
```

Set channel fsm associated to this acquisition instance.

Implements [AcquisitionInterface](#).

Definition at line 95 of file [galileo\\_e5a\\_noncoherent\\_iq\\_acquisition\\_caf.h](#).

**10.139.2.13 set\_doppler\_max()**

```
void GalileoE5aNoncoherentIQAcquisitionCaf::set_doppler_max (
    unsigned int doppler_max) [override], [virtual]
```

Set maximum Doppler off grid search.

Implements [AcquisitionInterface](#).

**10.139.2.14 set\_doppler\_step()**

```
void GalileoE5aNoncoherentIQAcquisitionCaf::set_doppler_step (
    unsigned int doppler_step) [override], [virtual]
```

Set Doppler steps for the grid search.

Implements [AcquisitionInterface](#).

**10.139.2.15 set\_gnss\_synchro()**

```
void GalileoE5aNoncoherentIQAcquisitionCaf::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [AcquisitionInterface](#).

**10.139.2.16 set\_local\_code()**

```
void GalileoE5aNoncoherentIQAcquisitionCaf::set_local_code () [override], [virtual]
```

Sets local Galileo E5a code for PCPS acquisition algorithm.

Implements [AcquisitionInterface](#).

**10.139.2.17 set\_resampler\_latency()**

```
void GalileoE5aNoncoherentIQAcquisitionCaf::set_resampler_latency (
    uint32_t latency_samples __attribute__((unused))) [inline], [override]
```

Definition at line 148 of file [galileo\\_e5a\\_noncoherent\\_iq\\_acquisition\\_caf.h](#).

**10.139.2.18 set\_state()**

```
void GalileoE5aNoncoherentIQAcquisitionCaf::set_state (
    int state) [override], [virtual]
```

If set to 1, ensures that acquisition starts at the first available sample.

**Parameters**

<i>state</i>	- int=1 forces start of acquisition
--------------	-------------------------------------

---

Implements [AcquisitionInterface](#).

#### 10.139.2.19 `set_threshold()`

```
void GalileoE5aNoncoherentIQAcquisitionCaf::set_threshold (
    float threshold) [override], [virtual]
```

Set statistics threshold of PCPS algorithm.

Implements [AcquisitionInterface](#).

#### 10.139.2.20 `stop_acquisition()`

```
void GalileoE5aNoncoherentIQAcquisitionCaf::stop_acquisition () [override], [virtual]
```

Stop running acquisition.

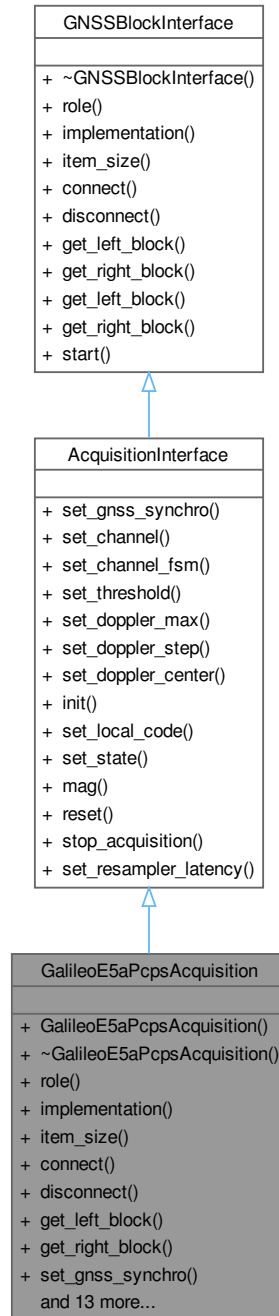
Implements [AcquisitionInterface](#).

The documentation for this class was generated from the following file:

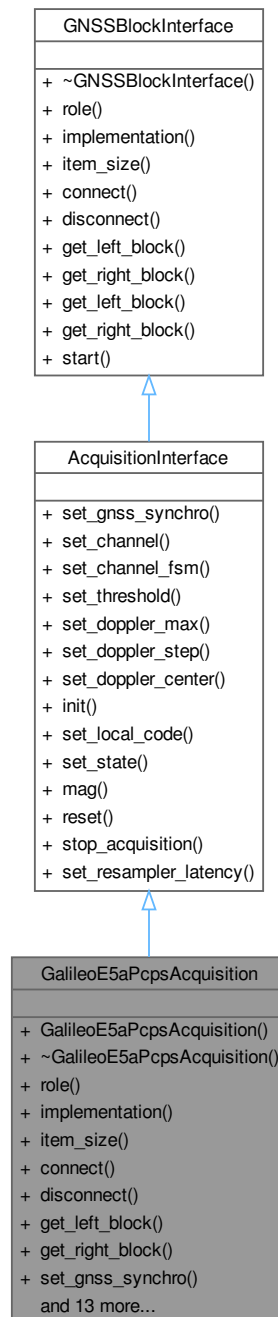
- [galileo\\_e5a\\_noncoherent\\_iq\\_acquisition\\_caf.h](#)

## 10.140 GalileoE5aPcpsAcquisition Class Reference

Inheritance diagram for GalileoE5aPcpsAcquisition:



Collaboration diagram for GalileoE5aPcpsAcquisition:



## Public Member Functions

- **GalileoE5aPcpsAcquisition** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override



- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_gnss\\_synchro](#) ([Gnss\\_Synchro](#) \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [set\\_channel](#) (unsigned int channel) override  
*Set acquisition channel unique ID.*
- void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm) override  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold) override  
*Set statistics threshold of PCPS algorithm.*
- void [set\\_doppler\\_max](#) (unsigned int doppler\_max) override  
*Set maximum Doppler off grid search.*
- void [set\\_doppler\\_step](#) (unsigned int doppler\_step) override  
*Set Doppler steps for the grid search.*
- void [set\\_doppler\\_center](#) (int doppler\_center) override  
*Set Doppler center for the grid search.*
- void [init](#) () override  
*Initializes acquisition algorithm.*
- void [set\\_local\\_code](#) () override  
*Sets local Galileo E5a code for PCPS acquisition algorithm.*
- signed int [mag](#) () override  
*Returns the maximum peak of grid search.*
- void [reset](#) () override  
*Restart acquisition algorithm.*
- void [set\\_state](#) (int state) override  
*If set to 1, ensures that acquisition starts at the first available sample.*
- void [stop\\_acquisition](#) () override  
*Stop running acquisition.*
- void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples) override  
*Sets the resampler latency to account it in the acquisition code delay estimation.*

### Public Member Functions inherited from [AcquisitionInterface](#)

- virtual void [set\\_doppler\\_center](#) (int doppler\_center \_\_attribute\_\_((unused)))

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

## 10.140.1 Detailed Description

Definition at line 38 of file [galileo\\_e5a\\_pcps\\_acquisition.h](#).

## 10.140.2 Member Function Documentation

### 10.140.2.1 [connect\(\)](#)

```
void GalileoE5aPcpsAcquisition::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

**10.140.2.2 disconnect()**

```
void GalileoE5aPcpsAcquisition::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

**10.140.2.3 get\_left\_block()**

```
gr::basic_block_sptr GalileoE5aPcpsAcquisition::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

**10.140.2.4 get\_right\_block()**

```
gr::basic_block_sptr GalileoE5aPcpsAcquisition::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

**10.140.2.5 implementation()**

```
std::string GalileoE5aPcpsAcquisition::implementation () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 54 of file [galileo\\_e5a\\_pcps\\_acquisition.h](#).

**10.140.2.6 init()**

```
void GalileoE5aPcpsAcquisition::init () [override], [virtual]
```

Initializes acquisition algorithm.

Implements [AcquisitionInterface](#).

**10.140.2.7 item\_size()**

```
size_t GalileoE5aPcpsAcquisition::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 59 of file [galileo\\_e5a\\_pcps\\_acquisition.h](#).

**10.140.2.8 mag()**

```
signed int GalileoE5aPcpsAcquisition::mag () [override], [virtual]
```

Returns the maximum peak of grid search.

Implements [AcquisitionInterface](#).

**10.140.2.9 reset()**

```
void GalileoE5aPcpsAcquisition::reset () [override], [virtual]
```

Restart acquisition algorithm.

Implements [AcquisitionInterface](#).

**10.140.2.10 role()**

```
std::string GalileoE5aPcpsAcquisition::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 49 of file [galileo\\_e5a\\_pcps\\_acquisition.h](#).

**10.140.2.11 set\_channel()**

```
void GalileoE5aPcpsAcquisition::set_channel (
    unsigned int channel) [inline], [override], [virtual]
```

Set acquisition channel unique ID.

Implements [AcquisitionInterface](#).

Definition at line 79 of file [galileo\\_e5a\\_pcps\\_acquisition.h](#).

**10.140.2.12 set\_channel\_fsm()**

```
void GalileoE5aPcpsAcquisition::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline], [override], [virtual]
```

Set channel fsm associated to this acquisition instance.

Implements [AcquisitionInterface](#).

Definition at line 88 of file [galileo\\_e5a\\_pcps\\_acquisition.h](#).

**10.140.2.13 set\_doppler\_center()**

```
void GalileoE5aPcpsAcquisition::set_doppler_center (
    int doppler_center) [override]
```

Set Doppler center for the grid search.

**10.140.2.14 set\_doppler\_max()**

```
void GalileoE5aPcpsAcquisition::set_doppler_max (
    unsigned int doppler_max) [override], [virtual]
```

Set maximum Doppler off grid search.

Implements [AcquisitionInterface](#).

**10.140.2.15 set\_doppler\_step()**

```
void GalileoE5aPcpsAcquisition::set_doppler_step (
    unsigned int doppler_step) [override], [virtual]
```

Set Doppler steps for the grid search.

Implements [AcquisitionInterface](#).

**10.140.2.16 set\_gnss\_synchro()**

```
void GalileoE5aPcpsAcquisition::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [AcquisitionInterface](#).

**10.140.2.17 set\_local\_code()**

```
void GalileoE5aPcpsAcquisition::set_local_code () [override], [virtual]
```

Sets local Galileo E5a code for PCPS acquisition algorithm.

Implements [AcquisitionInterface](#).

**10.140.2.18 set\_resampler\_latency()**

```
void GalileoE5aPcpsAcquisition::set_resampler_latency (
    uint32_t latency_samples) [override], [virtual]
```

Sets the resampler latency to account it in the acquisition code delay estimation.

Implements [AcquisitionInterface](#).

**10.140.2.19 set\_state()**

```
void GalileoE5aPcpsAcquisition::set_state (
    int state) [override], [virtual]
```

If set to 1, ensures that acquisition starts at the first available sample.

**Parameters**

<i>state</i>	- int=1 forces start of acquisition
--------------	-------------------------------------

Implements [AcquisitionInterface](#).

#### 10.140.2.20 `set_threshold()`

```
void GalileoE5aPcpsAcquisition::set_threshold (
    float threshold) [override], [virtual]
```

Set statistics threshold of PCPS algorithm.

Implements [AcquisitionInterface](#).

#### 10.140.2.21 `stop_acquisition()`

```
void GalileoE5aPcpsAcquisition::stop_acquisition () [override], [virtual]
```

Stop running acquisition.

Implements [AcquisitionInterface](#).

The documentation for this class was generated from the following file:

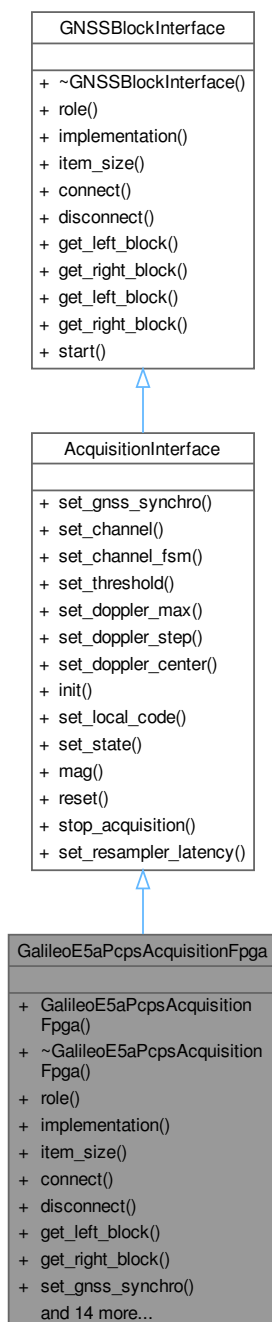
- [galileo\\_e5a\\_pcps\\_acquisition.h](#)

### 10.141 GalileoE5aPcpsAcquisitionFpga Class Reference

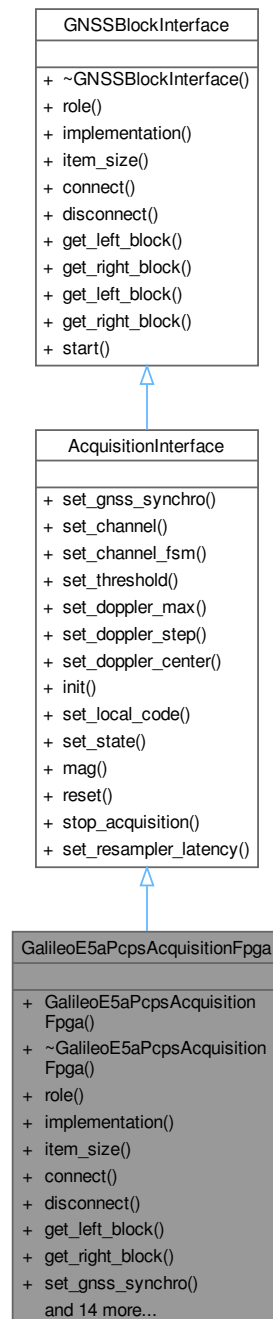
This class adapts a PCPS acquisition block off-loaded on an FPGA to an [AcquisitionInterface](#) for Galileo E5a signals.

```
#include <galileo_e5a_pcps_acquisition_fpga.h>
```

Inheritance diagram for GalileoE5aPcpsAcquisitionFpga:



Collaboration diagram for GalileoE5aPcpsAcquisitionFpga:



## Public Member Functions

- [GalileoE5aPcpsAcquisitionFpga](#) (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)

*Constructor.*

- [~GalileoE5aPcpsAcquisitionFpga](#) ()=default

*Destructor.*

- std::string [role](#) () override

- Role.*
- std::string [implementation](#) () override  
*Returns "Galileo\_E5a\_Pcps\_Acquisition\_FPGA".*
- size\_t [item\\_size](#) () override  
*Returns size of lv\_16sc\_t.*
- void [connect](#) (gr::top\_block\_sptr top\_block) override  
*Connect.*
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override  
*Disconnect.*
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override  
*Get left block.*
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override  
*Get right block.*
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common Gnss\_Synchro object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [set\\_channel](#) (unsigned int channel) override  
*Set acquisition channel unique ID.*
- void [set\\_channel\\_fsm](#) (std::weak\_ptr< ChannelFsm > channel\_fsm) override  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold) override  
*Set statistics threshold of PCPS algorithm.*
- void [set\\_doppler\\_max](#) (unsigned int doppler\_max) override  
*Set maximum Doppler off grid search.*
- void [set\\_doppler\\_step](#) (unsigned int doppler\_step) override  
*Set Doppler steps for the grid search.*
- void [set\\_doppler\\_center](#) (int doppler\_center) override  
*Set Doppler center for the grid search.*
- void [init](#) () override  
*Initializes acquisition algorithm.*
- void [set\\_local\\_code](#) () override  
*Sets local Galileo E5a code for PCPS acquisition algorithm.*
- signed int [mag](#) () override  
*Returns the maximum peak of grid search.*
- void [reset](#) () override  
*Restart acquisition algorithm.*
- void [set\\_state](#) (int state) override  
*If set to 1, ensures that acquisition starts at the first available sample.*
- void [set\\_single\\_doppler\\_flag](#) (unsigned int single\_doppler\_flag)  
*This function is only used in the unit tests.*
- void [stop\\_acquisition](#) () override  
*Stop running acquisition.*
- void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples \_\_attribute\_\_((unused))) override  
*Set resampler latency.*

### Public Member Functions inherited from [AcquisitionInterface](#)

- virtual void [set\\_doppler\\_center](#) (int doppler\_center \_\_attribute\_\_((unused)))
- virtual void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples)=0

## Public Member Functions inherited from GNSSBlockInterface

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.141.1 Detailed Description

This class adapts a PCPS acquisition block off-loaded on an FPGA to an [AcquisitionInterface](#) for Galileo E5a signals. Definition at line 43 of file [galileo\\_e5a\\_pcps\\_acquisition\\_fpga.h](#).

### 10.141.2 Constructor & Destructor Documentation

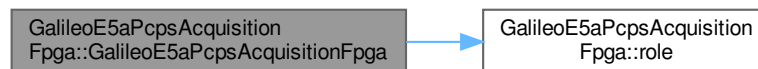
#### 10.141.2.1 GalileoE5aPcpsAcquisitionFpga()

```
GalileoE5aPcpsAcquisitionFpga::GalileoE5aPcpsAcquisitionFpga (
    const ConfigurationInterface * configuration,
    const std::string & role,
    unsigned int in_streams,
    unsigned int out_streams)
```

Constructor.

References [role\(\)](#).

Here is the call graph for this function:



#### 10.141.2.2 ~GalileoE5aPcpsAcquisitionFpga()

```
GalileoE5aPcpsAcquisitionFpga::~GalileoE5aPcpsAcquisitionFpga () [default]
```

Destructor.

### 10.141.3 Member Function Documentation

#### 10.141.3.1 connect()

```
void GalileoE5aPcpsAcquisitionFpga::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Connect.

Implements [GNSSBlockInterface](#).

#### 10.141.3.2 disconnect()

```
void GalileoE5aPcpsAcquisitionFpga::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Disconnect.

Implements [GNSSBlockInterface](#).



### 10.141.3.3 get\_left\_block()

```
gr::basic_block_sptr GalileoE5aPcpsAcquisitionFpga::get_left_block () [override], [virtual]
```

Get left block.

Implements [GNSSBlockInterface](#).

### 10.141.3.4 get\_right\_block()

```
gr::basic_block_sptr GalileoE5aPcpsAcquisitionFpga::get_right_block () [override], [virtual]
```

Get right block.

Implements [GNSSBlockInterface](#).

### 10.141.3.5 implementation()

```
std::string GalileoE5aPcpsAcquisitionFpga::implementation () [inline], [override], [virtual]
```

Returns "Galileo\_E5a\_Pcps\_Acquisition\_FPGA".

Implements [GNSSBlockInterface](#).

Definition at line 71 of file [galileo\\_e5a\\_pcps\\_acquisition\\_fpga.h](#).

### 10.141.3.6 init()

```
void GalileoE5aPcpsAcquisitionFpga::init () [override], [virtual]
```

Initializes acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.141.3.7 item\_size()

```
size_t GalileoE5aPcpsAcquisitionFpga::item_size () [inline], [override], [virtual]
```

Returns size of `lv_16sc_t`.

Implements [GNSSBlockInterface](#).

Definition at line 79 of file [galileo\\_e5a\\_pcps\\_acquisition\\_fpga.h](#).

### 10.141.3.8 mag()

```
signed int GalileoE5aPcpsAcquisitionFpga::mag () [override], [virtual]
```

Returns the maximum peak of grid search.

Implements [AcquisitionInterface](#).

### 10.141.3.9 reset()

```
void GalileoE5aPcpsAcquisitionFpga::reset () [override], [virtual]
```

Restart acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.141.3.10 role()

```
std::string GalileoE5aPcpsAcquisitionFpga::role () [inline], [override], [virtual]
```

Role.

Implements [GNSSBlockInterface](#).

Definition at line 63 of file [galileo\\_e5a\\_pcps\\_acquisition\\_fpga.h](#).

Referenced by [GalileoE5aPcpsAcquisitionFpga\(\)](#).

### 10.141.3.11 set\_channel()

```
void GalileoE5aPcpsAcquisitionFpga::set_channel (  
    unsigned int channel) [inline], [override], [virtual]
```

Set acquisition channel unique ID.

Implements [AcquisitionInterface](#).

Definition at line 114 of file [galileo\\_e5a\\_pcps\\_acquisition\\_fpga.h](#).

**10.141.3.12 set\_channel\_fsm()**

```
void GalileoE5aPcpsAcquisitionFpga::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline], [override], [virtual]
```

Set channel fsm associated to this acquisition instance.

Implements [AcquisitionInterface](#).

Definition at line 123 of file [galileo\\_e5a\\_pcps\\_acquisition\\_fpga.h](#).

**10.141.3.13 set\_doppler\_center()**

```
void GalileoE5aPcpsAcquisitionFpga::set_doppler_center (
    int doppler_center) [override]
```

Set Doppler center for the grid search.

**10.141.3.14 set\_doppler\_max()**

```
void GalileoE5aPcpsAcquisitionFpga::set_doppler_max (
    unsigned int doppler_max) [override], [virtual]
```

Set maximum Doppler off grid search.

Implements [AcquisitionInterface](#).

**10.141.3.15 set\_doppler\_step()**

```
void GalileoE5aPcpsAcquisitionFpga::set_doppler_step (
    unsigned int doppler_step) [override], [virtual]
```

Set Doppler steps for the grid search.

Implements [AcquisitionInterface](#).

**10.141.3.16 set\_gnss\_synchro()**

```
void GalileoE5aPcpsAcquisitionFpga::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [AcquisitionInterface](#).

**10.141.3.17 set\_local\_code()**

```
void GalileoE5aPcpsAcquisitionFpga::set_local_code () [override], [virtual]
```

Sets local Galileo E5a code for PCPS acquisition algorithm.

Implements [AcquisitionInterface](#).

**10.141.3.18 set\_resampler\_latency()**

```
void GalileoE5aPcpsAcquisitionFpga::set_resampler_latency (
    uint32_t latency_samples __attribute__((unused))) [inline], [override]
```

Set resampler latency.

Definition at line 189 of file [galileo\\_e5a\\_pcps\\_acquisition\\_fpga.h](#).

**10.141.3.19 set\_single\_doppler\_flag()**

```
void GalileoE5aPcpsAcquisitionFpga::set_single_doppler_flag (
    unsigned int single_doppler_flag)
```

This function is only used in the unit tests.

**10.141.3.20 set\_state()**

```
void GalileoE5aPcpsAcquisitionFpga::set_state (
    int state) [override], [virtual]
```

If set to 1, ensures that acquisition starts at the first available sample.

**Parameters**

<i>state</i>	- int=1 forces start of acquisition
--------------	-------------------------------------

Implements [AcquisitionInterface](#).

**10.141.3.21 set\_threshold()**

```
void GalileoE5aPcpsAcquisitionFpga::set_threshold (  
    float threshold) [override], [virtual]
```

Set statistics threshold of PCPS algorithm.

Implements [AcquisitionInterface](#).

**10.141.3.22 stop\_acquisition()**

```
void GalileoE5aPcpsAcquisitionFpga::stop_acquisition () [override], [virtual]
```

Stop running acquisition.

Implements [AcquisitionInterface](#).

The documentation for this class was generated from the following file:

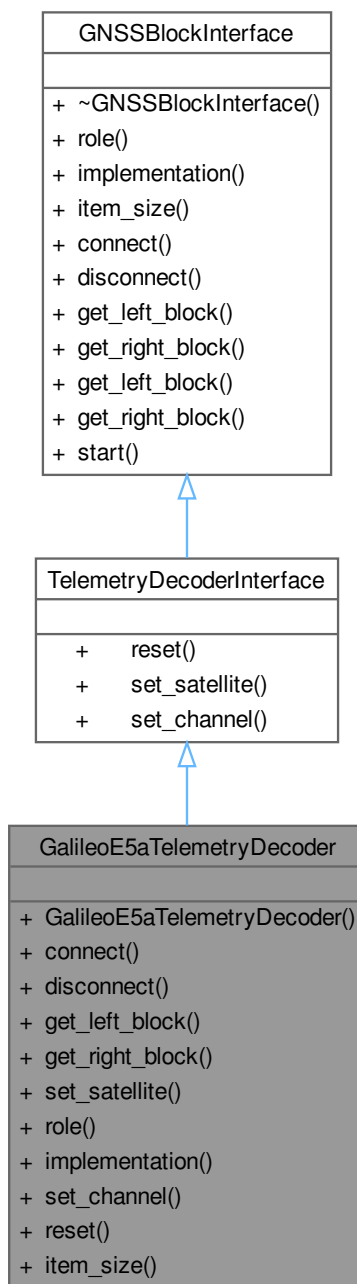
- [galileo\\_e5a\\_pcps\\_acquisition\\_fpga.h](#)

## 10.142 GalileoE5aTelemetryDecoder Class Reference

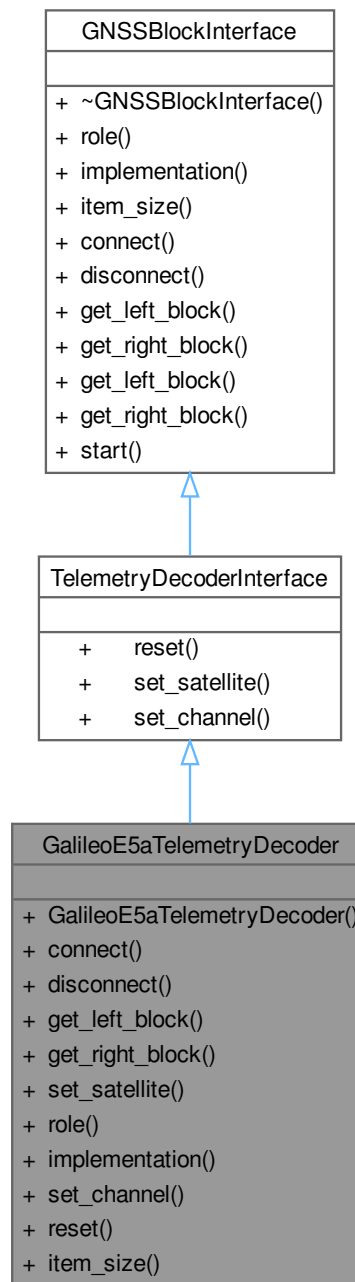
This class implements a NAV data decoder for Galileo INAV frames in E1B radio link.

```
#include <galileo_e5a_telemetry_decoder.h>
```

Inheritance diagram for GalileoE5aTelemetryDecoder:



Collaboration diagram for GalileoE5aTelemetryDecoder:



### Public Member Functions

- **GalileoE5aTelemetryDecoder** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override

- void [set\\_satellite](#) (const [Gnss\\_Satellite](#) &satellite) override
- std::string [role](#) () override
- std::string [implementation](#) () override  
*Returns "Galileo\_E5a\_Telemetry\_Decoder".*
- void [set\\_channel](#) (int channel) override
- void [reset](#) () override
- size\_t [item\\_size](#) () override

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.142.1 Detailed Description

This class implements a NAV data decoder for Galileo INAV frames in E1B radio link.  
Definition at line 46 of file [galileo\\_e5a\\_telemetry\\_decoder.h](#).

### 10.142.2 Member Function Documentation

#### 10.142.2.1 [connect\(\)](#)

```
void GalileoE5aTelemetryDecoder::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.142.2.2 [disconnect\(\)](#)

```
void GalileoE5aTelemetryDecoder::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.142.2.3 [get\\_left\\_block\(\)](#)

```
gr::basic_block_sptr GalileoE5aTelemetryDecoder::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.142.2.4 [get\\_right\\_block\(\)](#)

```
gr::basic_block_sptr GalileoE5aTelemetryDecoder::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.142.2.5 [implementation\(\)](#)

```
std::string GalileoE5aTelemetryDecoder::implementation () [inline], [override], [virtual]
```

Returns "Galileo\_E5a\_Telemetry\_Decoder".

Implements [GNSSBlockInterface](#).

Definition at line 70 of file [galileo\\_e5a\\_telemetry\\_decoder.h](#).

#### 10.142.2.6 [item\\_size\(\)](#)

```
size_t GalileoE5aTelemetryDecoder::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 82 of file [galileo\\_e5a\\_telemetry\\_decoder.h](#).

#### 10.142.2.7 reset()

```
void GalileoE5aTelemetryDecoder::reset () [inline], [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

Definition at line 77 of file [galileo\\_e5a\\_telemetry\\_decoder.h](#).

#### 10.142.2.8 role()

```
std::string GalileoE5aTelemetryDecoder::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 62 of file [galileo\\_e5a\\_telemetry\\_decoder.h](#).

#### 10.142.2.9 set\_channel()

```
void GalileoE5aTelemetryDecoder::set_channel (  
    int channel) [inline], [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

Definition at line 75 of file [galileo\\_e5a\\_telemetry\\_decoder.h](#).

#### 10.142.2.10 set\_satellite()

```
void GalileoE5aTelemetryDecoder::set_satellite (  
    const Gnss\_Satellite & satellite) [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

The documentation for this class was generated from the following file:

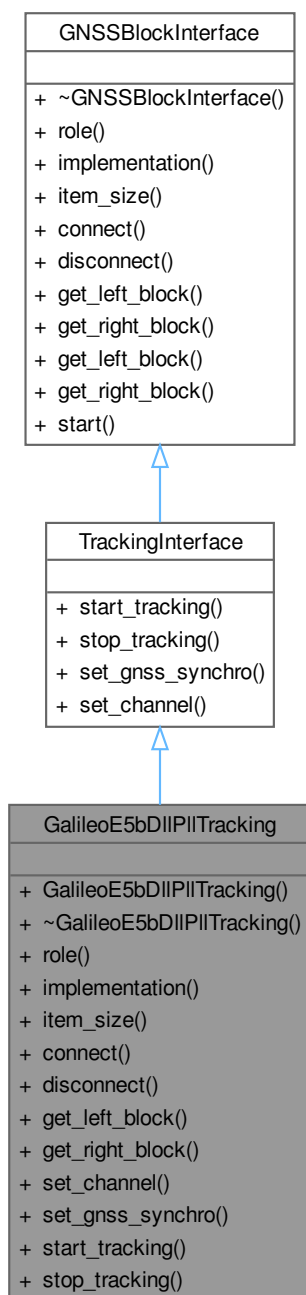
- [galileo\\_e5a\\_telemetry\\_decoder.h](#)

## 10.143 GalileoE5bDIPIITracking Class Reference

This class implements a code DLL + carrier PLL tracking loop.

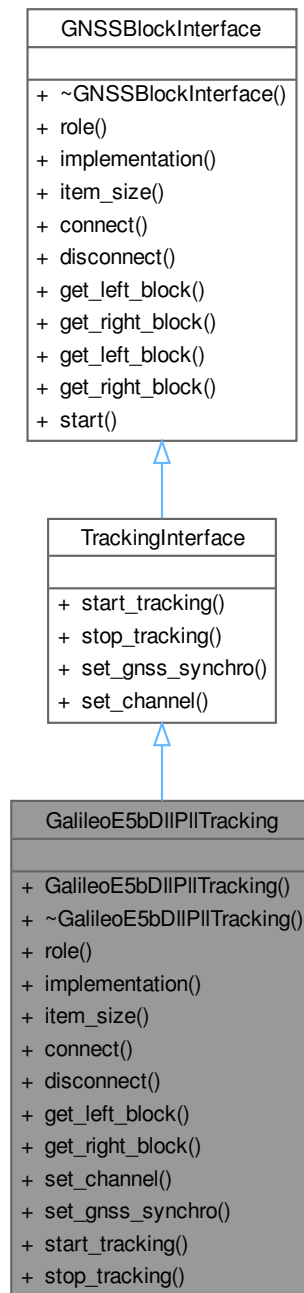
```
#include <galileo_e5b_dll_pll_tracking.h>
```

Inheritance diagram for GalileoE5bDIIPIITracking:





Collaboration diagram for GalileoE5bDIPIITracking:



### Public Member Functions

- **GalileoE5bDIPIITracking** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "Galileo\_E5b\_DLL\_PLL\_Tracking".
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override  
*Connect.*
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override  
*Disconnect.*
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override  
*Get left block.*
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override  
*Get right block.*
- void [set\\_channel](#) (unsigned int channel) override  
*Set tracking channel unique ID.*
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [start\\_tracking](#) () override
- void [stop\\_tracking](#) () override  
*Stop running tracking.*

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()  
*Start the flow of samples if needed.*

### 10.143.1 Detailed Description

This class implements a code DLL + carrier PLL tracking loop.  
Definition at line 42 of file [galileo\\_e5b\\_dll\\_pll\\_tracking.h](#).

### 10.143.2 Member Function Documentation

#### 10.143.2.1 connect()

```
void GalileoE5bDllPllTracking::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Connect.  
Implements [GNSSBlockInterface](#).

#### 10.143.2.2 disconnect()

```
void GalileoE5bDllPllTracking::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Disconnect.  
Implements [GNSSBlockInterface](#).

#### 10.143.2.3 get\_left\_block()

```
gr::basic_block_sptr GalileoE5bDllPllTracking::get_left_block () [override], [virtual]
```

Get left block.  
Implements [GNSSBlockInterface](#).

#### 10.143.2.4 get\_right\_block()

```
gr::basic_block_sptr GalileoE5bDllPllTracking::get_right_block () [override], [virtual]
```

Get right block.  
Implements [GNSSBlockInterface](#).

#### 10.143.2.5 implementation()

```
std::string GalileoE5bD11P11Tracking::implementation () [inline], [override], [virtual]
```

Returns "Galileo\_E5b\_DLL\_PLL\_Tracking".

Implements [GNSSBlockInterface](#).

Definition at line 59 of file [galileo\\_e5b\\_dll\\_pll\\_tracking.h](#).

#### 10.143.2.6 item\_size()

```
size_t GalileoE5bD11P11Tracking::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 64 of file [galileo\\_e5b\\_dll\\_pll\\_tracking.h](#).

#### 10.143.2.7 role()

```
std::string GalileoE5bD11P11Tracking::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 53 of file [galileo\\_e5b\\_dll\\_pll\\_tracking.h](#).

#### 10.143.2.8 set\_channel()

```
void GalileoE5bD11P11Tracking::set_channel (
    unsigned int channel) [override], [virtual]
```

Set tracking channel unique ID.

Implements [TrackingInterface](#).

#### 10.143.2.9 set\_gnss\_synchro()

```
void GalileoE5bD11P11Tracking::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [TrackingInterface](#).

#### 10.143.2.10 start\_tracking()

```
void GalileoE5bD11P11Tracking::start_tracking () [override], [virtual]
```

Implements [TrackingInterface](#).

#### 10.143.2.11 stop\_tracking()

```
void GalileoE5bD11P11Tracking::stop_tracking () [override], [virtual]
```

Stop running tracking.

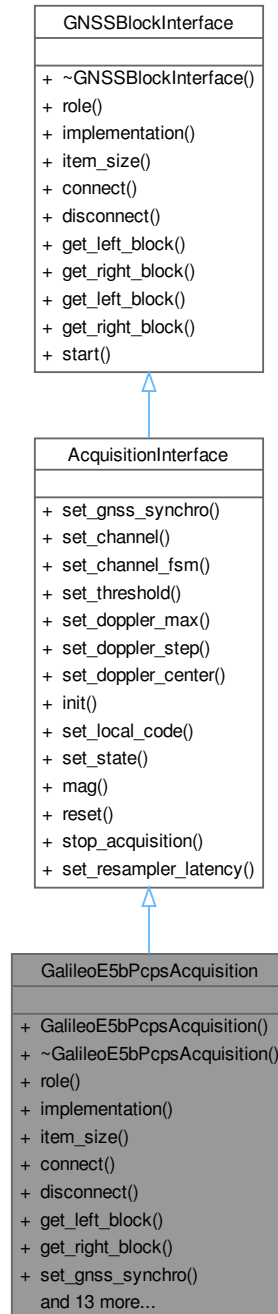
Implements [TrackingInterface](#).

The documentation for this class was generated from the following file:

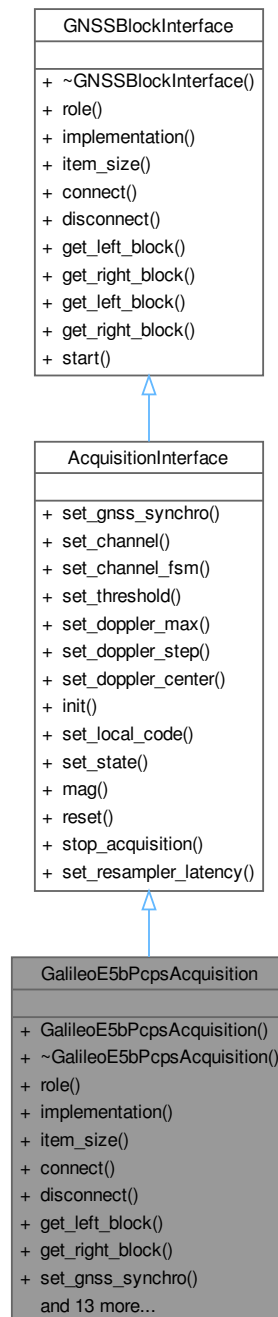
- [galileo\\_e5b\\_dll\\_pll\\_tracking.h](#)

## 10.144 GalileoE5bPcpsAcquisition Class Reference

Inheritance diagram for GalileoE5bPcpsAcquisition:



Collaboration diagram for GalileoE5bPcpsAcquisition:



## Public Member Functions

- [GalileoE5bPcpsAcquisition](#) (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)  
*Constructor.*
- [~GalileoE5bPcpsAcquisition](#) ()=default  
*Destructor.*
- std::string [role](#) () override

- Role.*

  - `std::string implementation ()` override  
*Returns "GALILEO\_E5b\_PCPS\_Acquisition".*
  - `size_t item_size ()` override  
*Returns size of lv\_16sc\_t.*
  - `void connect (gr::top_block_sptr top_block)` override  
*Connect.*
  - `void disconnect (gr::top_block_sptr top_block)` override  
*Disconnect.*
  - `gr::basic_block_sptr get_left_block ()` override  
*Get left block.*
  - `gr::basic_block_sptr get_right_block ()` override  
*Get right block.*
  - `void set_gnss_synchro (Gnss_Synchro *p_gnss_synchro)` override  
*Set acquisition/tracking common Gnss\_Synchro object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
  - `void set_channel (unsigned int channel)` override  
*Set acquisition channel unique ID.*
  - `void set_channel_fsm (std::weak_ptr< ChannelFsm > channel_fsm)` override  
*Set channel fsm associated to this acquisition instance.*
  - `void set_threshold (float threshold)` override  
*Set statistics threshold of PCPS algorithm.*
  - `void set_doppler_max (unsigned int doppler_max)` override  
*Set maximum Doppler off grid search.*
  - `void set_doppler_step (unsigned int doppler_step)` override  
*Set Doppler steps for the grid search.*
  - `void set_doppler_center (int doppler_center)` override  
*Set Doppler center for the grid search.*
  - `void init ()` override  
*Initializes acquisition algorithm.*
  - `void set_local_code ()` override  
*Sets local Galileo E5b code for PCPS acquisition algorithm.*
  - `signed int mag ()` override  
*Returns the maximum peak of grid search.*
  - `void reset ()` override  
*Restart acquisition algorithm.*
  - `void set_state (int state)` override  
*If set to 1, ensures that acquisition starts at the first available sample.*
  - `void stop_acquisition ()` override  
*Stop running acquisition.*
  - `void set_resampler_latency (uint32_t latency_samples)` override  
*Sets the resampler latency to account it in the acquisition code delay estimation.*

### Public Member Functions inherited from **AcquisitionInterface**

- virtual `void set_doppler_center (int doppler_center __attribute__((unused)))`

### Public Member Functions inherited from **GNSSBlockInterface**

- virtual `gr::basic_block_sptr get_left_block (int RF_channel)`
- virtual `gr::basic_block_sptr get_right_block (int RF_channel)`
- virtual `void start ()`  
*Start the flow of samples if needed.*

### 10.144.1 Detailed Description

Definition at line 39 of file [galileo\\_e5b\\_pcps\\_acquisition.h](#).

### 10.144.2 Constructor & Destructor Documentation

#### 10.144.2.1 GalileoE5bPcpsAcquisition()

```
GalileoE5bPcpsAcquisition::GalileoE5bPcpsAcquisition (
    const ConfigurationInterface * configuration,
    const std::string & role,
    unsigned int in_streams,
    unsigned int out_streams)
```

Constructor.

References [role\(\)](#).

Here is the call graph for this function:



#### 10.144.2.2 ~GalileoE5bPcpsAcquisition()

```
GalileoE5bPcpsAcquisition::~~GalileoE5bPcpsAcquisition () [default]
```

Destructor.

### 10.144.3 Member Function Documentation

#### 10.144.3.1 connect()

```
void GalileoE5bPcpsAcquisition::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Connect.

Implements [GNSSBlockInterface](#).

#### 10.144.3.2 disconnect()

```
void GalileoE5bPcpsAcquisition::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Disconnect.

Implements [GNSSBlockInterface](#).

#### 10.144.3.3 get\_left\_block()

```
gr::basic_block_sptr GalileoE5bPcpsAcquisition::get_left_block () [override], [virtual]
```

Get left block.

Implements [GNSSBlockInterface](#).

#### 10.144.3.4 get\_right\_block()

```
gr::basic_block_sptr GalileoE5bPcpsAcquisition::get_right_block () [override], [virtual]
```

Get right block.

Implements [GNSSBlockInterface](#).

**10.144.3.5 implementation()**

```
std::string GalileoE5bPcpsAcquisition::implementation () [inline], [override], [virtual]
```

Returns "GALILEO\_E5b\_PCPS\_Acquisition".

Implements [GNSSBlockInterface](#).

Definition at line 67 of file [galileo\\_e5b\\_pcps\\_acquisition.h](#).

**10.144.3.6 init()**

```
void GalileoE5bPcpsAcquisition::init () [override], [virtual]
```

Initializes acquisition algorithm.

Implements [AcquisitionInterface](#).

**10.144.3.7 item\_size()**

```
size_t GalileoE5bPcpsAcquisition::item_size () [inline], [override], [virtual]
```

Returns size of `lv_16sc_t`.

Implements [GNSSBlockInterface](#).

Definition at line 75 of file [galileo\\_e5b\\_pcps\\_acquisition.h](#).

**10.144.3.8 mag()**

```
signed int GalileoE5bPcpsAcquisition::mag () [override], [virtual]
```

Returns the maximum peak of grid search.

Implements [AcquisitionInterface](#).

**10.144.3.9 reset()**

```
void GalileoE5bPcpsAcquisition::reset () [override], [virtual]
```

Restart acquisition algorithm.

Implements [AcquisitionInterface](#).

**10.144.3.10 role()**

```
std::string GalileoE5bPcpsAcquisition::role () [inline], [override], [virtual]
```

Role.

Implements [GNSSBlockInterface](#).

Definition at line 58 of file [galileo\\_e5b\\_pcps\\_acquisition.h](#).

Referenced by [GalileoE5bPcpsAcquisition\(\)](#).

**10.144.3.11 set\_channel()**

```
void GalileoE5bPcpsAcquisition::set_channel (
    unsigned int channel) [inline], [override], [virtual]
```

Set acquisition channel unique ID.

Implements [AcquisitionInterface](#).

Definition at line 110 of file [galileo\\_e5b\\_pcps\\_acquisition.h](#).

**10.144.3.12 set\_channel\_fsm()**

```
void GalileoE5bPcpsAcquisition::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline], [override], [virtual]
```

Set channel fsm associated to this acquisition instance.

Implements [AcquisitionInterface](#).

Definition at line 119 of file [galileo\\_e5b\\_pcps\\_acquisition.h](#).

**10.144.3.13 set\_doppler\_center()**

```
void GalileoE5bPcpsAcquisition::set_doppler_center (
    int doppler_center) [override]
```



Set Doppler center for the grid search.

#### 10.144.3.14 `set_doppler_max()`

```
void GalileoE5bPcpsAcquisition::set_doppler_max (
    unsigned int doppler_max) [override], [virtual]
```

Set maximum Doppler off grid search.

Implements [AcquisitionInterface](#).

#### 10.144.3.15 `set_doppler_step()`

```
void GalileoE5bPcpsAcquisition::set_doppler_step (
    unsigned int doppler_step) [override], [virtual]
```

Set Doppler steps for the grid search.

Implements [AcquisitionInterface](#).

#### 10.144.3.16 `set_gnss_synchro()`

```
void GalileoE5bPcpsAcquisition::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [AcquisitionInterface](#).

#### 10.144.3.17 `set_local_code()`

```
void GalileoE5bPcpsAcquisition::set_local_code () [override], [virtual]
```

Sets local Galileo E5b code for PCPS acquisition algorithm.

Implements [AcquisitionInterface](#).

#### 10.144.3.18 `set_resampler_latency()`

```
void GalileoE5bPcpsAcquisition::set_resampler_latency (
    uint32_t latency_samples) [override], [virtual]
```

Sets the resampler latency to account it in the acquisition code delay estimation.

Implements [AcquisitionInterface](#).

#### 10.144.3.19 `set_state()`

```
void GalileoE5bPcpsAcquisition::set_state (
    int state) [override], [virtual]
```

If set to 1, ensures that acquisition starts at the first available sample.

##### Parameters

<i>state</i>	- int=1 forces start of acquisition
--------------	-------------------------------------

Implements [AcquisitionInterface](#).

#### 10.144.3.20 `set_threshold()`

```
void GalileoE5bPcpsAcquisition::set_threshold (
    float threshold) [override], [virtual]
```

Set statistics threshold of PCPS algorithm.

Implements [AcquisitionInterface](#).

#### 10.144.3.21 stop\_acquisition()

```
void GalileoE5bPcpsAcquisition::stop_acquisition () [override], [virtual]
```

Stop running acquisition.

Implements [AcquisitionInterface](#).

The documentation for this class was generated from the following file:

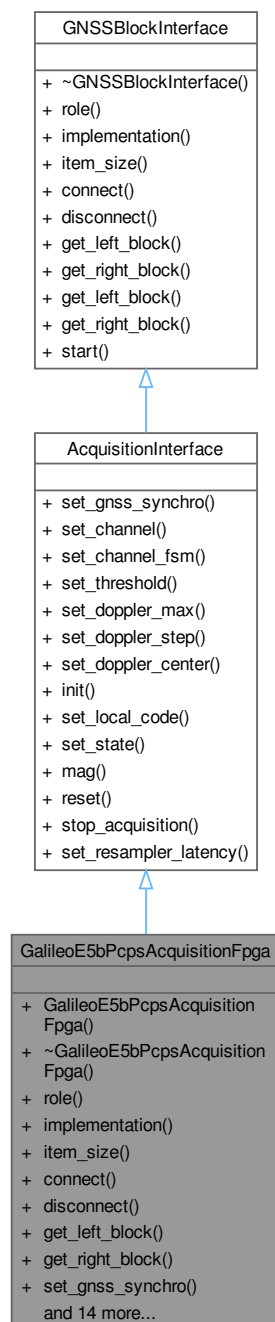
- [galileo\\_e5b\\_pcps\\_acquisition.h](#)

### 10.145 GalileoE5bPcpsAcquisitionFpga Class Reference

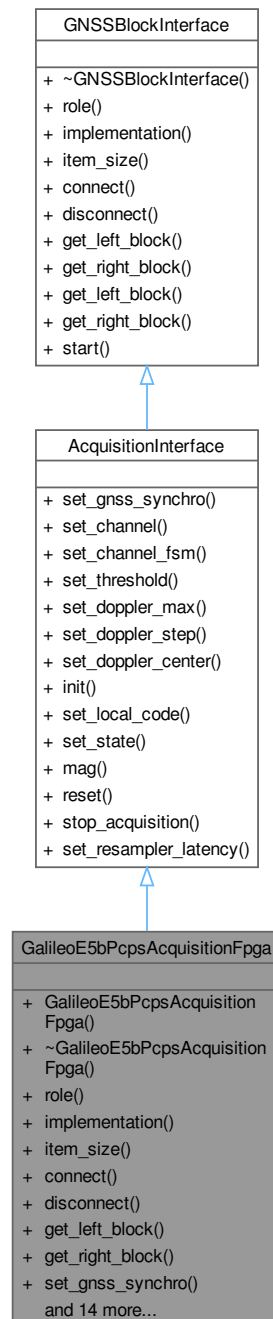
This class adapts a PCPS acquisition block off-loaded on an FPGA to an [AcquisitionInterface](#) for Galileo E5b signals.

```
#include <galileo_e5b_pcps_acquisition_fpga.h>
```

Inheritance diagram for GalileoE5bPcpsAcquisitionFpga:



Collaboration diagram for GalileoE5bPcpsAcquisitionFpga:



## Public Member Functions

- [GalileoE5bPcpsAcquisitionFpga](#) (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)  
*Constructor.*
- [~GalileoE5bPcpsAcquisitionFpga](#) ()=default  
*Destructor.*
- std::string [role](#) () override

- Role.*
- std::string [implementation](#) () override  
*Returns "Galileo\_E5b\_Pcps\_Acquisition\_FPGA".*
- size\_t [item\\_size](#) () override  
*Returns size of lv\_16sc\_t.*
- void [connect](#) (gr::top\_block\_sptr top\_block) override  
*Connect.*
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override  
*Disconnect.*
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override  
*Get left block.*
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override  
*Get right block.*
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common Gnss\_Synchro object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [set\\_channel](#) (unsigned int channel) override  
*Set acquisition channel unique ID.*
- void [set\\_channel\\_fsm](#) (std::weak\_ptr< ChannelFsm > channel\_fsm) override  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold) override  
*Set statistics threshold of PCPS algorithm.*
- void [set\\_doppler\\_max](#) (unsigned int doppler\_max) override  
*Set maximum Doppler off grid search.*
- void [set\\_doppler\\_step](#) (unsigned int doppler\_step) override  
*Set Doppler steps for the grid search.*
- void [set\\_doppler\\_center](#) (int doppler\_center) override  
*Set Doppler center for the grid search.*
- void [init](#) () override  
*Initializes acquisition algorithm.*
- void [set\\_local\\_code](#) () override  
*Sets local Galileo E5b code for PCPS acquisition algorithm.*
- signed int [mag](#) () override  
*Returns the maximum peak of grid search.*
- void [reset](#) () override  
*Restart acquisition algorithm.*
- void [set\\_state](#) (int state) override  
*If set to 1, ensures that acquisition starts at the first available sample.*
- void [set\\_single\\_doppler\\_flag](#) (unsigned int single\_doppler\_flag)  
*This function is only used in the unit tests.*
- void [stop\\_acquisition](#) () override  
*Stop running acquisition.*
- void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples \_\_attribute\_\_((unused))) override  
*Set resampler latency.*

### Public Member Functions inherited from [AcquisitionInterface](#)

- virtual void [set\\_doppler\\_center](#) (int doppler\_center \_\_attribute\_\_((unused)))
- virtual void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples)=0

## Public Member Functions inherited from GNSSBlockInterface

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.145.1 Detailed Description

This class adapts a PCPS acquisition block off-loaded on an FPGA to an [AcquisitionInterface](#) for Galileo E5b signals. Definition at line 43 of file [galileo\\_e5b\\_pcps\\_acquisition\\_fpga.h](#).

### 10.145.2 Constructor & Destructor Documentation

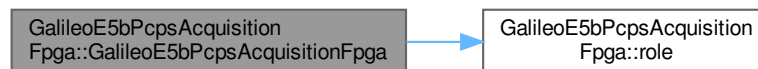
#### 10.145.2.1 GalileoE5bPcpsAcquisitionFpga()

```
GalileoE5bPcpsAcquisitionFpga::GalileoE5bPcpsAcquisitionFpga (
    const ConfigurationInterface * configuration,
    const std::string & role,
    unsigned int in_streams,
    unsigned int out_streams)
```

Constructor.

References [role\(\)](#).

Here is the call graph for this function:



#### 10.145.2.2 ~GalileoE5bPcpsAcquisitionFpga()

```
GalileoE5bPcpsAcquisitionFpga::~GalileoE5bPcpsAcquisitionFpga () [default]
Destructor.
```

### 10.145.3 Member Function Documentation

#### 10.145.3.1 connect()

```
void GalileoE5bPcpsAcquisitionFpga::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Connect.

Implements [GNSSBlockInterface](#).

#### 10.145.3.2 disconnect()

```
void GalileoE5bPcpsAcquisitionFpga::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Disconnect.

Implements [GNSSBlockInterface](#).

### 10.145.3.3 get\_left\_block()

```
gr::basic_block_sptr GalileoE5bPcpsAcquisitionFpga::get_left_block () [override], [virtual]
```

Get left block.

Implements [GNSSBlockInterface](#).

### 10.145.3.4 get\_right\_block()

```
gr::basic_block_sptr GalileoE5bPcpsAcquisitionFpga::get_right_block () [override], [virtual]
```

Get right block.

Implements [GNSSBlockInterface](#).

### 10.145.3.5 implementation()

```
std::string GalileoE5bPcpsAcquisitionFpga::implementation () [inline], [override], [virtual]
```

Returns "Galileo\_E5b\_Pcps\_Acquisition\_FPGA".

Implements [GNSSBlockInterface](#).

Definition at line 70 of file [galileo\\_e5b\\_pcps\\_acquisition\\_fpga.h](#).

### 10.145.3.6 init()

```
void GalileoE5bPcpsAcquisitionFpga::init () [override], [virtual]
```

Initializes acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.145.3.7 item\_size()

```
size_t GalileoE5bPcpsAcquisitionFpga::item_size () [inline], [override], [virtual]
```

Returns size of `lv_16sc_t`.

Implements [GNSSBlockInterface](#).

Definition at line 78 of file [galileo\\_e5b\\_pcps\\_acquisition\\_fpga.h](#).

### 10.145.3.8 mag()

```
signed int GalileoE5bPcpsAcquisitionFpga::mag () [override], [virtual]
```

Returns the maximum peak of grid search.

Implements [AcquisitionInterface](#).

### 10.145.3.9 reset()

```
void GalileoE5bPcpsAcquisitionFpga::reset () [override], [virtual]
```

Restart acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.145.3.10 role()

```
std::string GalileoE5bPcpsAcquisitionFpga::role () [inline], [override], [virtual]
```

Role.

Implements [GNSSBlockInterface](#).

Definition at line 62 of file [galileo\\_e5b\\_pcps\\_acquisition\\_fpga.h](#).

Referenced by [GalileoE5bPcpsAcquisitionFpga\(\)](#).

### 10.145.3.11 set\_channel()

```
void GalileoE5bPcpsAcquisitionFpga::set_channel (  
    unsigned int channel) [inline], [override], [virtual]
```

Set acquisition channel unique ID.

Implements [AcquisitionInterface](#).

Definition at line 113 of file [galileo\\_e5b\\_pcps\\_acquisition\\_fpga.h](#).

**10.145.3.12 set\_channel\_fsm()**

```
void GalileoE5bPcpsAcquisitionFpga::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline], [override], [virtual]
```

Set channel fsm associated to this acquisition instance.

Implements [AcquisitionInterface](#).

Definition at line 122 of file [galileo\\_e5b\\_pcps\\_acquisition\\_fpga.h](#).

**10.145.3.13 set\_doppler\_center()**

```
void GalileoE5bPcpsAcquisitionFpga::set_doppler_center (
    int doppler_center) [override]
```

Set Doppler center for the grid search.

**10.145.3.14 set\_doppler\_max()**

```
void GalileoE5bPcpsAcquisitionFpga::set_doppler_max (
    unsigned int doppler_max) [override], [virtual]
```

Set maximum Doppler off grid search.

Implements [AcquisitionInterface](#).

**10.145.3.15 set\_doppler\_step()**

```
void GalileoE5bPcpsAcquisitionFpga::set_doppler_step (
    unsigned int doppler_step) [override], [virtual]
```

Set Doppler steps for the grid search.

Implements [AcquisitionInterface](#).

**10.145.3.16 set\_gnss\_synchro()**

```
void GalileoE5bPcpsAcquisitionFpga::set_gnss_synchro (
    Gnss\_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [AcquisitionInterface](#).

**10.145.3.17 set\_local\_code()**

```
void GalileoE5bPcpsAcquisitionFpga::set_local_code () [override], [virtual]
```

Sets local Galileo E5b code for PCPS acquisition algorithm.

Implements [AcquisitionInterface](#).

**10.145.3.18 set\_resampler\_latency()**

```
void GalileoE5bPcpsAcquisitionFpga::set_resampler_latency (
    uint32_t latency_samples __attribute__((unused))) [inline], [override]
```

Set resampler latency.

Definition at line 188 of file [galileo\\_e5b\\_pcps\\_acquisition\\_fpga.h](#).

**10.145.3.19 set\_single\_doppler\_flag()**

```
void GalileoE5bPcpsAcquisitionFpga::set_single_doppler_flag (
    unsigned int single_doppler_flag)
```

This function is only used in the unit tests.

**10.145.3.20 set\_state()**

```
void GalileoE5bPcpsAcquisitionFpga::set_state (
    int state) [override], [virtual]
```

If set to 1, ensures that acquisition starts at the first available sample.



**Parameters**

<i>state</i>	- int=1 forces start of acquisition
--------------	-------------------------------------

Implements [AcquisitionInterface](#).

**10.145.3.21 set\_threshold()**

```
void GalileoE5bPcpsAcquisitionFpga::set_threshold (  
    float threshold) [override], [virtual]
```

Set statistics threshold of PCPS algorithm.

Implements [AcquisitionInterface](#).

**10.145.3.22 stop\_acquisition()**

```
void GalileoE5bPcpsAcquisitionFpga::stop_acquisition () [override], [virtual]
```

Stop running acquisition.

Implements [AcquisitionInterface](#).

The documentation for this class was generated from the following file:

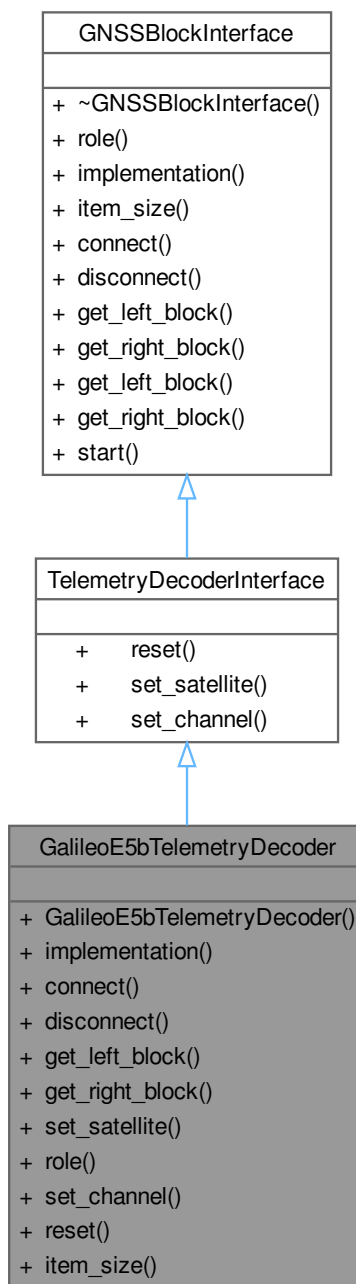
- [galileo\\_e5b\\_pcps\\_acquisition\\_fpga.h](#)

## 10.146 GalileoE5bTelemetryDecoder Class Reference

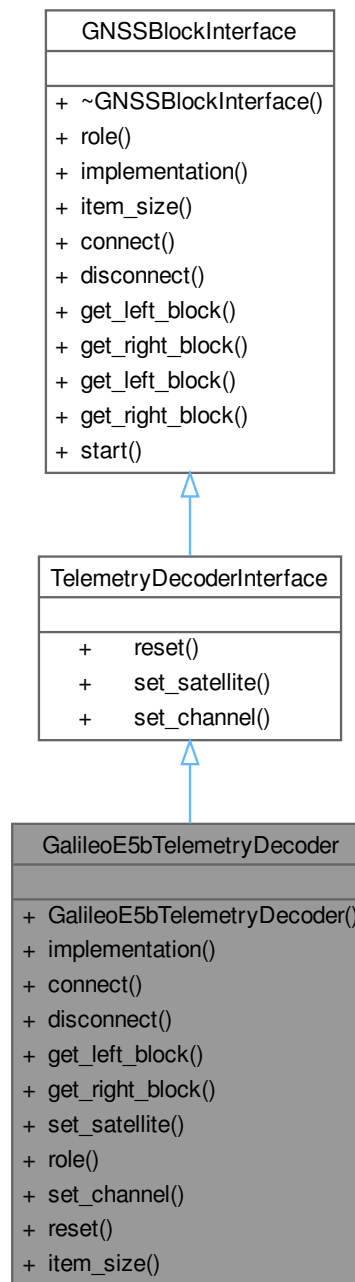
This class implements a NAV data decoder for Galileo INAV frames in E5b radio link.

```
#include <galileo_e5b_telemetry_decoder.h>
```

Inheritance diagram for GalileoE5bTelemetryDecoder:



Collaboration diagram for GalileoE5bTelemetryDecoder:



## Public Member Functions

- **GalileoE5bTelemetryDecoder** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [implementation](#) () override  
*Returns "Galileo\_E5b\_Telemetry\_Decoder".*
- void [connect](#) (gr::top\_block\_sptr top\_block) override  
*Connect.*

- void [disconnect](#) (gr::top\_block\_sptr top\_block) override  
*Disconnect.*
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override  
*Get left block.*
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override  
*Get right block.*
- void [set\\_satellite](#) (const [Gnss\\_Satellite](#) &satellite) override
- std::string [role](#) () override
- void [set\\_channel](#) (int channel) override
- void [reset](#) () override
- size\_t [item\\_size](#) () override

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()  
*Start the flow of samples if needed.*

### 10.146.1 Detailed Description

This class implements a NAV data decoder for Galileo INAV frames in E5b radio link.  
Definition at line 44 of file [galileo\\_e5b\\_telemetry\\_decoder.h](#).

### 10.146.2 Member Function Documentation

#### 10.146.2.1 connect()

```
void GalileoE5bTelemetryDecoder::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Connect.  
Implements [GNSSBlockInterface](#).

#### 10.146.2.2 disconnect()

```
void GalileoE5bTelemetryDecoder::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Disconnect.  
Implements [GNSSBlockInterface](#).

#### 10.146.2.3 get\_left\_block()

```
gr::basic_block_sptr GalileoE5bTelemetryDecoder::get_left_block () [override], [virtual]
```

Get left block.  
Implements [GNSSBlockInterface](#).

#### 10.146.2.4 get\_right\_block()

```
gr::basic_block_sptr GalileoE5bTelemetryDecoder::get_right_block () [override], [virtual]
```

Get right block.  
Implements [GNSSBlockInterface](#).

#### 10.146.2.5 implementation()

```
std::string GalileoE5bTelemetryDecoder::implementation () [inline], [override], [virtual]
```

Returns "Galileo\_E5b\_Telemetry\_Decoder".  
Implements [GNSSBlockInterface](#).  
Definition at line 56 of file [galileo\\_e5b\\_telemetry\\_decoder.h](#).

#### 10.146.2.6 item\_size()

```
size_t GalileoE5bTelemetryDecoder::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 95 of file [galileo\\_e5b\\_telemetry\\_decoder.h](#).

#### 10.146.2.7 reset()

```
void GalileoE5bTelemetryDecoder::reset () [inline], [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

Definition at line 90 of file [galileo\\_e5b\\_telemetry\\_decoder.h](#).

#### 10.146.2.8 role()

```
std::string GalileoE5bTelemetryDecoder::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 83 of file [galileo\\_e5b\\_telemetry\\_decoder.h](#).

#### 10.146.2.9 set\_channel()

```
void GalileoE5bTelemetryDecoder::set_channel (  
    int channel) [inline], [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

Definition at line 88 of file [galileo\\_e5b\\_telemetry\\_decoder.h](#).

#### 10.146.2.10 set\_satellite()

```
void GalileoE5bTelemetryDecoder::set_satellite (  
    const Gnss\_Satellite & satellite) [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

The documentation for this class was generated from the following file:

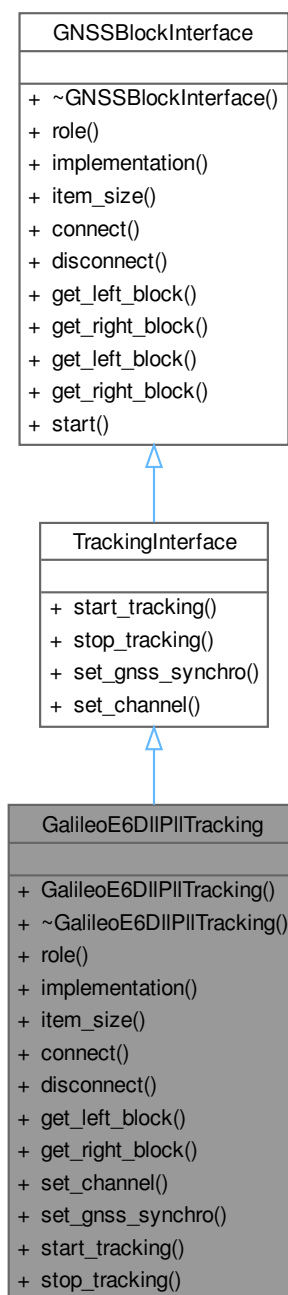
- [galileo\\_e5b\\_telemetry\\_decoder.h](#)

## 10.147 GalileoE6DIPIITracking Class Reference

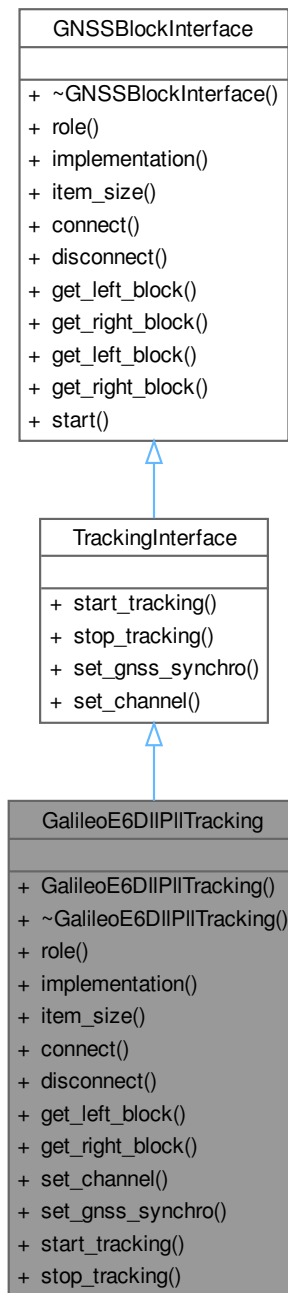
This class implements a code DLL + carrier PLL tracking loop.

```
#include <galileo_e6_dll_pll_tracking.h>
```

Inheritance diagram for GalileoE6DIIPIITracking:



Collaboration diagram for GalileoE6DIPIITracking:



### Public Member Functions

- **GalileoE6DIPIITracking** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "Galileo\_E6\_DLL\_PLL\_Tracking".
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override  
*Connect.*
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override  
*Disconnect.*
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override  
*Get left block.*
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override  
*Get right block.*
- void [set\\_channel](#) (unsigned int channel) override  
*Set tracking channel unique ID.*
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [start\\_tracking](#) () override
- void [stop\\_tracking](#) () override  
*Stop running tracking.*

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()  
*Start the flow of samples if needed.*

### 10.147.1 Detailed Description

This class implements a code DLL + carrier PLL tracking loop.  
Definition at line 37 of file [galileo\\_e6\\_dll\\_pll\\_tracking.h](#).

### 10.147.2 Member Function Documentation

#### 10.147.2.1 connect()

```
void GalileoE6DllPllTracking::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Connect.  
Implements [GNSSBlockInterface](#).

#### 10.147.2.2 disconnect()

```
void GalileoE6DllPllTracking::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Disconnect.  
Implements [GNSSBlockInterface](#).

#### 10.147.2.3 get\_left\_block()

```
gr::basic_block_sptr GalileoE6DllPllTracking::get_left_block () [override], [virtual]
```

Get left block.  
Implements [GNSSBlockInterface](#).

#### 10.147.2.4 get\_right\_block()

```
gr::basic_block_sptr GalileoE6DllPllTracking::get_right_block () [override], [virtual]
```

Get right block.  
Implements [GNSSBlockInterface](#).



**10.147.2.5 implementation()**

```
std::string GalileoE6D1lP1lTracking::implementation () [inline], [override], [virtual]
```

Returns "Galileo\_E6\_DLL\_PLL\_Tracking".

Implements [GNSSBlockInterface](#).

Definition at line 54 of file [galileo\\_e6\\_dll\\_pll\\_tracking.h](#).

**10.147.2.6 item\_size()**

```
size_t GalileoE6D1lP1lTracking::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 59 of file [galileo\\_e6\\_dll\\_pll\\_tracking.h](#).

**10.147.2.7 role()**

```
std::string GalileoE6D1lP1lTracking::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 48 of file [galileo\\_e6\\_dll\\_pll\\_tracking.h](#).

**10.147.2.8 set\_channel()**

```
void GalileoE6D1lP1lTracking::set_channel (
    unsigned int channel) [override], [virtual]
```

Set tracking channel unique ID.

Implements [TrackingInterface](#).

**10.147.2.9 set\_gnss\_synchro()**

```
void GalileoE6D1lP1lTracking::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [TrackingInterface](#).

**10.147.2.10 start\_tracking()**

```
void GalileoE6D1lP1lTracking::start_tracking () [override], [virtual]
```

Implements [TrackingInterface](#).

**10.147.2.11 stop\_tracking()**

```
void GalileoE6D1lP1lTracking::stop_tracking () [override], [virtual]
```

Stop running tracking.

Implements [TrackingInterface](#).

The documentation for this class was generated from the following file:

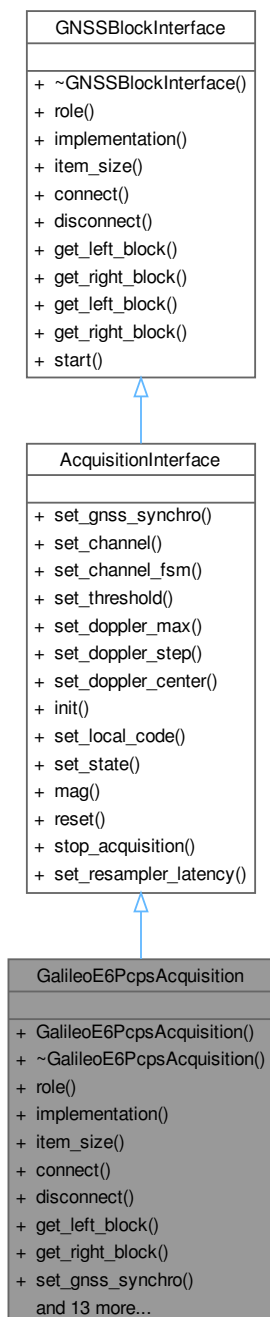
- [galileo\\_e6\\_dll\\_pll\\_tracking.h](#)

**10.148 GalileoE6PcpsAcquisition Class Reference**

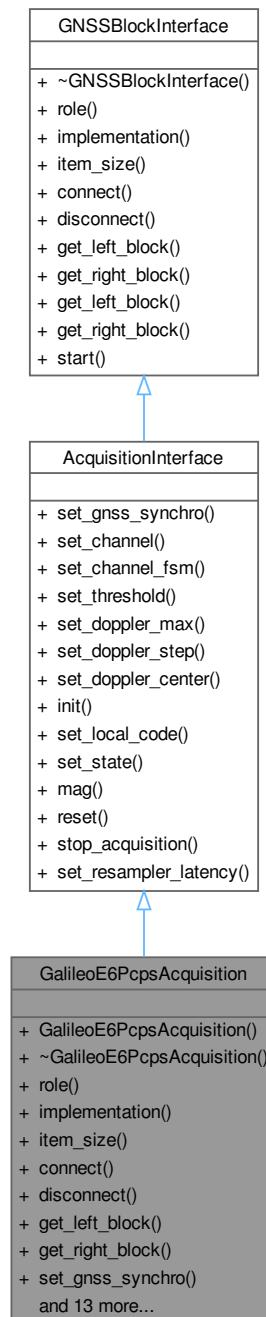
This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Galileo E6 Signals.

```
#include <galileo_e6_pcps_acquisition.h>
```

Inheritance diagram for GalileoE6PcpsAcquisition:



Collaboration diagram for GalileoE6PcpsAcquisition:



## Public Member Functions

- **GalileoE6PcpsAcquisition** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override
  - Returns "Galileo\_E6\_PCPS\_Acquisition".
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [set\\_channel](#) (unsigned int channel) override  
*Set acquisition channel unique ID.*
- void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm) override  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold) override  
*Set statistics threshold of PCPS algorithm.*
- void [set\\_doppler\\_max](#) (unsigned int doppler\_max) override  
*Set maximum Doppler off grid search.*
- void [set\\_doppler\\_step](#) (unsigned int doppler\_step) override  
*Set Doppler steps for the grid search.*
- void [set\\_doppler\\_center](#) (int doppler\_center) override  
*Set Doppler center for the grid search.*
- void [init](#) () override  
*Initializes acquisition algorithm.*
- void [set\\_local\\_code](#) () override  
*Sets local code for Galileo E1 PCPS acquisition algorithm.*
- signed int [mag](#) () override  
*Returns the maximum peak of grid search.*
- void [reset](#) () override  
*Restart acquisition algorithm.*
- void [set\\_state](#) (int state) override  
*If state = 1, it forces the block to start acquiring from the first sample.*
- void [stop\\_acquisition](#) () override  
*Stop running acquisition.*
- void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples) override  
*Sets the resampler latency to account it in the acquisition code delay estimation.*

### Public Member Functions inherited from [AcquisitionInterface](#)

- virtual void [set\\_doppler\\_center](#) (int doppler\_center \_\_attribute\_\_((unused)))

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

#### 10.148.1 Detailed Description

This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Galileo E6 Signals.  
 Definition at line 44 of file [galileo\\_e6\\_pcps\\_acquisition.h](#).

## 10.148.2 Member Function Documentation

### 10.148.2.1 connect()

```
void GalileoE6PcpsAcquisition::connect (  
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.148.2.2 disconnect()

```
void GalileoE6PcpsAcquisition::disconnect (  
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.148.2.3 get\_left\_block()

```
gr::basic_block_sptr GalileoE6PcpsAcquisition::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.148.2.4 get\_right\_block()

```
gr::basic_block_sptr GalileoE6PcpsAcquisition::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.148.2.5 implementation()

```
std::string GalileoE6PcpsAcquisition::implementation () [inline], [override], [virtual]
```

Returns "Galileo\_E6\_PCPS\_Acquisition".

Implements [GNSSBlockInterface](#).

Definition at line 63 of file [galileo\\_e6\\_pcps\\_acquisition.h](#).

### 10.148.2.6 init()

```
void GalileoE6PcpsAcquisition::init () [override], [virtual]
```

Initializes acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.148.2.7 item\_size()

```
size_t GalileoE6PcpsAcquisition::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 68 of file [galileo\\_e6\\_pcps\\_acquisition.h](#).

### 10.148.2.8 mag()

```
signed int GalileoE6PcpsAcquisition::mag () [override], [virtual]
```

Returns the maximum peak of grid search.

Implements [AcquisitionInterface](#).

### 10.148.2.9 reset()

```
void GalileoE6PcpsAcquisition::reset () [override], [virtual]
```

Restart acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.148.2.10 role()

```
std::string GalileoE6PcpsAcquisition::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 55 of file [galileo\\_e6\\_pcps\\_acquisition.h](#).

**10.148.2.11 set\_channel()**

```
void GalileoE6PcpsAcquisition::set_channel (
    unsigned int channel) [inline], [override], [virtual]
```

Set acquisition channel unique ID.

Implements [AcquisitionInterface](#).

Definition at line 88 of file [galileo\\_e6\\_pcps\\_acquisition.h](#).

**10.148.2.12 set\_channel\_fsm()**

```
void GalileoE6PcpsAcquisition::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline], [override], [virtual]
```

Set channel fsm associated to this acquisition instance.

Implements [AcquisitionInterface](#).

Definition at line 97 of file [galileo\\_e6\\_pcps\\_acquisition.h](#).

**10.148.2.13 set\_doppler\_center()**

```
void GalileoE6PcpsAcquisition::set_doppler_center (
    int doppler_center) [override]
```

Set Doppler center for the grid search.

**10.148.2.14 set\_doppler\_max()**

```
void GalileoE6PcpsAcquisition::set_doppler_max (
    unsigned int doppler_max) [override], [virtual]
```

Set maximum Doppler off grid search.

Implements [AcquisitionInterface](#).

**10.148.2.15 set\_doppler\_step()**

```
void GalileoE6PcpsAcquisition::set_doppler_step (
    unsigned int doppler_step) [override], [virtual]
```

Set Doppler steps for the grid search.

Implements [AcquisitionInterface](#).

**10.148.2.16 set\_gnss\_synchro()**

```
void GalileoE6PcpsAcquisition::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [AcquisitionInterface](#).

**10.148.2.17 set\_local\_code()**

```
void GalileoE6PcpsAcquisition::set_local_code () [override], [virtual]
```

Sets local code for Galileo E1 PCPS acquisition algorithm.

Implements [AcquisitionInterface](#).

**10.148.2.18 set\_resampler\_latency()**

```
void GalileoE6PcpsAcquisition::set_resampler_latency (
    uint32_t latency_samples) [override], [virtual]
```

Sets the resampler latency to account it in the acquisition code delay estimation.

Implements [AcquisitionInterface](#).

**10.148.2.19 set\_state()**

```
void GalileoE6PcpsAcquisition::set_state (  
    int state) [override], [virtual]
```

If state = 1, it forces the block to start acquiring from the first sample.

Implements [AcquisitionInterface](#).

**10.148.2.20 set\_threshold()**

```
void GalileoE6PcpsAcquisition::set_threshold (  
    float threshold) [override], [virtual]
```

Set statistics threshold of PCPS algorithm.

Implements [AcquisitionInterface](#).

**10.148.2.21 stop\_acquisition()**

```
void GalileoE6PcpsAcquisition::stop_acquisition () [override], [virtual]
```

Stop running acquisition.

Implements [AcquisitionInterface](#).

The documentation for this class was generated from the following file:

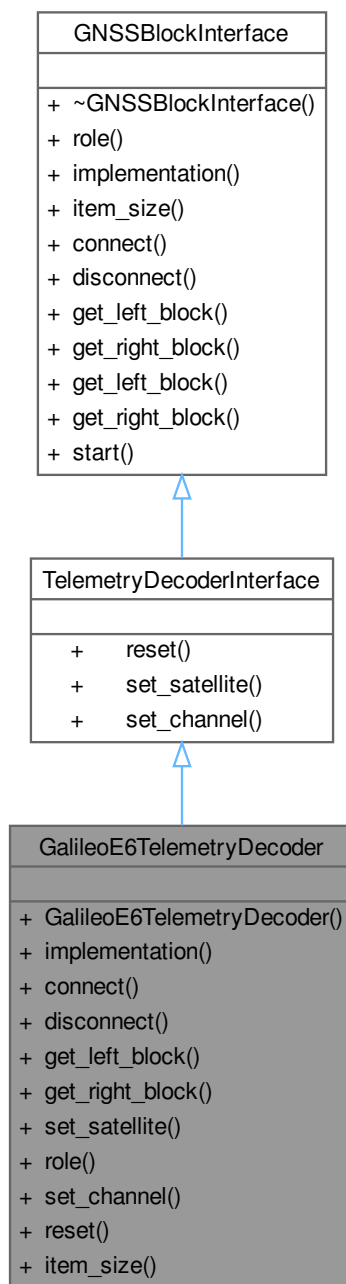
- [galileo\\_e6\\_pcps\\_acquisition.h](#)

**10.149 GalileoE6TelemetryDecoder Class Reference**

This class implements a NAV data decoder for Galileo CNAV frames in E6 radio link.

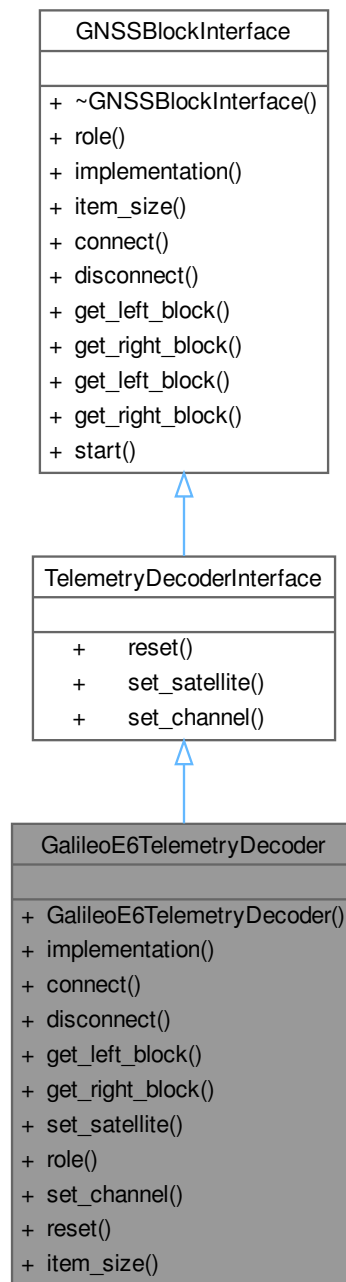
```
#include <galileo_e6_telemetry_decoder.h>
```

Inheritance diagram for GalileoE6TelemetryDecoder:





Collaboration diagram for GalileoE6TelemetryDecoder:



## Public Member Functions

- **GalileoE6TelemetryDecoder** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [implementation](#) () override  
Returns "Galileo\_E6\_Telemetry\_Decoder".
- void [connect](#) (gr::top\_block\_sptr top\_block) override  
Connect.

- void [disconnect](#) (gr::top\_block\_sptr top\_block) override  
*Disconnect.*
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override  
*Get left block.*
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override  
*Get right block.*
- void [set\\_satellite](#) (const [Gnss\\_Satellite](#) &satellite) override
- std::string [role](#) () override
- void [set\\_channel](#) (int channel) override
- void [reset](#) () override
- size\_t [item\\_size](#) () override

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()  
*Start the flow of samples if needed.*

### 10.149.1 Detailed Description

This class implements a NAV data decoder for Galileo CNAV frames in E6 radio link.  
Definition at line 43 of file [galileo\\_e6\\_telemetry\\_decoder.h](#).

### 10.149.2 Member Function Documentation

#### 10.149.2.1 connect()

```
void GalileoE6TelemetryDecoder::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Connect.  
Implements [GNSSBlockInterface](#).

#### 10.149.2.2 disconnect()

```
void GalileoE6TelemetryDecoder::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Disconnect.  
Implements [GNSSBlockInterface](#).

#### 10.149.2.3 get\_left\_block()

```
gr::basic_block_sptr GalileoE6TelemetryDecoder::get_left_block () [override], [virtual]
```

Get left block.  
Implements [GNSSBlockInterface](#).

#### 10.149.2.4 get\_right\_block()

```
gr::basic_block_sptr GalileoE6TelemetryDecoder::get_right_block () [override], [virtual]
```

Get right block.  
Implements [GNSSBlockInterface](#).

#### 10.149.2.5 implementation()

```
std::string GalileoE6TelemetryDecoder::implementation () [inline], [override], [virtual]
```

Returns "Galileo\_E6\_Telemetry\_Decoder".  
Implements [GNSSBlockInterface](#).  
Definition at line 55 of file [galileo\\_e6\\_telemetry\\_decoder.h](#).

#### 10.149.2.6 item\_size()

```
size_t GalileoE6TelemetryDecoder::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 94 of file [galileo\\_e6\\_telemetry\\_decoder.h](#).

#### 10.149.2.7 reset()

```
void GalileoE6TelemetryDecoder::reset () [inline], [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

Definition at line 89 of file [galileo\\_e6\\_telemetry\\_decoder.h](#).

#### 10.149.2.8 role()

```
std::string GalileoE6TelemetryDecoder::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 82 of file [galileo\\_e6\\_telemetry\\_decoder.h](#).

#### 10.149.2.9 set\_channel()

```
void GalileoE6TelemetryDecoder::set_channel (
    int channel) [inline], [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

Definition at line 87 of file [galileo\\_e6\\_telemetry\\_decoder.h](#).

#### 10.149.2.10 set\_satellite()

```
void GalileoE6TelemetryDecoder::set_satellite (
    const Gnss\_Satellite & satellite) [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

The documentation for this class was generated from the following file:

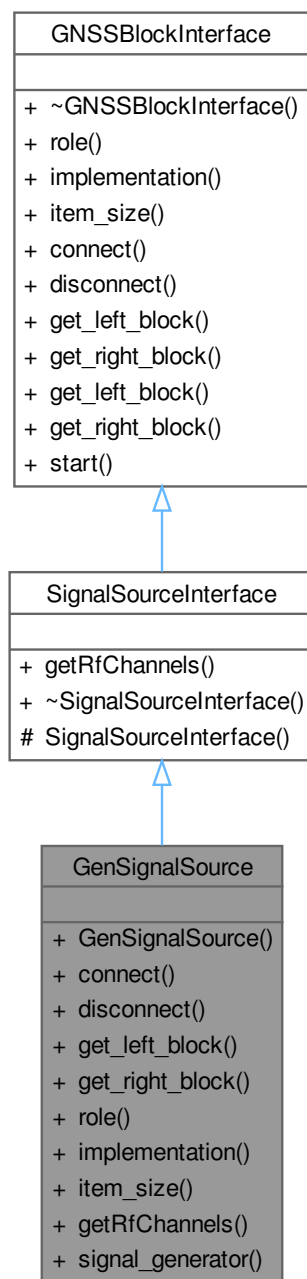
- [galileo\\_e6\\_telemetry\\_decoder.h](#)

## 10.150 GenSignalSource Class Reference

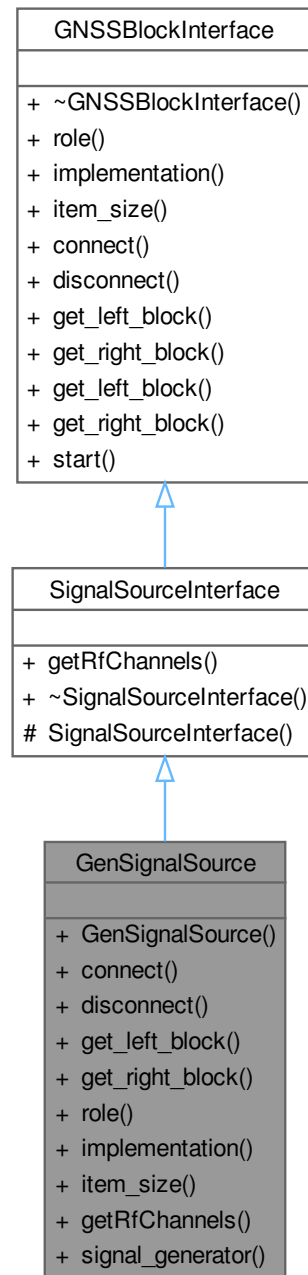
This class wraps blocks that generates synthesized GNSS signal and filters the signal.

```
#include <gen_signal_source.h>
```

Inheritance diagram for GenSignalSource:



Collaboration diagram for GenSignalSource:



### Public Member Functions

- [GenSignalSource](#) (`std::shared_ptr< GNSSBlockInterface > signal_generator, std::shared_ptr< GNSSBlockInterface > filter, std::string role, Concurrent\_Queue< pmt::pmt_t > *queue)`  
*Constructor.*
- void [connect](#) (`gr::top_block_sptr top_block`) override
- void [disconnect](#) (`gr::top_block_sptr top_block`) override
- `gr::basic_block_sptr` [get\\_left\\_block](#) () override

- `gr::basic_block_sptr` [get\\_right\\_block](#) () override
- `std::string` [role](#) () override
- `std::string` [implementation](#) () override
- *Returns "Signal Source".*
- `size_t` [item\\_size](#) () override
- `size_t` [getRfChannels](#) () const final
- `std::shared_ptr< GNSSBlockInterface >` [signal\\_generator](#) () const

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual `gr::basic_block_sptr` [get\\_left\\_block](#) (int RF\_channel)
- virtual `gr::basic_block_sptr` [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.150.1 Detailed Description

This class wraps blocks that generates synthesized GNSS signal and filters the signal.  
Definition at line 41 of file [gen\\_signal\\_source.h](#).

### 10.150.2 Constructor & Destructor Documentation

#### 10.150.2.1 GenSignalSource()

```
GenSignalSource::GenSignalSource (
    std::shared_ptr< GNSSBlockInterface > signal_generator,
    std::shared_ptr< GNSSBlockInterface > filter,
    std::string role,
    Concurrent_Queue< pmt::pmt_t > * queue)
```

Constructor.

### 10.150.3 Member Function Documentation

#### 10.150.3.1 connect()

```
void GenSignalSource::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.150.3.2 disconnect()

```
void GenSignalSource::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.150.3.3 get\_left\_block()

```
gr::basic_block_sptr GenSignalSource::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.150.3.4 get\_right\_block()

```
gr::basic_block_sptr GenSignalSource::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.150.3.5 getRfChannels()

```
size_t GenSignalSource::getRfChannels () const [inline], [final], [virtual]
```

Implements [SignalSourceInterface](#).

Definition at line 57 of file [gen\\_signal\\_source.h](#).

**10.150.3.6 implementation()**

```
std::string GenSignalSource::implementation () [inline], [override], [virtual]
```

Returns "Signal Source".

Implements [GNSSBlockInterface](#).

Definition at line 55 of file [gen\\_signal\\_source.h](#).

**10.150.3.7 item\_size()**

```
size_t GenSignalSource::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 56 of file [gen\\_signal\\_source.h](#).

**10.150.3.8 role()**

```
std::string GenSignalSource::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 53 of file [gen\\_signal\\_source.h](#).

**10.150.3.9 signal\_generator()**

```
std::shared_ptr< GNSSBlockInterface > GenSignalSource::signal_generator () const [inline]
```

Definition at line 59 of file [gen\\_signal\\_source.h](#).

The documentation for this class was generated from the following file:

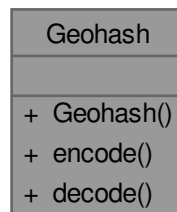
- [gen\\_signal\\_source.h](#)

**10.151 Geohash Class Reference**

Class for geohash encoding / decoding See <https://en.wikipedia.org/wiki/Geohash>.

```
#include <geohash.h>
```

Collaboration diagram for Geohash:

**Public Member Functions**

- std::string [encode](#) (double lat, double lon, int precision=-1) const
- std::array< double, 2 > [decode](#) (std::string geohash) const

**10.151.1 Detailed Description**

Class for geohash encoding / decoding See <https://en.wikipedia.org/wiki/Geohash>.

Definition at line 34 of file [geohash.h](#).

## 10.151.2 Member Function Documentation

### 10.151.2.1 decode()

```
std::array< double, 2 > Geohash::decode (
    std::string geohash) const
```

Decode geohash to latitude/longitude (location is approximate centre of geohash cell, to reasonable precision).

#### Parameters

<i>{string}</i>	geohash - <a href="#">Geohash</a> string to be converted to latitude/longitude.
-----------------	---

#### Returns

{lat, lon} (Center of) geohashed location.

#### Exceptions

<i>Invalid</i>	geohash.
----------------	----------

### 10.151.2.2 encode()

```
std::string Geohash::encode (
    double lat,
    double lon,
    int precision = -1) const
```

Encodes latitude/longitude to geohash, either to specified precision or to automatically evaluated precision.

#### Parameters

<i>{double}</i>	lat - Latitude in degrees.
<i>{double}</i>	lon - Longitude in degrees.
<i>{int}</i>	[precision] - Number of characters in resulting geohash.

#### Returns

{string} [Geohash](#) of supplied latitude/longitude.

#### Exceptions

<i>Invalid</i>	geohash.
----------------	----------

The documentation for this class was generated from the following file:

- [geohash.h](#)

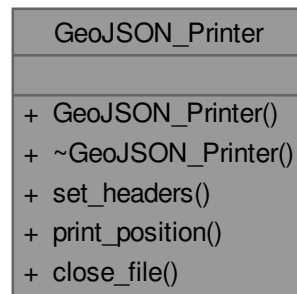
## 10.152 GeoJSON\_Printer Class Reference

Prints PVT solutions in GeoJSON format file.

```
#include <geojson_printer.h>
```



Collaboration diagram for GeoJSON\_Printer:



### Public Member Functions

- **GeoJSON\_Printer** (const std::string &base\_path=".")
- bool **set\_headers** (const std::string &filename, bool time\_tag\_name=true)
- bool **print\_position** (const [Pvt\\_Solution](#) \*const position)
- bool **close\_file** ()

### 10.152.1 Detailed Description

Prints PVT solutions in GeoJSON format file.

See <https://tools.ietf.org/html/rfc7946>

Definition at line 39 of file [geojson\\_printer.h](#).

The documentation for this class was generated from the following file:

- [geojson\\_printer.h](#)

## 10.153 `geph_t` Struct Reference

Collaboration diagram for `geph_t`:



### Public Attributes

- int `sat`
- int `iode`
- int `frq`
- int `svh`
- int `sva`
- int `age`
- `gtime_t` `toe`
- `gtime_t` `tof`
- double `pos` [3]
- double `vel` [3]
- double `acc` [3]
- double `taun`
- double `gamn`
- double `dtaun`

### 10.153.1 Detailed Description

Definition at line 462 of file [rtklib.h](#).

### 10.153.2 Member Data Documentation

#### 10.153.2.1 `acc`

```
double geph_t::acc[3]
```

Definition at line 472 of file [rtklib.h](#).

#### 10.153.2.2 `age`

```
int geph_t::age
```

Definition at line 467 of file [rtklib.h](#).

#### 10.153.2.3 `dtaun`

```
double geph_t::dtaun
```

Definition at line 474 of file [rtklib.h](#).

#### 10.153.2.4 `frq`

```
int geph_t::frq
```

Definition at line 466 of file [rtklib.h](#).

#### 10.153.2.5 `gamn`

```
double geph_t::gamn
```

Definition at line 473 of file [rtklib.h](#).

#### 10.153.2.6 `iode`

```
int geph_t::iode
```

Definition at line 465 of file [rtklib.h](#).

#### 10.153.2.7 `pos`

```
double geph_t::pos[3]
```

Definition at line 470 of file [rtklib.h](#).

#### 10.153.2.8 `sat`

```
int geph_t::sat
```

Definition at line 464 of file [rtklib.h](#).

#### 10.153.2.9 `sva`

```
int geph_t::sva
```

Definition at line 467 of file [rtklib.h](#).

#### 10.153.2.10 `svh`

```
int geph_t::svh
```

Definition at line 467 of file [rtklib.h](#).

#### 10.153.2.11 `taun`

```
double geph_t::taun
```

Definition at line 473 of file [rtklib.h](#).

**10.153.2.12 toe**

`gtime_t` `geph_t::toe`

Definition at line 468 of file [rtklib.h](#).

**10.153.2.13 tof**

`gtime_t` `geph_t::tof`

Definition at line 469 of file [rtklib.h](#).

**10.153.2.14 vel**

`double` `geph_t::vel[3]`

Definition at line 471 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

**10.154 Glonass\_Gnav\_Almanac Class Reference**

This class is a storage for the GLONASS SV ALMANAC data as described GLONASS ICD (Edition 5.1)

`#include <glonass_gnav_almanac.h>`

Collaboration diagram for Glonass\_Gnav\_Almanac:

Glonass_Gnav_Almanac
+ d_n_A
+ d_H_n_A
+ d_lambda_n_A
+ d_t_lambda_n_A
+ d_Delta_i_n_A
+ d_Delta_T_n_A
+ d_Delta_T_n_A_dot
+ d_epsilon_n_A
+ d_omega_n_A
+ d_M_n_A
and 7 more...
+ Glonass_Gnav_Almanac()
+ serialize()

**Public Member Functions**

- [Glonass\\_Gnav\\_Almanac](#) ()=default
- `template<class Archive>`  
void [serialize](#) (Archive &archive, const uint32\_t version)

*Serialize is a boost standard method to be called by the boost XML serialization. Here is used to save the almanac data on disk file.*

## Public Attributes

- double [d\\_n\\_A](#) {}  
*Conventional number of satellite within GLONASS space segment [dimensionless].*
- double [d\\_H\\_n\\_A](#) {}  
*Carrier frequency number of navigation RF signal transmitted by d\_nA satellite as table 4.10 (0-31) [dimensionless].*
- double [d\\_lambda\\_n\\_A](#) {}  
*Longitude of the first (within the d\_NA day) ascending node of d\_nA [radians].*
- double [d\\_t\\_lambda\\_n\\_A](#) {}  
*Time of first ascending node passage [s].*
- double [d\\_Delta\\_i\\_n\\_A](#) {}  
*Correction of the mean value of inclination of d\_n\_A satellite at instant t\_lambda\_n\_A [radians].*
- double [d\\_Delta\\_T\\_n\\_A](#) {}  
*Correction to the mean value of Draconian period of d\_n\_A satellite at instant t\_lambda\_n\_A [s / orbital period].*
- double [d\\_Delta\\_T\\_n\\_A\\_dot](#) {}  
*Rate of change of Draconian period of d\_n\_A satellite at instant t\_lambda\_n\_A [s / orbital period<sup>2</sup>].*
- double [d\\_epsilon\\_n\\_A](#) {}  
*Eccentricity of d\_n\_A satellite at instant t\_lambda\_n\_A [dimensionless].*
- double [d\\_omega\\_n\\_A](#) {}  
*Argument of perigee of d\_n\_A satellite at instant t\_lambda\_n\_A [radians].*
- double [d\\_M\\_n\\_A](#) {}  
*Type of satellite n\_A [dimensionless].*
- double [d\\_KP](#) {}  
*Notification on forthcoming leap second correction of UTC [dimensionless].*
- double [d\\_tau\\_n\\_A](#) {}  
*Coarse value of d\_n\_A satellite time correction to GLONASS time at instant t\_lambda\_n\_A[s].*
- bool [d\\_C\\_n](#) {}  
*Generalized “unhealthy flag” of n\_A satellite at instant of almanac upload [dimensionless].*
- bool [d\\_I\\_n](#) {}  
*Health flag for nth satellite; In = 0 indicates the n-th satellite is healthy, In = 1 indicates malfunction of this nth satellite [dimensionless].*
- int32\_t [i\\_satellite\\_freq\\_channel](#) {}  
*SV Frequency [Channel](#) Number.*
- uint32\_t [PRN](#) {}  
*SV PRN Number, equivalent to slot number for compatibility with GPS.*
- uint32\_t [i\\_satellite\\_slot\\_number](#) {}  
*SV Slot Number.*

### 10.154.1 Detailed Description

This class is a storage for the GLONASS SV ALMANAC data as described GLONASS ICD (Edition 5.1)

#### Note

Code added as part of GSoC 2017 program

#### See also

[GLONASS ICD](#)

Definition at line 37 of file [glonass\\_gnav\\_almanac.h](#).

## 10.154.2 Constructor & Destructor Documentation

### 10.154.2.1 Glonass\_Gnav\_Almanac()

Glonass\_Gnav\_Almanac::Glonass\_Gnav\_Almanac () [default]

Default constructor

## 10.154.3 Member Function Documentation

### 10.154.3.1 serialize()

```
template<class Archive>
void Glonass_Gnav_Almanac::serialize (
    Archive & archive,
    const uint32_t version) [inline]
```

Serialize is a boost standard method to be called by the boost XML serialization. Here is used to save the almanac data on disk file.

Definition at line 70 of file [glonass\\_gnav\\_almanac.h](#).

References [d\\_C\\_n](#), [d\\_Delta\\_i\\_n\\_A](#), [d\\_Delta\\_T\\_n\\_A](#), [d\\_Delta\\_T\\_n\\_A\\_dot](#), [d\\_epsilon\\_n\\_A](#), [d\\_H\\_n\\_A](#), [d\\_KP](#), [d\\_I\\_n](#), [d\\_lambda\\_n\\_A](#), [d\\_M\\_n\\_A](#), [d\\_n\\_A](#), [d\\_omega\\_n\\_A](#), [d\\_t\\_lambda\\_n\\_A](#), [d\\_tau\\_n\\_A](#), [i\\_satellite\\_freq\\_channel](#), [i\\_satellite\\_slot\\_number](#), and [PRN](#).

## 10.154.4 Member Data Documentation

### 10.154.4.1 d\_C\_n

bool Glonass\_Gnav\_Almanac::d\_C\_n {}

Generalized “unhealthy flag” of  $n_A$  satellite at instant of almanac upload [dimensionless].

Definition at line 57 of file [glonass\\_gnav\\_almanac.h](#).

Referenced by [serialize\(\)](#).

### 10.154.4.2 d\_Delta\_i\_n\_A

double Glonass\_Gnav\_Almanac::d\_Delta\_i\_n\_A {}

Correction of the mean value of inclination of  $d_n_A$  satellite at instant  $t_{\lambda n_A}$  [radians].

Definition at line 49 of file [glonass\\_gnav\\_almanac.h](#).

Referenced by [serialize\(\)](#).

### 10.154.4.3 d\_Delta\_T\_n\_A

double Glonass\_Gnav\_Almanac::d\_Delta\_T\_n\_A {}

Correction to the mean value of Draconian period of  $d_n_A$  satellite at instant  $t_{\lambda n_A}$  [s / orbital period].

Definition at line 50 of file [glonass\\_gnav\\_almanac.h](#).

Referenced by [serialize\(\)](#).

### 10.154.4.4 d\_Delta\_T\_n\_A\_dot

double Glonass\_Gnav\_Almanac::d\_Delta\_T\_n\_A\_dot {}

Rate of change of Draconian period of  $d_n_A$  satellite at instant  $t_{\lambda n_A}$  [s / orbital period<sup>2</sup>].

Definition at line 51 of file [glonass\\_gnav\\_almanac.h](#).

Referenced by [serialize\(\)](#).

### 10.154.4.5 d\_epsilon\_n\_A

double Glonass\_Gnav\_Almanac::d\_epsilon\_n\_A {}

Eccentricity of  $d_n_A$  satellite at instant  $t_{\lambda n_A}$  [dimensionless].

Definition at line 52 of file [glonass\\_gnav\\_almanac.h](#).

Referenced by [serialize\(\)](#).

#### 10.154.4.6 d\_H\_n\_A

```
double Glonass_Gnav_Almanac::d_H_n_A {}
```

Carrier frequency number of navigation RF signal transmitted by d\_nA satellite as table 4.10 (0-31) [dimensionless].

Definition at line 46 of file [glonass\\_gnav\\_almanac.h](#).

Referenced by [serialize\(\)](#).

#### 10.154.4.7 d\_KP

```
double Glonass_Gnav_Almanac::d_KP {}
```

Notification on forthcoming leap second correction of UTC [dimensionless].

Definition at line 55 of file [glonass\\_gnav\\_almanac.h](#).

Referenced by [serialize\(\)](#).

#### 10.154.4.8 d\_l\_n

```
bool Glonass_Gnav_Almanac::d_l_n {}
```

Health flag for nth satellite; ln = 0 indicates the n-th satellite is healthy, ln = 1 indicates malfunction of this nth satellite [dimensionless].

Definition at line 58 of file [glonass\\_gnav\\_almanac.h](#).

Referenced by [serialize\(\)](#).

#### 10.154.4.9 d\_lambda\_n\_A

```
double Glonass_Gnav_Almanac::d_lambda_n_A {}
```

Longitude of the first (within the d\_nA day) ascending node of d\_nA [radians].

Definition at line 47 of file [glonass\\_gnav\\_almanac.h](#).

Referenced by [serialize\(\)](#).

#### 10.154.4.10 d\_M\_n\_A

```
double Glonass_Gnav_Almanac::d_M_n_A {}
```

Type of satellite n\_A [dimensionless].

Definition at line 54 of file [glonass\\_gnav\\_almanac.h](#).

Referenced by [serialize\(\)](#).

#### 10.154.4.11 d\_n\_A

```
double Glonass_Gnav_Almanac::d_n_A {}
```

Conventional number of satellite within GLONASS space segment [dimensionless].

Definition at line 45 of file [glonass\\_gnav\\_almanac.h](#).

Referenced by [serialize\(\)](#).

#### 10.154.4.12 d\_omega\_n\_A

```
double Glonass_Gnav_Almanac::d_omega_n_A {}
```

Argument of perigee of d\_n\_A satellite at instant t\_lambdan\_A [radians].

Definition at line 53 of file [glonass\\_gnav\\_almanac.h](#).

Referenced by [serialize\(\)](#).

#### 10.154.4.13 d\_t\_lambda\_n\_A

```
double Glonass_Gnav_Almanac::d_t_lambda_n_A {}
```

Time of first ascending node passage [s].

Definition at line 48 of file [glonass\\_gnav\\_almanac.h](#).

Referenced by [serialize\(\)](#).

#### 10.154.4.14 d\_tau\_n\_A

```
double Glonass_Gnav_Almanac::d_tau_n_A {}
```

Coarse value of d\_n\_A satellite time correction to GLONASS time at instant t\_lambdan\_A[s].

Definition at line 56 of file [glonass\\_gnav\\_almanac.h](#).

Referenced by [serialize\(\)](#).

#### 10.154.4.15 i\_satellite\_freq\_channel

```
int32_t Glonass_Gnav_Almanac::i_satellite_freq_channel {}
```

SV Frequency [Channel](#) Number.

Definition at line 61 of file [glonass\\_gnav\\_almanac.h](#).

Referenced by [serialize\(\)](#).

#### 10.154.4.16 i\_satellite\_slot\_number

```
uint32_t Glonass_Gnav_Almanac::i_satellite_slot_number {}
```

SV Slot Number.

Definition at line 63 of file [glonass\\_gnav\\_almanac.h](#).

Referenced by [serialize\(\)](#).

#### 10.154.4.17 PRN

```
uint32_t Glonass_Gnav_Almanac::PRN {}
```

SV PRN Number, equivalent to slot number for compatibility with GPS.

Definition at line 62 of file [glonass\\_gnav\\_almanac.h](#).

Referenced by [serialize\(\)](#).

The documentation for this class was generated from the following file:

- [glonass\\_gnav\\_almanac.h](#)

### 10.155 Glonass\_Gnav\_Ephemeris Class Reference

This class is a storage and orbital model functions for the GLONASS SV ephemeris data as described in GLONASS ICD (Edition 5.1)

```
#include <glonass_gnav_ephemeris.h>
```



Collaboration diagram for Glonass\_Gnav\_Ephemeris:

Glonass_Gnav_Ephemeris
<ul style="list-style-type: none"> <li>+ d_m</li> <li>+ d_t_k</li> <li>+ d_t_b</li> <li>+ d_M</li> <li>+ d_gamma_n</li> <li>+ d_tau_n</li> <li>+ d_Xn</li> <li>+ d_Yn</li> <li>+ d_Zn</li> <li>+ d_VXn</li> <li>and 29 more...</li> </ul>
<ul style="list-style-type: none"> <li>+ Glonass_Gnav_Ephemeris()</li> <li>+ sv_clock_drift()</li> <li>+ compute_GLONASS_time()</li> <li>+ glot_to_utc()</li> <li>+ glot_to_gpst()</li> <li>+ serialize()</li> </ul>

## Public Member Functions

- [Glonass\\_Gnav\\_Ephemeris](#) ()=default
- double [sv\\_clock\\_drift](#) (double transmitTime, double timeCorrUTC)  
*Sets (d\_satClkDrift) and returns the clock drift in seconds according to the User Algorithm for SV Clock Correction.*
- boost::posix\_time::ptime [compute\\_GLONASS\\_time](#) (double offset\_time) const  
*Computes the GLONASS System Time and returns a boost::posix\_time::ptime object \ param offset\_time Is the start of day offset to compute the time.*
- boost::posix\_time::ptime [glot\\_to\\_utc](#) (double offset\_time, double glot2utc\_corr) const  
*Converts from GLONASST to UTC.*
- void [glot\\_to\\_gpst](#) (double tod\_offset, double glot2utc\_corr, double glot2gpst\_corr, int32\_t \*WN, double \*TOW) const  
*Converts from GLONASST to GPST.*
- template<class Archive>  
void [serialize](#) (Archive &archive, const uint32\_t version)  
*Serialize is a boost standard method to be called by the boost XML serialization. Here is used to save the ephemeris data on disk file.*

## Public Attributes

- double [d\\_m](#) {}  
*String number within frame [dimensionless].*
- double [d\\_t\\_k](#) {}

- GLONASS Time (UTC(SU) + 3 h) referenced to the beginning of the frame within the current day [s].
- double [d\\_t\\_b](#) {}  
Reference ephemeris relative time in GLONASS Time (UTC(SU) + 3 h). Index of a time interval within current day according to UTC(SU) + 03 hours 00 min. [s].
  - double [d\\_M](#) {}  
Type of satellite transmitting navigation signal [dimensionless].
  - double [d\\_gamma\\_n](#) {}  
Relative deviation of predicted carrier frequency value of n- satellite from nominal value at the instant  $t_b$  [dimensionless].
  - double [d\\_tau\\_n](#) {}  
Correction to the nth satellite time ( $t_n$ ) relative to GLONASS time ( $t_e$ ),.
  - double [d\\_Xn](#) {}  
Earth-fixed coordinate x of the satellite in PZ-90.02 coordinate system [km].
  - double [d\\_Yn](#) {}  
Earth-fixed coordinate y of the satellite in PZ-90.02 coordinate system [km].
  - double [d\\_Zn](#) {}  
Earth-fixed coordinate z of the satellite in PZ-90.02 coordinate system [km].
  - double [d\\_VXn](#) {}  
Earth-fixed velocity coordinate x of the satellite in PZ-90.02 coordinate system [km/s].
  - double [d\\_VYn](#) {}  
Earth-fixed velocity coordinate y of the satellite in PZ-90.02 coordinate system [km/s].
  - double [d\\_VZn](#) {}  
Earth-fixed velocity coordinate z of the satellite in PZ-90.02 coordinate system [km/s].
  - double [d\\_AXn](#) {}  
Earth-fixed acceleration coordinate x of the satellite in PZ-90.02 coordinate system [km/s<sup>2</sup>].
  - double [d\\_AYn](#) {}  
Earth-fixed acceleration coordinate y of the satellite in PZ-90.02 coordinate system [km/s<sup>2</sup>].
  - double [d\\_AZn](#) {}  
Earth-fixed acceleration coordinate z of the satellite in PZ-90.02 coordinate system [km/s<sup>2</sup>].
  - double [d\\_B\\_n](#) {}  
Health flag [dimensionless].
  - double [d\\_P](#) {}  
Technological parameter of control segment, indication the satellite operation mode in respect of time parameters [dimensionless].
  - double [d\\_N\\_T](#) {}  
Current date, calendar number of day within four-year interval starting from the 1-st of January in a leap year [days].
  - double [d\\_F\\_T](#) {}  
Parameter that provides the predicted satellite user range accuracy at time  $t_b$  [dimensionless].
  - double [d\\_n](#) {}  
Index of the satellite transmitting given navigation signal. It corresponds to a slot number within GLONASS constellation.
  - double [d\\_Delta\\_tau\\_n](#) {}  
Time difference between navigation RF signal transmitted in L2 sub- band and aviation RF signal transmitted in L1 sub-band by nth satellite. [dimensionless].
  - double [d\\_E\\_n](#) {}  
Characterises "age" of a current information [days].
  - double [d\\_P\\_1](#) {}  
Flag of the immediate data updating [minutes].
  - bool [d\\_P\\_2](#) {}  
Flag of oddness ("1") or evenness ("0") of the value of ( $t_b$ ) [dimensionless].
  - bool [d\\_P\\_3](#) {}

Flag indicating a number of satellites for which almanac is transmitted within given frame: "1" corresponds to 5 satellites and "0" corresponds to 4 satellites [dimensionless].

- bool [d\\_P\\_4](#) {}

Flag to show that ephemeris parameters are present. "1" indicates that updated ephemeris or frequency/time parameters have been uploaded by the control segment [dimensionless].

- bool [d\\_l3rd\\_n](#) {}

Health flag for nth satellite; In = 0 indicates the n-th satellite is healthy, In = 1 indicates malfunction of this nth satellite [dimensionless].

- bool [d\\_l5th\\_n](#) {}

Health flag for nth satellite; In = 0 indicates the n-th satellite is healthy, In = 1 indicates malfunction of this nth satellite [dimensionless].

- int32\_t [i\\_satellite\\_freq\\_channel](#) {}

SV Frequency [Channel](#) Number.

- uint32\_t [PRN](#) {}

SV PRN Number, equivalent to slot number for compatibility with GPS.

- uint32\_t [i\\_satellite\\_slot\\_number](#) {}

SV Slot Number.

- double [d\\_yr](#) = 1972.0

Current year.

- double [d\\_satClkDrift](#) {}

GLONASS clock error.

- double [d\\_dtr](#) {}

relativistic clock correction term

- double [d\\_iodb](#) {}

Issue of data, ephemeris (Bit 0-6 of tb)

- double [d\\_tau\\_c](#) {}

GLONASS 2 UTC correction (todo) may be eliminated.

- double [d\\_TOW](#) {}

GLONASS IN GPST seconds of week.

- int32\_t [d\\_WN](#) {}

GLONASS IN GPST week number of the start of frame.

- double [d\\_tod](#) {}

Time of Day since ephemeris where decoded.

### 10.155.1 Detailed Description

This class is a storage and orbital model functions for the GLONASS SV ephemeris data as described in GLONASS ICD (Edition 5.1)

#### Note

Code added as part of GSoC 2017 program

#### See also

[GLONASS ICD](#)

Definition at line 40 of file [glonass\\_gnav\\_ephemeris.h](#).

### 10.155.2 Constructor & Destructor Documentation

#### 10.155.2.1 Glonass\_Gnav\_Ephemeris()

```
Glonass_Gnav_Ephemeris::Glonass_Gnav_Ephemeris () [default]
```

Default constructor

### 10.155.3 Member Function Documentation

#### 10.155.3.1 compute\_GLONASS\_time()

```
boost::posix_time::ptime Glonass_Gnav_Ephemeris::compute_GLONASS_time (
    double offset_time) const
```

Computes the GLONASS System Time and returns a boost::posix\_time::ptime object \ param offset\_time Is the start of day offset to compute the time.

#### 10.155.3.2 glot\_to\_gpst()

```
void Glonass_Gnav_Ephemeris::glot_to_gpst (
    double tod_offset,
    double glot2utc_corr,
    double glot2gpst_corr,
    int32_t * WN,
    double * TOW) const
```

Converts from GLONASST to GPST.

Converts from GLONASST to GPST in time of week (TOW) and week number (WN) format

##### Parameters

in	<i>tod_offset</i>	Is the start of day offset
in	<i>glot2utc_corr</i>	Correction from GLONASST to UTC
in	<i>glot2gpst_corr</i>	Correction from GLONASST to GPST
out	<i>WN</i>	Week Number, not in mod(1024) format
out	<i>TOW</i>	Time of Week in seconds of week

#### 10.155.3.3 glot\_to\_utc()

```
boost::posix_time::ptime Glonass_Gnav_Ephemeris::glot_to_utc (
    double offset_time,
    double glot2utc_corr) const
```

Converts from GLONASST to UTC.

The function simply adjust for the 6 hrs offset between GLONASST and UTC

##### Parameters

in	<i>offset_time</i>	Is the start of day offset
in	<i>glot2utc_corr</i>	Correction from GLONASST to UTC

##### Returns

UTC time as a boost::posix\_time::ptime object

#### 10.155.3.4 serialize()

```
template<class Archive>
void Glonass_Gnav_Ephemeris::serialize (
    Archive & archive,
    const uint32_t version) [inline]
```

Serialize is a boost standard method to be called by the boost XML serialization. Here is used to save the ephemeris data on disk file.

< SV PRN frequency channel number

< String number within frame [dimensionless]

< Time referenced to the beginning of the frame within the current day [hours, minutes, seconds]

< Index of a time interval within current day according to UTC(SU) + 03 hours 00 min. [minutes]  
 < Type of satellite transmitting navigation signal [dimensionless]  
 < Relative deviation of predicted carrier frequency value of n- satellite from nominal value at the instant tb [dimensionless]  
 < Correction to the nth satellite time (tn) relative to GLONASS time (te)  
 < Earth-fixed coordinate x of the satellite in PZ-90.02 coordinate system [km].  
 < Earth-fixed coordinate y of the satellite in PZ-90.02 coordinate system [km]  
 < Earth-fixed coordinate z of the satellite in PZ-90.02 coordinate system [km]  
 < Earth-fixed velocity coordinate x of the satellite in PZ-90.02 coordinate system [km/s]  
 < Earth-fixed velocity coordinate y of the satellite in PZ-90.02 coordinate system [km/s]  
 < Earth-fixed velocity coordinate z of the satellite in PZ-90.02 coordinate system [km/s]  
 < Earth-fixed acceleration coordinate x of the satellite in PZ-90.02 coordinate system [km/s<sup>2</sup>]  
 < Earth-fixed acceleration coordinate y of the satellite in PZ-90.02 coordinate system [km/s<sup>2</sup>]  
 < Earth-fixed acceleration coordinate z of the satellite in PZ-90.02 coordinate system [km/s<sup>2</sup>]  
 < Health flag [dimensionless]  
 < Technological parameter of control segment, indication the satellite operation mode in respect of time parameters [dimensionless]  
 < Current date, calendar number of day within four-year interval starting from the 1-st of January in a leap year [days]  
 < Parameter that provides the predicted satellite user range accuracy at time tb [dimensionless]  
 < Index of the satellite transmitting given navigation signal. It corresponds to a slot number within GLONASS constellation  
 < Time difference between navigation RF signal transmitted in L2 sub- band and aviation RF signal transmitted in L1 sub-band by nth satellite. [dimensionless]  
 < Characterises "age" of a current information [days]  
 < Flag of the immediate data updating.  
 < Flag of oddness ("1") or evenness ("0") of the value of (tb) [dimensionless]  
 < Flag indicating a number of satellites for which almanac is transmitted within given frame: "1" corresponds to 5 satellites and "0" corresponds to 4 satellites [dimensionless]  
 < Flag to show that ephemeris parameters are present. "1" indicates that updated ephemeris or frequency/time parameters have been uploaded by the control segment [dimensionless]  
 < Health flag for nth satellite; In = 0 indicates the n-th satellite is helthy, In = 1 indicates malfunction of this nth satellite [dimensionless]  
 < Health flag for nth satellite; In = 0 indicates the n-th satellite is helthy, In = 1 indicates malfunction of this nth satellite [dimensionless]  
 Definition at line 128 of file [glonass\\_gnav\\_ephemeris.h](#).  
 References [d\\_AXn](#), [d\\_AYn](#), [d\\_AZn](#), [d\\_B\\_n](#), [d\\_Delta\\_tau\\_n](#), [d\\_E\\_n](#), [d\\_F\\_T](#), [d\\_gamma\\_n](#), [d\\_I3rd\\_n](#), [d\\_I5th\\_n](#), [d\\_M](#), [d\\_m](#), [d\\_n](#), [d\\_N\\_T](#), [d\\_P](#), [d\\_P\\_1](#), [d\\_P\\_2](#), [d\\_P\\_3](#), [d\\_P\\_4](#), [d\\_t\\_b](#), [d\\_t\\_k](#), [d\\_tau\\_n](#), [d\\_VXn](#), [d\\_VYn](#), [d\\_VZn](#), [d\\_Xn](#), [d\\_Yn](#), [d\\_Zn](#), [i\\_satellite\\_freq\\_channel](#), [i\\_satellite\\_slot\\_number](#), and [PRN](#).

### 10.155.3.5 sv\_clock\_drift()

```
double Glonass_Gnav_Ephemeris::sv_clock_drift (
    double transmitTime,
    double timeCorrUTC)
```

Sets (*d\_satClkDrift*) and returns the clock drift in seconds according to the User Algorithm for SV Clock Correction.

## 10.155.4 Member Data Documentation

### 10.155.4.1 d\_AXn

```
double Glonass_Gnav_Ephemeris::d_AXn {}
```

Earth-fixed acceleration coordinate x of the satellite in PZ-90.02 coordinate system [km/s<sup>2</sup>].

Definition at line 60 of file [glonass\\_gnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

### 10.155.4.2 d\_AYn

```
double Glonass_Gnav_Ephemeris::d_AYn {}
```

Earth-fixed acceleration coordinate y of the satellite in PZ-90.02 coordinate system [km/s<sup>2</sup>].  
Definition at line 61 of file [glonass\\_gnav\\_ephemeris.h](#).  
Referenced by [serialize\(\)](#).

#### 10.155.4.3 d\_AZn

```
double Glonass_Gnav_Ephemeris::d_AZn {}
```

Earth-fixed acceleration coordinate z of the satellite in PZ-90.02 coordinate system [km/s<sup>2</sup>].  
Definition at line 62 of file [glonass\\_gnav\\_ephemeris.h](#).  
Referenced by [serialize\(\)](#).

#### 10.155.4.4 d\_B\_n

```
double Glonass_Gnav_Ephemeris::d_B_n {}
```

Health flag [dimensionless].  
Definition at line 63 of file [glonass\\_gnav\\_ephemeris.h](#).  
Referenced by [serialize\(\)](#).

#### 10.155.4.5 d\_Delta\_tau\_n

```
double Glonass_Gnav_Ephemeris::d_Delta_tau_n {}
```

Time difference between navigation RF signal transmitted in L2 sub- band and aviation RF signal transmitted in L1 sub-band by nth satellite. [dimensionless].  
Definition at line 68 of file [glonass\\_gnav\\_ephemeris.h](#).  
Referenced by [serialize\(\)](#).

#### 10.155.4.6 d\_dtr

```
double Glonass_Gnav_Ephemeris::d_dtr {}
```

relativistic clock correction term  
Definition at line 84 of file [glonass\\_gnav\\_ephemeris.h](#).

#### 10.155.4.7 d\_E\_n

```
double Glonass_Gnav_Ephemeris::d_E_n {}
```

Characterises "age" of a current information [days].  
Definition at line 69 of file [glonass\\_gnav\\_ephemeris.h](#).  
Referenced by [serialize\(\)](#).

#### 10.155.4.8 d\_F\_T

```
double Glonass_Gnav_Ephemeris::d_F_T {}
```

Parameter that provides the predicted satellite user range accuracy at time tb [dimensionless].  
Definition at line 66 of file [glonass\\_gnav\\_ephemeris.h](#).  
Referenced by [serialize\(\)](#).

#### 10.155.4.9 d\_gamma\_n

```
double Glonass_Gnav_Ephemeris::d_gamma_n {}
```

Relative deviation of predicted carrier frequency value of n- satellite from nominal value at the instant tb [dimensionless].  
Definition at line 52 of file [glonass\\_gnav\\_ephemeris.h](#).  
Referenced by [serialize\(\)](#).

#### 10.155.4.10 d\_iode

```
double Glonass_Gnav_Ephemeris::d_iode {}
```

Issue of data, ephemeris (Bit 0-6 of tb)  
Definition at line 85 of file [glonass\\_gnav\\_ephemeris.h](#).

#### 10.155.4.11 d\_l3rd\_n

```
bool Glonass_Gnav_Ephemeris::d_l3rd_n {}
```

Health flag for nth satellite; In = 0 indicates the n-th satellite is healthy, In = 1 indicates malfunction of this nth satellite [dimensionless].

Definition at line 74 of file [glonass\\_gnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.155.4.12 d\_l5th\_n

```
bool Glonass_Gnav_Ephemeris::d_l5th_n {}
```

Health flag for nth satellite; In = 0 indicates the n-th satellite is healthy, In = 1 indicates malfunction of this nth satellite [dimensionless].

Definition at line 75 of file [glonass\\_gnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.155.4.13 d\_M

```
double Glonass_Gnav_Ephemeris::d_M {}
```

Type of satellite transmitting navigation signal [dimensionless].

Definition at line 51 of file [glonass\\_gnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.155.4.14 d\_m

```
double Glonass_Gnav_Ephemeris::d_m {}
```

String number within frame [dimensionless].

Definition at line 48 of file [glonass\\_gnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.155.4.15 d\_n

```
double Glonass_Gnav_Ephemeris::d_n {}
```

Index of the satellite transmitting given navigation signal. It corresponds to a slot number within GLONASS constellation.

Definition at line 67 of file [glonass\\_gnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.155.4.16 d\_N\_T

```
double Glonass_Gnav_Ephemeris::d_N_T {}
```

Current date, calendar number of day within four-year interval starting from the 1-st of January in a leap year [days].

Definition at line 65 of file [glonass\\_gnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.155.4.17 d\_P

```
double Glonass_Gnav_Ephemeris::d_P {}
```

Technological parameter of control segment, indication the satellite operation mode in respect of time parameters [dimensionless].

Definition at line 64 of file [glonass\\_gnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.155.4.18 d\_P\_1

```
double Glonass_Gnav_Ephemeris::d_P_1 {}
```

Flag of the immediate data updating [minutes].

Definition at line 70 of file [glonass\\_gnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.155.4.19 d\_P\_2

```
bool Glonass_Gnav_Ephemeris::d_P_2 {}
```

Flag of oddness ("1") or evenness ("0") of the value of (tb) [dimensionless].

Definition at line 71 of file [glonass\\_gnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.155.4.20 d\_P\_3

```
bool Glonass_Gnav_Ephemeris::d_P_3 {}
```

Flag indicating a number of satellites for which almanac is transmitted within given frame: "1" corresponds to 5 satellites and "0" corresponds to 4 satellites [dimensionless].

Definition at line 72 of file [glonass\\_gnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.155.4.21 d\_P\_4

```
bool Glonass_Gnav_Ephemeris::d_P_4 {}
```

Flag to show that ephemeris parameters are present. "1" indicates that updated ephemeris or frequency/time parameters have been uploaded by the control segment [dimensionless].

Definition at line 73 of file [glonass\\_gnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.155.4.22 d\_satClkDrift

```
double Glonass_Gnav_Ephemeris::d_satClkDrift {}
```

GLONASS clock error.

Definition at line 83 of file [glonass\\_gnav\\_ephemeris.h](#).

#### 10.155.4.23 d\_t\_b

```
double Glonass_Gnav_Ephemeris::d_t_b {}
```

Reference ephemeris relative time in GLONASS Time (UTC(SU) + 3 h). Index of a time interval within current day according to UTC(SU) + 03 hours 00 min. [s].

Definition at line 50 of file [glonass\\_gnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.155.4.24 d\_t\_k

```
double Glonass_Gnav_Ephemeris::d_t_k {}
```

GLONASS Time (UTC(SU) + 3 h) referenced to the beginning of the frame within the current day [s].

Definition at line 49 of file [glonass\\_gnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.155.4.25 d\_tau\_c

```
double Glonass_Gnav_Ephemeris::d_tau_c {}
```

GLONASS 2 UTC correction (todo) may be eliminated.

Definition at line 86 of file [glonass\\_gnav\\_ephemeris.h](#).

#### 10.155.4.26 d\_tau\_n

```
double Glonass_Gnav_Ephemeris::d_tau_n {}
```

Correction to the nth satellite time (tn) relative to GLONASS time (te),.

Definition at line 53 of file [glonass\\_gnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).



**10.155.4.27 d\_tod**

```
double Glonass_Gnav_Ephemeris::d_tod {}
```

Time of Day since ephemeris where decoded.

Definition at line 89 of file [glonass\\_gnav\\_ephemeris.h](#).

**10.155.4.28 d\_TOW**

```
double Glonass_Gnav_Ephemeris::d_TOW {}
```

GLONASST IN GPST seconds of week.

Definition at line 87 of file [glonass\\_gnav\\_ephemeris.h](#).

**10.155.4.29 d\_VXn**

```
double Glonass_Gnav_Ephemeris::d_VXn {}
```

Earth-fixed velocity coordinate x of the satellite in PZ-90.02 coordinate system [km/s].

Definition at line 57 of file [glonass\\_gnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

**10.155.4.30 d\_VYn**

```
double Glonass_Gnav_Ephemeris::d_VYn {}
```

Earth-fixed velocity coordinate y of the satellite in PZ-90.02 coordinate system [km/s].

Definition at line 58 of file [glonass\\_gnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

**10.155.4.31 d\_VZn**

```
double Glonass_Gnav_Ephemeris::d_VZn {}
```

Earth-fixed velocity coordinate z of the satellite in PZ-90.02 coordinate system [km/s].

Definition at line 59 of file [glonass\\_gnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

**10.155.4.32 d\_WN**

```
int32_t Glonass_Gnav_Ephemeris::d_WN {}
```

GLONASST IN GPST week number of the start of frame.

Definition at line 88 of file [glonass\\_gnav\\_ephemeris.h](#).

**10.155.4.33 d\_Xn**

```
double Glonass_Gnav_Ephemeris::d_Xn {}
```

Earth-fixed coordinate x of the satellite in PZ-90.02 coordinate system [km].

Definition at line 54 of file [glonass\\_gnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

**10.155.4.34 d\_Yn**

```
double Glonass_Gnav_Ephemeris::d_Yn {}
```

Earth-fixed coordinate y of the satellite in PZ-90.02 coordinate system [km].

Definition at line 55 of file [glonass\\_gnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

**10.155.4.35 d\_yr**

```
double Glonass_Gnav_Ephemeris::d_yr = 1972.0
```

Current year.

Definition at line 82 of file [glonass\\_gnav\\_ephemeris.h](#).

#### 10.155.4.36 d\_Zn

```
double Glonass_Gnav_Ephemeris::d_Zn {}
```

Earth-fixed coordinate z of the satellite in PZ-90.02 coordinate system [km].

Definition at line 56 of file [glonass\\_gnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.155.4.37 i\_satellite\_freq\_channel

```
int32_t Glonass_Gnav_Ephemeris::i_satellite_freq_channel {}
```

SV Frequency [Channel](#) Number.

Definition at line 79 of file [glonass\\_gnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.155.4.38 i\_satellite\_slot\_number

```
uint32_t Glonass_Gnav_Ephemeris::i_satellite_slot_number {}
```

SV Slot Number.

Definition at line 81 of file [glonass\\_gnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.155.4.39 PRN

```
uint32_t Glonass_Gnav_Ephemeris::PRN {}
```

SV PRN Number, equivalent to slot number for compatibility with GPS.

Definition at line 80 of file [glonass\\_gnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

The documentation for this class was generated from the following file:

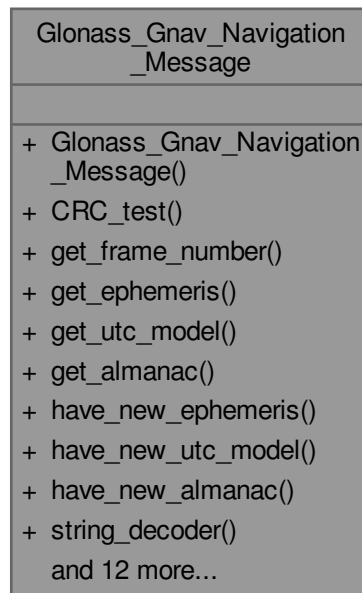
- [glonass\\_gnav\\_ephemeris.h](#)

## 10.156 Glonass\_Gnav\_Navigation\_Message Class Reference

This class decodes a GLONASS GNAV Data message as described in GLONASS ICD (Edition 5.1)

```
#include <glonass_gnav_navigation_message.h>
```

Collaboration diagram for Glonass\_Gnav\_Navigation\_Message:



## Public Member Functions

- [Glonass\\_Gnav\\_Navigation\\_Message](#) ()
- bool [CRC\\_test](#) (std::bitset< [GLONASS\\_GNAV\\_STRING\\_BITS](#) > &bits) const  
*Compute CRC for GLONASS GNAV strings.*
- uint32\_t [get\\_frame\\_number](#) (uint32\_t satellite\_slot\_number)  
*Computes the frame number being decoded given the satellite slot number.*
- [Glonass\\_Gnav\\_Ephemeris](#) [get\\_ephemeris](#) () const  
*Obtain a GLONASS GNAV SV Ephemeris class filled with current SV data.*
- [Glonass\\_Gnav\\_Utc\\_Model](#) [get\\_utc\\_model](#) () const  
*Obtain a GLONASS GNAV UTC model parameters class filled with current SV data.*
- [Glonass\\_Gnav\\_Almanac](#) [get\\_almanac](#) (uint32\_t satellite\_slot\_number) const  
*Returns a [Glonass\\_Gnav\\_Almanac](#) object filled with the latest navigation data received.*
- bool [have\\_new\\_ephemeris](#) ()  
*Returns true if a new [Glonass\\_Gnav\\_Ephemeris](#) object has arrived.*
- bool [have\\_new\\_utc\\_model](#) ()  
*Returns true if new [Glonass\\_Gnav\\_Utc\\_Model](#) object has arrived.*
- bool [have\\_new\\_almanac](#) ()  
*Returns true if new [Glonass\\_Gnav\\_Almanac](#) object has arrived.*
- int32\_t [string\\_decoder](#) (const std::string &frame\_string)  
*Decodes the GLONASS GNAV string.*
- bool [get\\_flag\\_CRC\\_test](#) () const
- void [set\\_rf\\_link](#) (int32\_t rf\_link)
- uint32\_t [get\\_alm\\_satellite\\_slot\\_number](#) () const
- bool [get\\_flag\\_update\\_slot\\_number](#) () const
- void [set\\_flag\\_update\\_slot\\_number](#) (bool flag\_slot)

- bool [get\\_flag\\_TOW\\_new](#) () const
- void [set\\_flag\\_TOW\\_new](#) (bool tow\_new)
- bool [is\\_flag\\_TOW\\_set](#) () const
- void [set\\_flag\\_ephemeris\\_str\\_1](#) (bool ephemeris\_str\_1)
- void [set\\_flag\\_ephemeris\\_str\\_2](#) (bool ephemeris\_str\_2)
- void [set\\_flag\\_ephemeris\\_str\\_3](#) (bool ephemeris\_str\_3)
- void [set\\_flag\\_ephemeris\\_str\\_4](#) (bool ephemeris\_str\_4)

### 10.156.1 Detailed Description

This class decodes a GLONASS GNAV Data message as described in GLONASS ICD (Edition 5.1)

#### Note

Code added as part of GSoC 2017 program

#### See also

[GLONASS ICD](#)

Definition at line 46 of file [glonass\\_gnav\\_navigation\\_message.h](#).

### 10.156.2 Constructor & Destructor Documentation

#### 10.156.2.1 Glonass\_Gnav\_Navigation\_Message()

```
Glonass_Gnav_Navigation_Message::Glonass_Gnav_Navigation_Message ()
```

Default constructor

### 10.156.3 Member Function Documentation

#### 10.156.3.1 CRC\_test()

```
bool Glonass_Gnav_Navigation_Message::CRC_test (
    std::bitset< GLONASS\_GNAV\_STRING\_BITS > & bits) const
```

Compute CRC for GLONASS GNAV strings.

#### Parameters

<i>bits</i>	Bits of the string message where to compute CRC
-------------	---

#### 10.156.3.2 get\_alm\_satellite\_slot\_number()

```
uint32_t Glonass_Gnav_Navigation_Message::get_alm_satellite_slot_number () const [inline]
```

Definition at line 122 of file [glonass\\_gnav\\_navigation\\_message.h](#).

#### 10.156.3.3 get\_almanac()

```
Glonass\_Gnav\_Almanac Glonass_Gnav_Navigation_Message::get_almanac (
    uint32_t satellite_slot_number) const
```

Returns a [Glonass\\_Gnav\\_Almanac](#) object filled with the latest navigation data received.

#### Parameters

<i>satellite_slot_number</i>	Slot number identifier for the satellite
------------------------------	--

#### Returns

Returns the [Glonass\\_Gnav\\_Almanac](#) object for the input slot number

**10.156.3.4 get\_ephemeris()**

[Glonass\\_Gnav\\_Ephemeris](#) `Glonass_Gnav_Navigation_Message::get_ephemeris () const [inline]`

Obtain a GLONASS GNAV SV Ephemeris class filled with current SV data.

Definition at line 70 of file [glonass\\_gnav\\_navigation\\_message.h](#).

**10.156.3.5 get\_flag\_CRC\_test()**

`bool Glonass_Gnav_Navigation_Message::get_flag_CRC_test () const [inline]`

Definition at line 112 of file [glonass\\_gnav\\_navigation\\_message.h](#).

**10.156.3.6 get\_flag\_TOW\_new()**

`bool Glonass_Gnav_Navigation_Message::get_flag_TOW_new () const [inline]`

Definition at line 137 of file [glonass\\_gnav\\_navigation\\_message.h](#).

**10.156.3.7 get\_flag\_update\_slot\_number()**

`bool Glonass_Gnav_Navigation_Message::get_flag_update_slot_number () const [inline]`

Definition at line 127 of file [glonass\\_gnav\\_navigation\\_message.h](#).

**10.156.3.8 get\_frame\_number()**

`uint32_t Glonass_Gnav_Navigation_Message::get_frame_number (`  
     `uint32_t satellite_slot_number)`

Computes the frame number being decoded given the satellite slot number.

**Parameters**

<i>satellite_slot_number</i>	[in] Satellite slot number identifier
------------------------------	---------------------------------------

**Returns**

Frame number being decoded, 0 if operation was not successful.

**10.156.3.9 get\_utc\_model()**

[Glonass\\_Gnav\\_Utc\\_Model](#) `Glonass_Gnav_Navigation_Message::get_utc_model () const [inline]`

Obtain a GLONASS GNAV UTC model parameters class filled with current SV data.

Definition at line 78 of file [glonass\\_gnav\\_navigation\\_message.h](#).

**10.156.3.10 have\_new\_almanac()**

`bool Glonass_Gnav_Navigation_Message::have_new_almanac ()`

Returns true if new [Glonass\\_Gnav\\_Almanac](#) object has arrived.

**10.156.3.11 have\_new\_ephemeris()**

`bool Glonass_Gnav_Navigation_Message::have_new_ephemeris ()`

Returns true if a new [Glonass\\_Gnav\\_Ephemeris](#) object has arrived.

**10.156.3.12 have\_new\_utc\_model()**

`bool Glonass_Gnav_Navigation_Message::have_new_utc_model ()`

Returns true if new [Glonass\\_Gnav\\_Utc\\_Model](#) object has arrived.

**10.156.3.13 is\_flag\_TOW\_set()**

`bool Glonass_Gnav_Navigation_Message::is_flag_TOW_set () const [inline]`

Definition at line 147 of file [glonass\\_gnav\\_navigation\\_message.h](#).

**10.156.3.14 set\_flag\_ephemeris\_str\_1()**

```
void Glonass_Gnav_Navigation_Message::set_flag_ephemeris_str_1 (
    bool ephemeris_str_1) [inline]
```

Definition at line 152 of file [glonass\\_gnav\\_navigation\\_message.h](#).

**10.156.3.15 set\_flag\_ephemeris\_str\_2()**

```
void Glonass_Gnav_Navigation_Message::set_flag_ephemeris_str_2 (
    bool ephemeris_str_2) [inline]
```

Definition at line 157 of file [glonass\\_gnav\\_navigation\\_message.h](#).

**10.156.3.16 set\_flag\_ephemeris\_str\_3()**

```
void Glonass_Gnav_Navigation_Message::set_flag_ephemeris_str_3 (
    bool ephemeris_str_3) [inline]
```

Definition at line 162 of file [glonass\\_gnav\\_navigation\\_message.h](#).

**10.156.3.17 set\_flag\_ephemeris\_str\_4()**

```
void Glonass_Gnav_Navigation_Message::set_flag_ephemeris_str_4 (
    bool ephemeris_str_4) [inline]
```

Definition at line 167 of file [glonass\\_gnav\\_navigation\\_message.h](#).

**10.156.3.18 set\_flag\_TOW\_new()**

```
void Glonass_Gnav_Navigation_Message::set_flag_TOW_new (
    bool tow_new) [inline]
```

Definition at line 142 of file [glonass\\_gnav\\_navigation\\_message.h](#).

**10.156.3.19 set\_flag\_update\_slot\_number()**

```
void Glonass_Gnav_Navigation_Message::set_flag_update_slot_number (
    bool flag_slot) [inline]
```

Definition at line 132 of file [glonass\\_gnav\\_navigation\\_message.h](#).

**10.156.3.20 set\_rf\_link()**

```
void Glonass_Gnav_Navigation_Message::set_rf_link (
    int32_t rf_link) [inline]
```

Definition at line 117 of file [glonass\\_gnav\\_navigation\\_message.h](#).

**10.156.3.21 string\_decoder()**

```
int32_t Glonass_Gnav_Navigation_Message::string_decoder (
    const std::string & frame_string)
```

Decodes the GLONASS GNAV string.

**Parameters**

<i>frame_string</i>	[in] is the string message within the parsed frame
---------------------	--

**Returns**

Returns the ID of the decoded string

The documentation for this class was generated from the following file:

- [glonass\\_gnav\\_navigation\\_message.h](#)

## 10.157 Glonass\_Gnav\_Utc\_Model Class Reference

This class is a storage for the GLONASS GNAV UTC MODEL data as described in GLONASS ICD (Edition 5.1)

```
#include <glonass_gnav_utc_model.h>
```

Collaboration diagram for Glonass\_Gnav\_Utc\_Model:

Glonass_Gnav_Utc_Model
+ valid
+ d_tau_c
+ d_tau_gps
+ d_N_4
+ d_N_A
+ d_B1
+ d_B2
+ Glonass_Gnav_Utc_Model()
+ utc_time()
+ serialize()

### Public Member Functions

- [Glonass\\_Gnav\\_Utc\\_Model](#) ()=default
- double [utc\\_time](#) (double glonass\_time\_corrected) const  
*Computes the Coordinated Universal Time (UTC) and returns it in [s] (GLONASS ICD (Edition 5.1) Section 3.3.3 GLONASS Time)*
- template<class Archive>  
void [serialize](#) (Archive &archive, const uint32\_t version)  
*Serialize is a boost standard method to be called by the boost XML serialization. Here is used to save the almanac data on disk file.*

### Public Attributes

- bool [valid](#) {}
- double [d\\_tau\\_c](#) {}  
*GLONASS time scale correction to UTC(SU) time. [s].*
- double [d\\_tau\\_gps](#) {}  
*Correction to GPS time to GLONASS time [day].*
- double [d\\_N\\_4](#) {}  
*Four year interval number starting from 1996 [4 year interval].*
- double [d\\_N\\_A](#) {}  
*Calendar day number within the four-year period beginning since the leap year for Almanac data [days].*
- double [d\\_B1](#) {}  
*Coefficient to determine DeltaUT1 [s].*
- double [d\\_B2](#) {}  
*Coefficient to determine DeltaUT1 [s/msd].*

### 10.157.1 Detailed Description

This class is a storage for the GLONASS GNAV UTC MODEL data as described in GLONASS ICD (Edition 5.1)

#### Note

Code added as part of GSoC 2017 program

#### See also

[GLONASS ICD](#)

Definition at line 37 of file [glonass\\_gnav\\_utc\\_model.h](#).

### 10.157.2 Constructor & Destructor Documentation

#### 10.157.2.1 Glonass\_Gnav\_Utc\_Model()

```
Glonass_Gnav_Utc_Model::Glonass_Gnav_Utc_Model () [default]
```

Default constructor

### 10.157.3 Member Function Documentation

#### 10.157.3.1 serialize()

```
template<class Archive>
void Glonass_Gnav_Utc_Model::serialize (
    Archive & archive,
    const uint32_t version) [inline]
```

Serialize is a boost standard method to be called by the boost XML serialization. Here is used to save the almanac data on disk file.

Definition at line 65 of file [glonass\\_gnav\\_utc\\_model.h](#).

References [d\\_B1](#), [d\\_B2](#), [d\\_N\\_4](#), [d\\_N\\_A](#), [d\\_tau\\_c](#), and [d\\_tau\\_gps](#).

#### 10.157.3.2 utc\_time()

```
double Glonass_Gnav_Utc_Model::utc_time (
    double glonass_time_corrected) const
```

Computes the Coordinated Universal Time (UTC) and returns it in [s] (GLONASS ICD (Edition 5.1) Section 3.3.3 GLONASS Time)

### 10.157.4 Member Data Documentation

#### 10.157.4.1 d\_B1

```
double Glonass_Gnav_Utc_Model::d_B1 {}
```

Coefficient to determine DeltaUT1 [s].

Definition at line 51 of file [glonass\\_gnav\\_utc\\_model.h](#).

Referenced by [serialize\(\)](#).

#### 10.157.4.2 d\_B2

```
double Glonass_Gnav_Utc_Model::d_B2 {}
```

Coefficient to determine DeltaUT1 [s/msd].

Definition at line 52 of file [glonass\\_gnav\\_utc\\_model.h](#).

Referenced by [serialize\(\)](#).

#### 10.157.4.3 d\_N\_4

```
double Glonass_Gnav_Utc_Model::d_N_4 {}
```

Four year interval number starting from 1996 [4 year interval].

Definition at line 49 of file [glonass\\_gnav\\_utc\\_model.h](#).

Referenced by [serialize\(\)](#).



#### 10.157.4.4 d\_N\_A

```
double Glonass_Gnav_Utc_Model::d_N_A {}
```

Calendar day number within the four-year period beginning since the leap year for Almanac data [days].

Definition at line 50 of file [glonass\\_gnav\\_utc\\_model.h](#).

Referenced by [serialize\(\)](#).

#### 10.157.4.5 d\_tau\_c

```
double Glonass_Gnav_Utc_Model::d_tau_c {}
```

GLONASS time scale correction to UTC(SU) time. [s].

Definition at line 47 of file [glonass\\_gnav\\_utc\\_model.h](#).

Referenced by [serialize\(\)](#).

#### 10.157.4.6 d\_tau\_gps

```
double Glonass_Gnav_Utc_Model::d_tau_gps {}
```

Correction to GPS time to GLONASS time [day].

Definition at line 48 of file [glonass\\_gnav\\_utc\\_model.h](#).

Referenced by [serialize\(\)](#).

#### 10.157.4.7 valid

```
bool Glonass_Gnav_Utc_Model::valid {}
```

Definition at line 45 of file [glonass\\_gnav\\_utc\\_model.h](#).

The documentation for this class was generated from the following file:

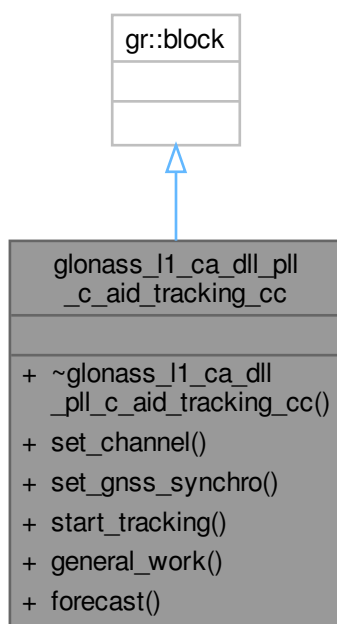
- [glonass\\_gnav\\_utc\\_model.h](#)

## 10.158 glonass\_l1\_ca\_dll\_pll\_c\_aid\_tracking\_cc Class Reference

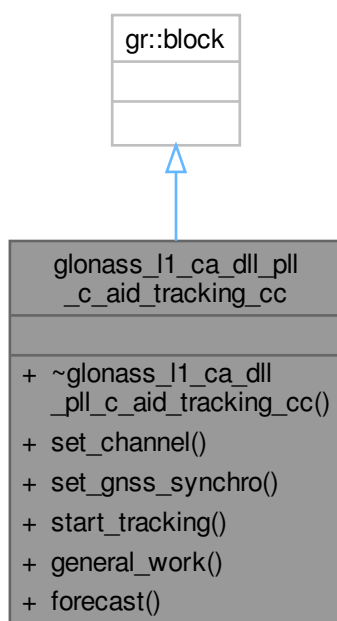
This class implements a DLL + PLL tracking loop block.

```
#include <glonass_l1_ca_dll_pll_c_aid_tracking_cc.h>
```

Inheritance diagram for `glonass_l1_ca_dll_pll_c_aid_tracking_cc`:



Collaboration diagram for `glonass_l1_ca_dll_pll_c_aid_tracking_cc`:



**Public Member Functions**

- void **set\_channel** (uint32\_t channel)
- void **set\_gnss\_synchro** ([Gnss\\_Synchro](#) \*p\_gnss\_synchro)
- void **start\_tracking** ()
- int **general\_work** (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)
- void **forecast** (int noutput\_items, gr\_vector\_int &ninput\_items\_required)

**Friends**

- [glonass\\_l1\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_cc\\_sptr](#) **glonass\_l1\_ca\_dll\_pll\_c\_aid\_make\_tracking\_cc** (int64\_t fs\_in, uint32\_t vector\_length, bool dump, const std::string &dump\_filename, float pll\_bw\_hz, float dll\_bw\_hz, float pll\_bw\_narrow\_hz, float dll\_bw\_narrow\_hz, int32\_t extend\_correlation\_ms, float early\_late\_space\_chips)

**10.158.1 Detailed Description**

This class implements a DLL + PLL tracking loop block.

Definition at line 68 of file [glonass\\_l1\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_cc.h](#).

The documentation for this class was generated from the following file:

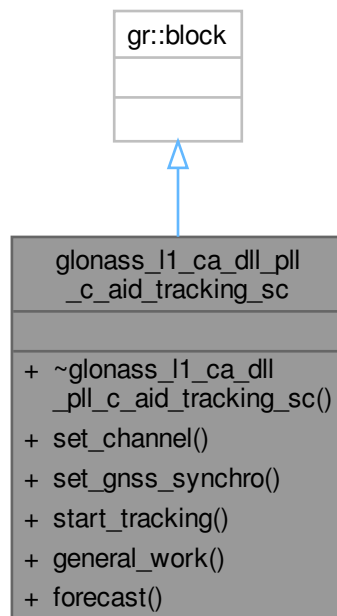
- [glonass\\_l1\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_cc.h](#)

**10.159 glonass\_l1\_ca\_dll\_pll\_c\_aid\_tracking\_sc Class Reference**

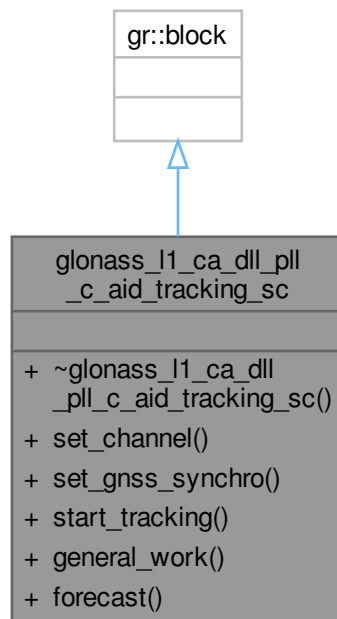
This class implements a DLL + PLL tracking loop block.

`#include <glonass_l1_ca_dll_pll_c_aid_tracking_sc.h>`

Inheritance diagram for `glonass_l1_ca_dll_pll_c_aid_tracking_sc`:



Collaboration diagram for `glonass_l1_ca_dll_pll_c_aid_tracking_sc`:



### Public Member Functions

- void **set\_channel** (uint32\_t channel)
- void **set\_gnss\_synchro** ([Gnss\\_Synchro](#) \*p\_gnss\_synchro)
- void **start\_tracking** ()
- int **general\_work** (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)
- void **forecast** (int noutput\_items, gr\_vector\_int &ninput\_items\_required)

### Friends

- `glonass_l1_ca_dll_pll_c_aid_tracking_sc_sptr` **glonass\_l1\_ca\_dll\_pll\_c\_aid\_make\_tracking\_sc** (int64\_t fs\_in, uint32\_t vector\_length, bool dump, const std::string &dump\_filename, float pll\_bw\_hz, float dll\_bw\_hz, float pll\_bw\_narrow\_hz, float dll\_bw\_narrow\_hz, int32\_t extend\_correlation\_ms, float early\_late\_space\_chips)

## 10.159.1 Detailed Description

This class implements a DLL + PLL tracking loop block.

Definition at line 69 of file [glonass\\_l1\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_sc.h](#).

The documentation for this class was generated from the following file:

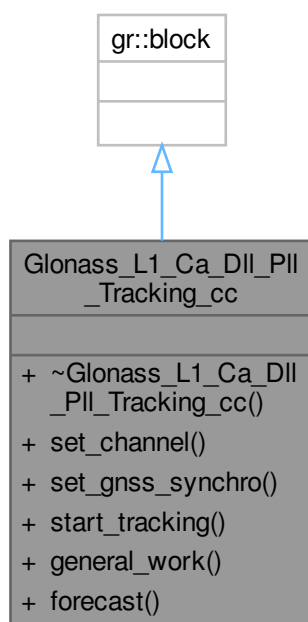
- [glonass\\_l1\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_sc.h](#)

## 10.160 Glonass\_L1\_Ca\_Dll\_Pll\_Tracking\_cc Class Reference

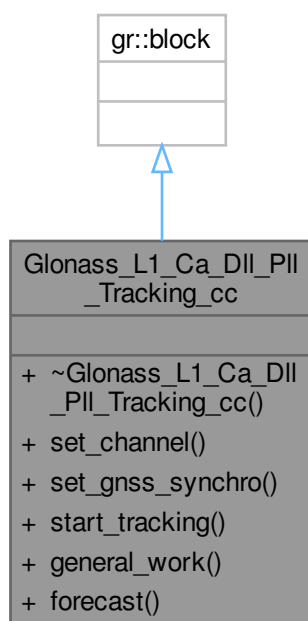
This class implements a DLL + PLL tracking loop block.

```
#include <glonass_l1_ca_dll_pll_tracking_cc.h>
```

Inheritance diagram for Glonass\_L1\_Ca\_Dll\_Pll\_Tracking\_cc:



Collaboration diagram for Glonass\_L1\_Ca\_Dll\_Pll\_Tracking\_cc:



### Public Member Functions

- void **set\_channel** (uint32\_t channel)
- void **set\_gnss\_synchro** ([Gnss\\_Synchro](#) \*p\_gnss\_synchro)
- void **start\_tracking** ()
- int **general\_work** (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)
- void **forecast** (int noutput\_items, gr\_vector\_int &ninput\_items\_required)

### Friends

- [glonass\\_l1\\_ca\\_dll\\_pll\\_tracking\\_cc](#) sptr **glonass\_l1\_ca\_dll\_pll\_make\_tracking\_cc** (int64\_t fs\_in, uint32\_t vector\_length, bool dump, const std::string &dump\_filename, float pll\_bw\_hz, float dll\_bw\_hz, float early\_↵late\_space\_chips)

### 10.160.1 Detailed Description

This class implements a DLL + PLL tracking loop block.

Definition at line 63 of file [glonass\\_l1\\_ca\\_dll\\_pll\\_tracking\\_cc.h](#).

The documentation for this class was generated from the following file:

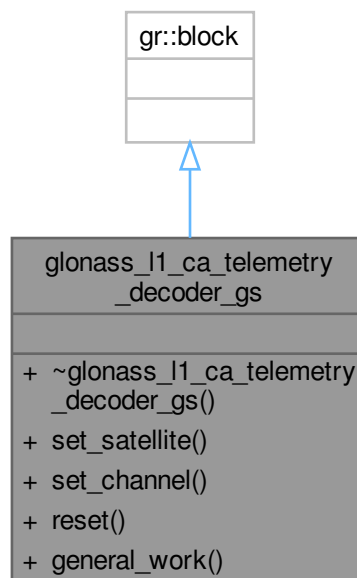
- [glonass\\_l1\\_ca\\_dll\\_pll\\_tracking\\_cc.h](#)

## 10.161 glonass\_l1\_ca\_telemetry\_decoder\_gs Class Reference

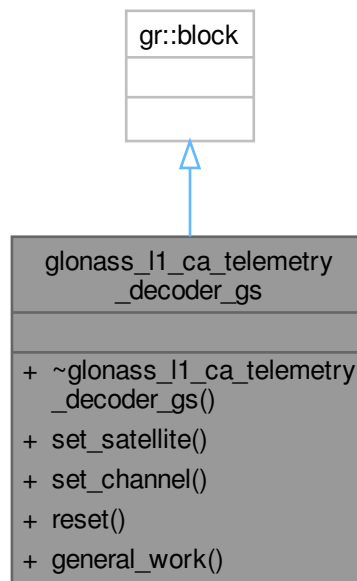
This class implements a block that decodes the GNAV data defined in GLONASS ICD v5.1.

#include <[glonass\\_l1\\_ca\\_telemetry\\_decoder\\_gs.h](#)>

Inheritance diagram for [glonass\\_l1\\_ca\\_telemetry\\_decoder\\_gs](#):



Collaboration diagram for glonass\_l1\_ca\_telemetry\_decoder\_gs:



## Public Member Functions

- `~glonass_l1_ca_telemetry_decoder_gs ()` override  
*Class destructor.*
- void `set_satellite` (const `Gnss_Satellite` &satellite)  
*Set satellite PRN.*
- void `set_channel` (int32\_t channel)  
*Set receiver's channel.*
- void `reset` ()
- int `general_work` (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items) override  
*This is where all signal processing takes place.*

## Friends

- `glonass_l1_ca_telemetry_decoder_gs_sptr glonass_l1_ca_make_telemetry_decoder_gs` (const `Gnss_Satellite` &satellite, const `Tlm_Conf` &conf)

### 10.161.1 Detailed Description

This class implements a block that decodes the GNAV data defined in GLONASS ICD v5.1.

#### Note

Code added as part of GSoC 2017 program

#### See also

[GLONASS ICD](#)

Definition at line 59 of file `glonass_l1_ca_telemetry_decoder_gs.h`.

## 10.161.2 Constructor & Destructor Documentation

### 10.161.2.1 `~glonass_l1_ca_telemetry_decoder_gs()`

`glonass_l1_ca_telemetry_decoder_gs::~~glonass_l1_ca_telemetry_decoder_gs () [override]`  
 Class destructor.

## 10.161.3 Member Function Documentation

### 10.161.3.1 `general_work()`

```
int glonass_l1_ca_telemetry_decoder_gs::general_work (
    int noutput_items,
    gr_vector_int & ninput_items,
    gr_vector_const_void_star & input_items,
    gr_vector_void_star & output_items) [override]
```

This is where all signal processing takes place.

### 10.161.3.2 `reset()`

`void glonass_l1_ca_telemetry_decoder_gs::reset () [inline]`  
 Definition at line 65 of file [glonass\\_l1\\_ca\\_telemetry\\_decoder\\_gs.h](#).

### 10.161.3.3 `set_channel()`

```
void glonass_l1_ca_telemetry_decoder_gs::set_channel (
    int32_t channel)
```

Set receiver's channel.

### 10.161.3.4 `set_satellite()`

```
void glonass_l1_ca_telemetry_decoder_gs::set_satellite (
    const Gnss_Satellite & satellite)
```

Set satellite PRN.

The documentation for this class was generated from the following file:

- [glonass\\_l1\\_ca\\_telemetry\\_decoder\\_gs.h](#)

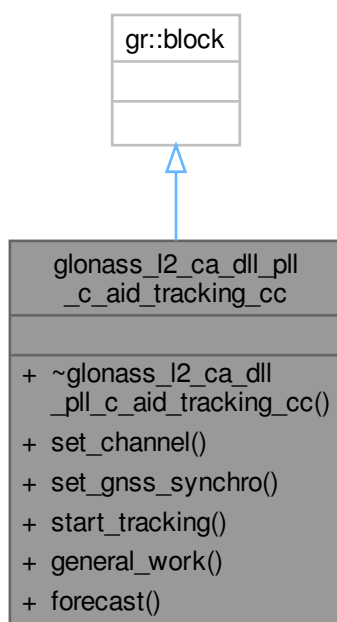
## 10.162 `glonass_l2_ca_dll_pll_c_aid_tracking_cc` Class Reference

This class implements a DLL + PLL tracking loop block.

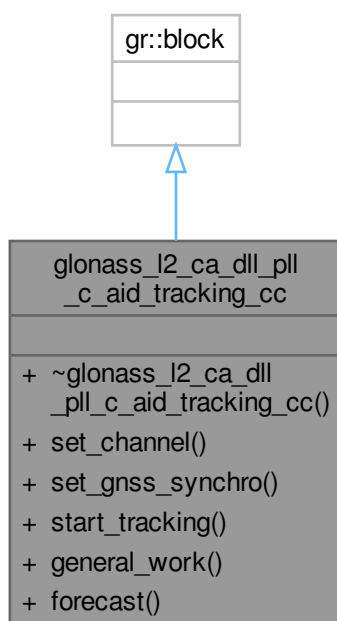
```
#include <glonass_l2_ca_dll_pll_c_aid_tracking_cc.h>
```



Inheritance diagram for glonass\_l2\_ca\_dll\_pll\_c\_aid\_tracking\_cc:



Collaboration diagram for glonass\_l2\_ca\_dll\_pll\_c\_aid\_tracking\_cc:



## Public Member Functions

- void **set\_channel** (uint32\_t channel)
- void **set\_gnss\_synchro** ([Gnss\\_Synchro](#) \*p\_gnss\_synchro)
- void **start\_tracking** ()
- int **general\_work** (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)
- void **forecast** (int noutput\_items, gr\_vector\_int &ninput\_items\_required)

## Friends

- [glonass\\_l2\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_cc\\_sptr](#) **glonass\_l2\_ca\_dll\_pll\_c\_aid\_make\_tracking\_cc** (int64\_t fs\_in, uint32\_t vector\_length, bool dump, const std::string &dump\_filename, float pll\_bw\_hz, float dll\_bw\_hz, float pll\_bw\_narrow\_hz, float dll\_bw\_narrow\_hz, int32\_t extend\_correlation\_ms, float early\_late\_space\_chips)

### 10.162.1 Detailed Description

This class implements a DLL + PLL tracking loop block.

Definition at line 65 of file [glonass\\_l2\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_cc.h](#).

The documentation for this class was generated from the following file:

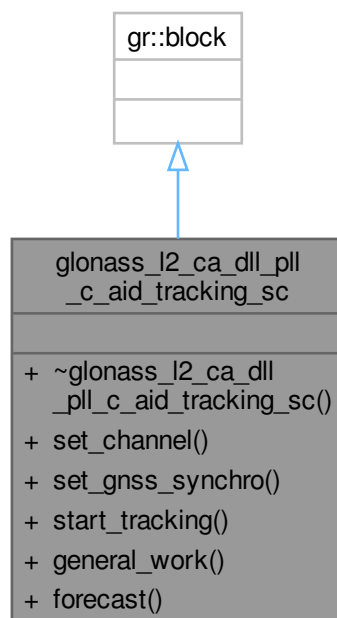
- [glonass\\_l2\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_cc.h](#)

### 10.163 glonass\_l2\_ca\_dll\_pll\_c\_aid\_tracking\_sc Class Reference

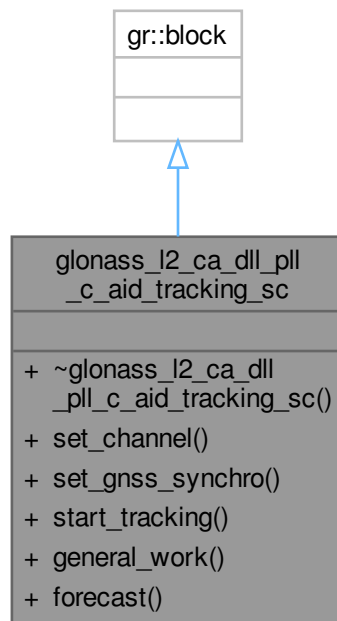
This class implements a DLL + PLL tracking loop block.

`#include <glonass_l2_ca_dll_pll_c_aid_tracking_sc.h>`

Inheritance diagram for `glonass_l2_ca_dll_pll_c_aid_tracking_sc`:



Collaboration diagram for `glonass_l2_ca_dll_pll_c_aid_tracking_sc`:



### Public Member Functions

- void **set\_channel** (uint32\_t channel)
- void **set\_gnss\_synchro** ([Gnss\\_Synchro](#) \*p\_gnss\_synchro)
- void **start\_tracking** ()
- int **general\_work** (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)
- void **forecast** (int noutput\_items, gr\_vector\_int &ninput\_items\_required)

### Friends

- `glonass_l2_ca_dll_pll_c_aid_tracking_sc_sptr` **glonass\_l2\_ca\_dll\_pll\_c\_aid\_make\_tracking\_sc** (int64\_t fs\_in, uint32\_t vector\_length, bool dump, const std::string &dump\_filename, float pll\_bw\_hz, float dll\_bw\_hz, float pll\_bw\_narrow\_hz, float dll\_bw\_narrow\_hz, int32\_t extend\_correlation\_ms, float early\_late\_space\_chips)

### 10.163.1 Detailed Description

This class implements a DLL + PLL tracking loop block.

Definition at line 65 of file [glonass\\_l2\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_sc.h](#).

The documentation for this class was generated from the following file:

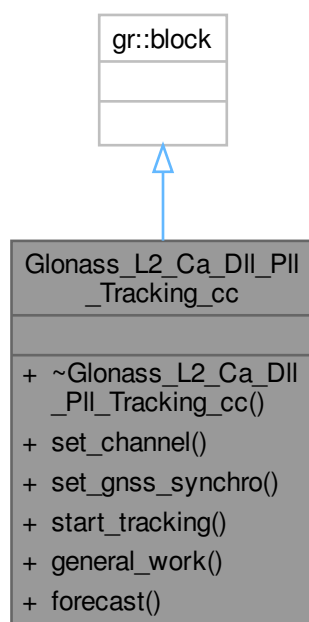
- [glonass\\_l2\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_sc.h](#)

## 10.164 Glonass\_L2\_Ca\_Dll\_Pll\_Tracking\_cc Class Reference

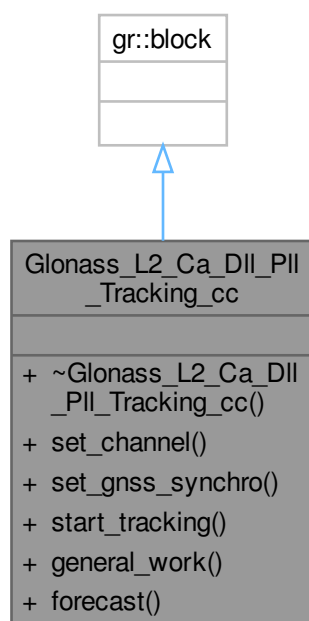
This class implements a DLL + PLL tracking loop block.

```
#include <glonass_l2_ca_dll_pll_tracking_cc.h>
```

Inheritance diagram for Glonass\_L2\_Ca\_Dll\_Pll\_Tracking\_cc:



Collaboration diagram for Glonass\_L2\_Ca\_Dll\_Pll\_Tracking\_cc:



**Public Member Functions**

- void **set\_channel** (uint32\_t channel)
- void **set\_gnss\_synchro** ([Gnss\\_Synchro](#) \*p\_gnss\_synchro)
- void **start\_tracking** ()
- int **general\_work** (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)
- void **forecast** (int noutput\_items, gr\_vector\_int &ninput\_items\_required)

**Friends**

- [glonass\\_l2\\_ca\\_dll\\_pll\\_tracking\\_cc](#) sptr **glonass\_l2\_ca\_dll\_pll\_make\_tracking\_cc** (int64\_t fs\_in, uint32\_t vector\_length, bool dump, const std::string &dump\_filename, float pll\_bw\_hz, float dll\_bw\_hz, float early\_↵late\_space\_chips)

**10.164.1 Detailed Description**

This class implements a DLL + PLL tracking loop block.

Definition at line 60 of file [glonass\\_l2\\_ca\\_dll\\_pll\\_tracking\\_cc.h](#).

The documentation for this class was generated from the following file:

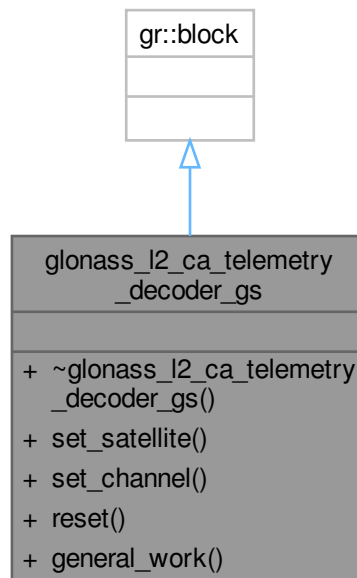
- [glonass\\_l2\\_ca\\_dll\\_pll\\_tracking\\_cc.h](#)

**10.165 glonass\_l2\_ca\_telemetry\_decoder\_gs Class Reference**

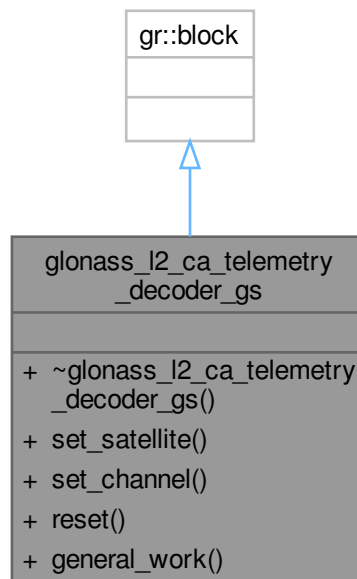
This class implements a block that decodes the GNAV data defined in GLONASS ICD v5.1.

```
#include <glonass_l2_ca_telemetry_decoder_gs.h>
```

Inheritance diagram for `glonass_l2_ca_telemetry_decoder_gs`:



Collaboration diagram for `glonass_l2_ca_telemetry_decoder_gs`:



## Public Member Functions

- `~glonass_l2_ca_telemetry_decoder_gs ()` override  
*Class destructor.*
- void `set_satellite` (const `Gnss_Satellite` &satellite)  
*Set satellite PRN.*
- void `set_channel` (int32\_t channel)  
*Set receiver's channel.*
- void `reset` ()
- int `general_work` (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items) override  
*This is where all signal processing takes place.*

## Friends

- `glonass_l2_ca_telemetry_decoder_gs_sptr glonass_l2_ca_make_telemetry_decoder_gs` (const `Gnss_Satellite` &satellite, const `Tlm_Conf` &conf)

### 10.165.1 Detailed Description

This class implements a block that decodes the GNAV data defined in GLONASS ICD v5.1.

See also

[GLONASS ICD](#)

Definition at line 57 of file `glonass_l2_ca_telemetry_decoder_gs.h`.

## 10.165.2 Constructor & Destructor Documentation

### 10.165.2.1 ~glonass\_l2\_ca\_telemetry\_decoder\_gs()

```
glonass_l2_ca_telemetry_decoder_gs::~glonass_l2_ca_telemetry_decoder_gs () [override]
```

Class destructor.

## 10.165.3 Member Function Documentation

### 10.165.3.1 general\_work()

```
int glonass_l2_ca_telemetry_decoder_gs::general_work (
    int noutput_items,
    gr_vector_int & ninput_items,
    gr_vector_const_void_star & input_items,
    gr_vector_void_star & output_items) [override]
```

This is where all signal processing takes place.

### 10.165.3.2 reset()

```
void glonass_l2_ca_telemetry_decoder_gs::reset () [inline]
```

Definition at line 63 of file [glonass\\_l2\\_ca\\_telemetry\\_decoder\\_gs.h](#).

### 10.165.3.3 set\_channel()

```
void glonass_l2_ca_telemetry_decoder_gs::set_channel (
    int32_t channel)
```

Set receiver's channel.

### 10.165.3.4 set\_satellite()

```
void glonass_l2_ca_telemetry_decoder_gs::set_satellite (
    const Gnss_Satellite & satellite)
```

Set satellite PRN.

The documentation for this class was generated from the following file:

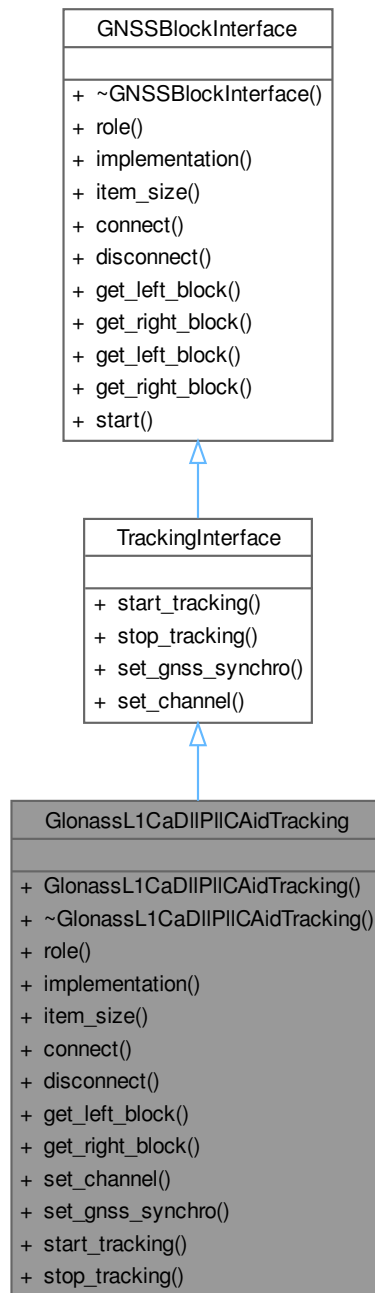
- [glonass\\_l2\\_ca\\_telemetry\\_decoder\\_gs.h](#)

## 10.166 GlonassL1CaDIIPICAidTracking Class Reference

This class implements a code DLL + carrier PLL tracking loop.

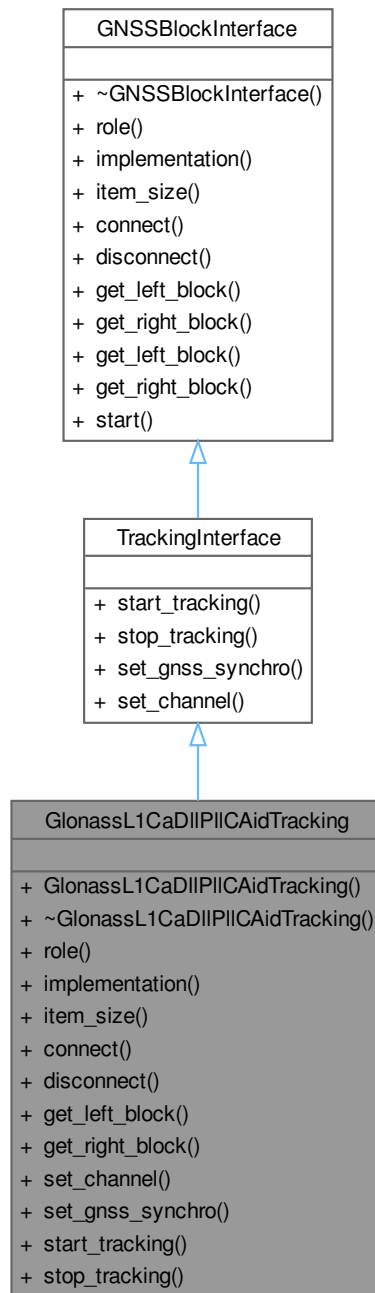
```
#include <glonass_l1_ca_dll_pll_c_aid_tracking.h>
```

Inheritance diagram for GlonassL1CaDIIPICAidTracking:





Collaboration diagram for GlonassL1CaDIIPICAidTracking:



### Public Member Functions

- **GlonassL1CaDIIPICAidTracking** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override
  - Returns "GLONASS\_L1\_CA\_DLL\_PLL\_C\_Aid\_Tracking".
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_channel](#) (unsigned int channel) override
 

*Set tracking channel unique ID.*
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override
 

*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [start\\_tracking](#) () override
- void [stop\\_tracking](#) () override
 

*Stop running tracking.*

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()
 

*Start the flow of samples if needed.*

### 10.166.1 Detailed Description

This class implements a code DLL + carrier PLL tracking loop.  
 Definition at line 45 of file [glonass\\_l1\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking.h](#).

### 10.166.2 Member Function Documentation

#### 10.166.2.1 connect()

```
void GlonassL1CaDllPllCAidTracking::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.166.2.2 disconnect()

```
void GlonassL1CaDllPllCAidTracking::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.166.2.3 get\_left\_block()

```
gr::basic_block_sptr GlonassL1CaDllPllCAidTracking::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.166.2.4 get\_right\_block()

```
gr::basic_block_sptr GlonassL1CaDllPllCAidTracking::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.166.2.5 implementation()

```
std::string GlonassL1CaDllPllCAidTracking::implementation () [inline], [override], [virtual]
```

Returns "GLONASS\_L1\_CA\_DLL\_PLL\_C\_Aid\_Tracking".  
 Implements [GNSSBlockInterface](#).  
 Definition at line 62 of file [glonass\\_l1\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking.h](#).

**10.166.2.6 item\_size()**

```
size_t GlonassL1CaDllPllCAidTracking::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 67 of file [glonass\\_l1\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking.h](#).

**10.166.2.7 role()**

```
std::string GlonassL1CaDllPllCAidTracking::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 56 of file [glonass\\_l1\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking.h](#).

**10.166.2.8 set\_channel()**

```
void GlonassL1CaDllPllCAidTracking::set_channel (
    unsigned int channel) [override], [virtual]
```

Set tracking channel unique ID.

Implements [TrackingInterface](#).

**10.166.2.9 set\_gnss\_synchro()**

```
void GlonassL1CaDllPllCAidTracking::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [TrackingInterface](#).

**10.166.2.10 start\_tracking()**

```
void GlonassL1CaDllPllCAidTracking::start_tracking () [override], [virtual]
```

Implements [TrackingInterface](#).

**10.166.2.11 stop\_tracking()**

```
void GlonassL1CaDllPllCAidTracking::stop_tracking () [override], [virtual]
```

Stop running tracking.

Implements [TrackingInterface](#).

The documentation for this class was generated from the following file:

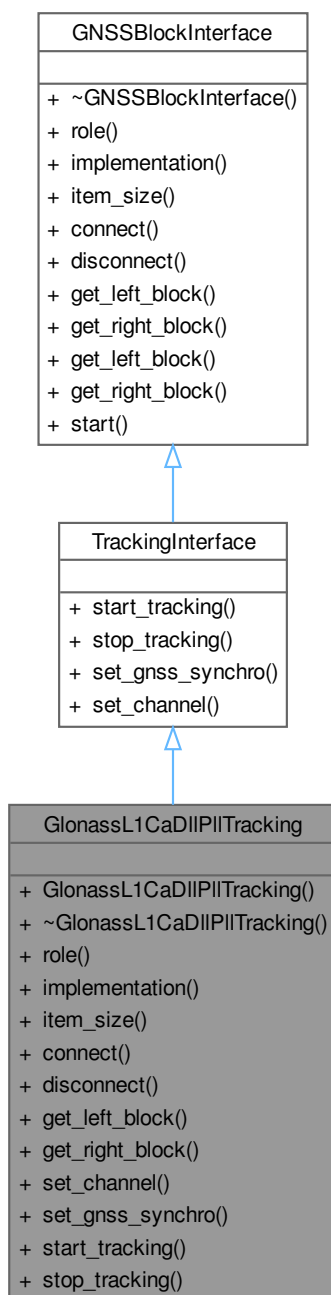
- [glonass\\_l1\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking.h](#)

**10.167 GlonassL1CaDllPllCAidTracking Class Reference**

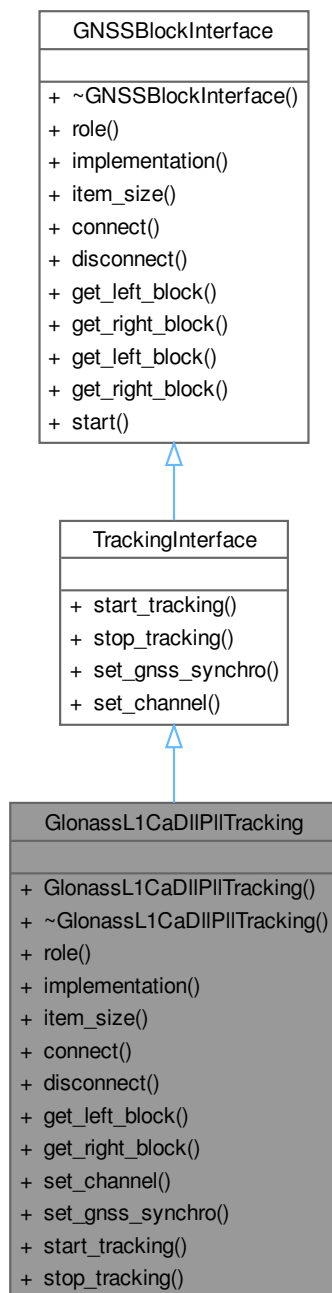
This class implements a code DLL + carrier PLL tracking loop.

```
#include <glonass_l1_ca_dll_pll_tracking.h>
```

Inheritance diagram for GlonassL1CaDIIPIITracking:



Collaboration diagram for GlonassL1CaDIIPIITracking:



## Public Member Functions

- **GlonassL1CaDIIPIITracking** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override
  - Returns "GLONASS\_L1\_CA\_DLL\_PLL\_Tracking".
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_channel](#) (unsigned int channel) override
 

*Set tracking channel unique ID.*
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override
 

*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [start\\_tracking](#) () override
- void [stop\\_tracking](#) () override
 

*Stop running tracking.*

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()
 

*Start the flow of samples if needed.*

### 10.167.1 Detailed Description

This class implements a code DLL + carrier PLL tracking loop.  
 Definition at line 43 of file [glonass\\_l1\\_ca\\_dll\\_pll\\_tracking.h](#).

### 10.167.2 Member Function Documentation

#### 10.167.2.1 connect()

```
void GlonassL1CaDllPllTracking::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.167.2.2 disconnect()

```
void GlonassL1CaDllPllTracking::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.167.2.3 get\_left\_block()

```
gr::basic_block_sptr GlonassL1CaDllPllTracking::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.167.2.4 get\_right\_block()

```
gr::basic_block_sptr GlonassL1CaDllPllTracking::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.167.2.5 implementation()

```
std::string GlonassL1CaDllPllTracking::implementation () [inline], [override], [virtual]
```

Returns "GLONASS\_L1\_CA\_DLL\_PLL\_Tracking".  
 Implements [GNSSBlockInterface](#).  
 Definition at line 60 of file [glonass\\_l1\\_ca\\_dll\\_pll\\_tracking.h](#).

**10.167.2.6 item\_size()**

```
size_t GlonassL1CaDllPllTracking::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 65 of file [glonass\\_l1\\_ca\\_dll\\_pll\\_tracking.h](#).

**10.167.2.7 role()**

```
std::string GlonassL1CaDllPllTracking::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 54 of file [glonass\\_l1\\_ca\\_dll\\_pll\\_tracking.h](#).

**10.167.2.8 set\_channel()**

```
void GlonassL1CaDllPllTracking::set_channel (
    unsigned int channel) [override], [virtual]
```

Set tracking channel unique ID.

Implements [TrackingInterface](#).

**10.167.2.9 set\_gnss\_synchro()**

```
void GlonassL1CaDllPllTracking::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [TrackingInterface](#).

**10.167.2.10 start\_tracking()**

```
void GlonassL1CaDllPllTracking::start_tracking () [override], [virtual]
```

Implements [TrackingInterface](#).

**10.167.2.11 stop\_tracking()**

```
void GlonassL1CaDllPllTracking::stop_tracking () [override], [virtual]
```

Stop running tracking.

Implements [TrackingInterface](#).

The documentation for this class was generated from the following file:

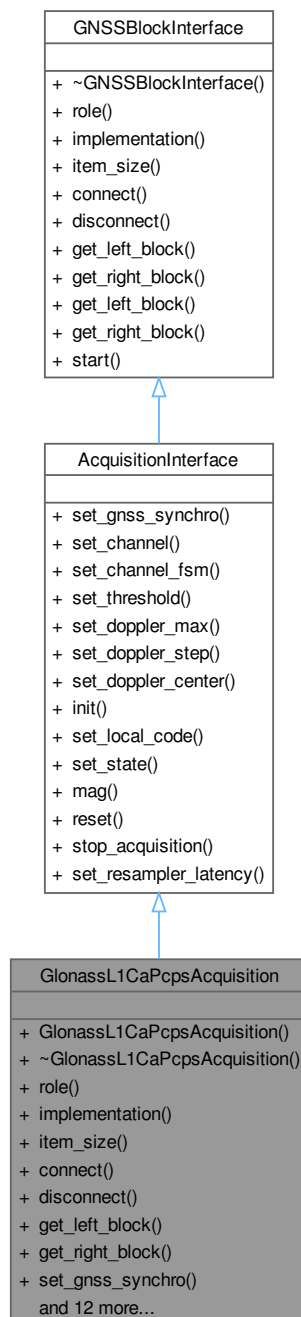
- [glonass\\_l1\\_ca\\_dll\\_pll\\_tracking.h](#)

**10.168 GlonassL1CaPcpsAcquisition Class Reference**

This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.

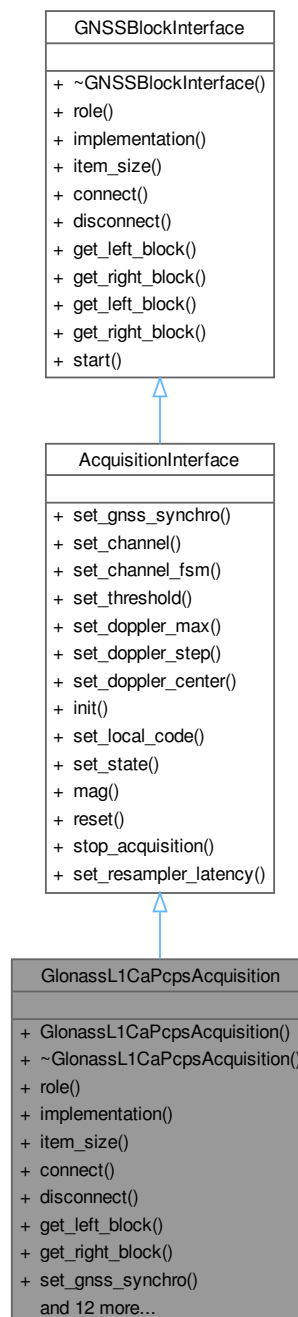
```
#include <glonass_l1_ca_pcps_acquisition.h>
```

Inheritance diagram for GlonassL1CaPcpsAcquisition:





Collaboration diagram for GlonassL1CaPcpsAcquisition:



## Public Member Functions

- **GlonassL1CaPcpsAcquisition** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override
 

*Returns "GLONASS\_L1\_CA\_PCPS\_Acquisition".*
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [set\\_channel](#) (unsigned int channel) override  
*Set acquisition channel unique ID.*
- void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm) override  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold) override  
*Set statistics threshold of PCPS algorithm.*
- void [set\\_doppler\\_max](#) (unsigned int doppler\_max) override  
*Set maximum Doppler off grid search.*
- void [set\\_doppler\\_step](#) (unsigned int doppler\_step) override  
*Set Doppler steps for the grid search.*
- void [init](#) () override  
*Initializes acquisition algorithm.*
- void [set\\_local\\_code](#) () override  
*Sets local code for GPS L1/CA PCPS acquisition algorithm.*
- signed int [mag](#) () override  
*Returns the maximum peak of grid search.*
- void [reset](#) () override  
*Restart acquisition algorithm.*
- void [set\\_state](#) (int state) override  
*If state = 1, it forces the block to start acquiring from the first sample.*
- void [stop\\_acquisition](#) () override  
*Stop running acquisition.*
- void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples \_\_attribute\_\_((unused))) override

## Public Member Functions inherited from [AcquisitionInterface](#)

- virtual void [set\\_doppler\\_center](#) (int doppler\_center \_\_attribute\_\_((unused)))
- virtual void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples)=0

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.168.1 Detailed Description

This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.  
Definition at line 46 of file [glonass\\_l1\\_ca\\_pcps\\_acquisition.h](#).

### 10.168.2 Member Function Documentation

#### 10.168.2.1 [connect\(\)](#)

```
void GlonassL1CaPcpsAcquisition::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.168.2.2 disconnect()

```
void GlonassL1CaPcpsAcquisition::disconnect (  
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.168.2.3 get\_left\_block()

```
gr::basic_block_sptr GlonassL1CaPcpsAcquisition::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.168.2.4 get\_right\_block()

```
gr::basic_block_sptr GlonassL1CaPcpsAcquisition::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.168.2.5 implementation()

```
std::string GlonassL1CaPcpsAcquisition::implementation () [inline], [override], [virtual]
```

Returns "GLONASS\_L1\_CA\_PCPS\_Acquisition".

Implements [GNSSBlockInterface](#).

Definition at line 65 of file [glonass\\_l1\\_ca\\_pcps\\_acquisition.h](#).

### 10.168.2.6 init()

```
void GlonassL1CaPcpsAcquisition::init () [override], [virtual]
```

Initializes acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.168.2.7 item\_size()

```
size_t GlonassL1CaPcpsAcquisition::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 70 of file [glonass\\_l1\\_ca\\_pcps\\_acquisition.h](#).

### 10.168.2.8 mag()

```
signed int GlonassL1CaPcpsAcquisition::mag () [override], [virtual]
```

Returns the maximum peak of grid search.

Implements [AcquisitionInterface](#).

### 10.168.2.9 reset()

```
void GlonassL1CaPcpsAcquisition::reset () [override], [virtual]
```

Restart acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.168.2.10 role()

```
std::string GlonassL1CaPcpsAcquisition::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 57 of file [glonass\\_l1\\_ca\\_pcps\\_acquisition.h](#).

### 10.168.2.11 set\_channel()

```
void GlonassL1CaPcpsAcquisition::set_channel (  
    unsigned int channel) [inline], [override], [virtual]
```

Set acquisition channel unique ID.

Implements [AcquisitionInterface](#).

Definition at line 90 of file [glonass\\_l1\\_ca\\_pcps\\_acquisition.h](#).

**10.168.2.12 set\_channel\_fsm()**

```
void GlonassL1CaPcpsAcquisition::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline], [override], [virtual]
```

Set channel fsm associated to this acquisition instance.

Implements [AcquisitionInterface](#).

Definition at line 99 of file [glonass\\_l1\\_ca\\_pcps\\_acquisition.h](#).

**10.168.2.13 set\_doppler\_max()**

```
void GlonassL1CaPcpsAcquisition::set_doppler_max (
    unsigned int doppler_max) [override], [virtual]
```

Set maximum Doppler off grid search.

Implements [AcquisitionInterface](#).

**10.168.2.14 set\_doppler\_step()**

```
void GlonassL1CaPcpsAcquisition::set_doppler_step (
    unsigned int doppler_step) [override], [virtual]
```

Set Doppler steps for the grid search.

Implements [AcquisitionInterface](#).

**10.168.2.15 set\_gnss\_synchro()**

```
void GlonassL1CaPcpsAcquisition::set_gnss_synchro (
    Gnss\_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [AcquisitionInterface](#).

**10.168.2.16 set\_local\_code()**

```
void GlonassL1CaPcpsAcquisition::set_local_code () [override], [virtual]
```

Sets local code for GPS L1/CA PCPS acquisition algorithm.

Implements [AcquisitionInterface](#).

**10.168.2.17 set\_resampler\_latency()**

```
void GlonassL1CaPcpsAcquisition::set_resampler_latency (
    uint32_t latency_samples __attribute__((unused))) [inline], [override]
```

Definition at line 150 of file [glonass\\_l1\\_ca\\_pcps\\_acquisition.h](#).

**10.168.2.18 set\_state()**

```
void GlonassL1CaPcpsAcquisition::set_state (
    int state) [override], [virtual]
```

If state = 1, it forces the block to start acquiring from the first sample.

Implements [AcquisitionInterface](#).

**10.168.2.19 set\_threshold()**

```
void GlonassL1CaPcpsAcquisition::set_threshold (
    float threshold) [override], [virtual]
```

Set statistics threshold of PCPS algorithm.

Implements [AcquisitionInterface](#).

**10.168.2.20 stop\_acquisition()**

```
void GlonassL1CaPcpsAcquisition::stop_acquisition () [override], [virtual]
```

Stop running acquisition.

Implements [AcquisitionInterface](#).

The documentation for this class was generated from the following file:

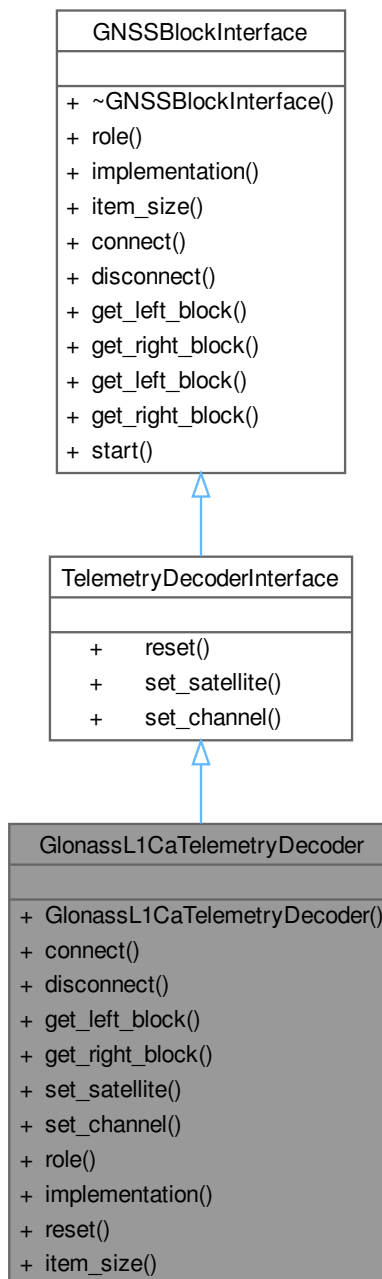
- [glonass\\_l1\\_ca\\_pcps\\_acquisition.h](#)

## 10.169 GlonassL1CaTelemetryDecoder Class Reference

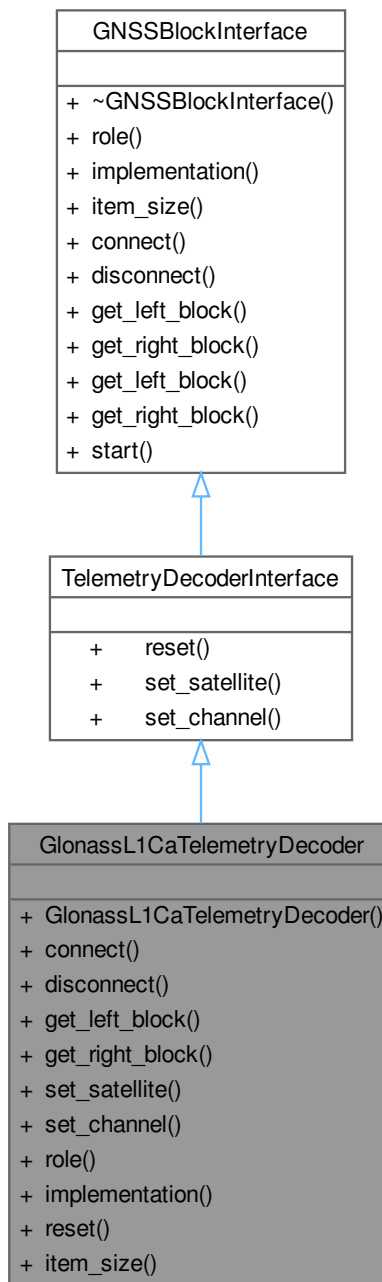
This class implements a NAV data decoder for GLONASS L1 C/A.

```
#include <glonass_l1_ca_telemetry_decoder.h>
```

Inheritance diagram for GlonassL1CaTelemetryDecoder:



Collaboration diagram for GlonassL1CaTelemetryDecoder:



### Public Member Functions

- **GlonassL1CaTelemetryDecoder** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override

- void [set\\_satellite](#) (const [Gnss\\_Satellite](#) &satellite) override
- void [set\\_channel](#) (int channel) override
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "[GLONASS\\_L1\\_CA\\_Telemetry\\_Decoder](#)".
- void [reset](#) () override
- size\_t [item\\_size](#) () override

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.169.1 Detailed Description

This class implements a NAV data decoder for GLONASS L1 C/A.  
Definition at line 43 of file [glonass\\_l1\\_ca\\_telemetry\\_decoder.h](#).

### 10.169.2 Member Function Documentation

#### 10.169.2.1 connect()

```
void GlonassL1CaTelemetryDecoder::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.169.2.2 disconnect()

```
void GlonassL1CaTelemetryDecoder::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.169.2.3 get\_left\_block()

```
gr::basic_block_sptr GlonassL1CaTelemetryDecoder::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.169.2.4 get\_right\_block()

```
gr::basic_block_sptr GlonassL1CaTelemetryDecoder::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.169.2.5 implementation()

```
std::string GlonassL1CaTelemetryDecoder::implementation () [inline], [override], [virtual]
```

Returns "[GLONASS\\_L1\\_CA\\_Telemetry\\_Decoder](#)".

Implements [GNSSBlockInterface](#).

Definition at line 66 of file [glonass\\_l1\\_ca\\_telemetry\\_decoder.h](#).

#### 10.169.2.6 item\_size()

```
size_t GlonassL1CaTelemetryDecoder::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 76 of file [glonass\\_l1\\_ca\\_telemetry\\_decoder.h](#).

#### 10.169.2.7 reset()

```
void GlonassL1CaTelemetryDecoder::reset () [inline], [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

Definition at line 71 of file [glonass\\_l1\\_ca\\_telemetry\\_decoder.h](#).

#### 10.169.2.8 role()

```
std::string GlonassL1CaTelemetryDecoder::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 60 of file [glonass\\_l1\\_ca\\_telemetry\\_decoder.h](#).

#### 10.169.2.9 set\_channel()

```
void GlonassL1CaTelemetryDecoder::set_channel (  
    int channel) [inline], [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

Definition at line 58 of file [glonass\\_l1\\_ca\\_telemetry\\_decoder.h](#).

#### 10.169.2.10 set\_satellite()

```
void GlonassL1CaTelemetryDecoder::set_satellite (  
    const Gnss\_Satellite & satellite) [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

The documentation for this class was generated from the following file:

- [glonass\\_l1\\_ca\\_telemetry\\_decoder.h](#)

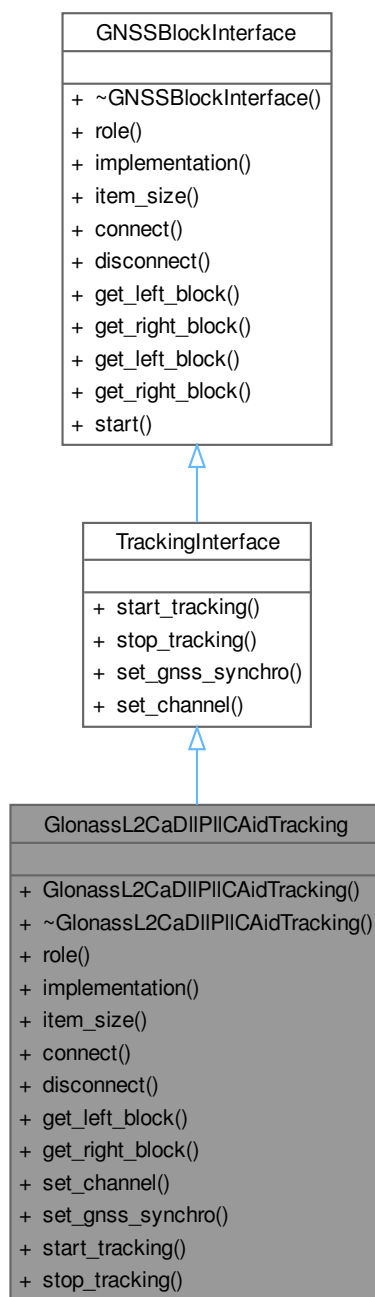
## 10.170 GlonassL2CaDIIPICaIdTracking Class Reference

This class implements a code DLL + carrier PLL tracking loop.

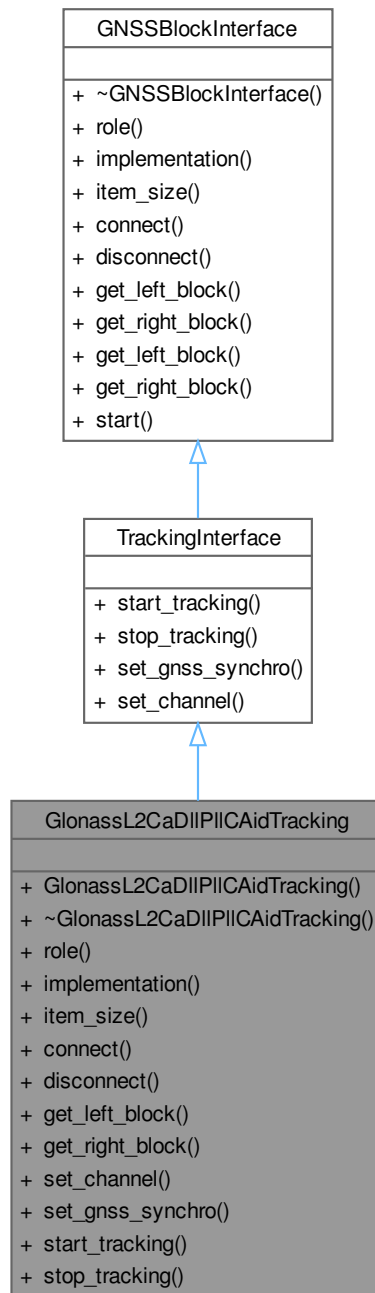
```
#include <glonass_l2_ca_dll_pll_c_aid_tracking.h>
```



Inheritance diagram for GlonassL2CaDIIPICAidTracking:



Collaboration diagram for GlonassL2CaDIIPICAidTracking:



## Public Member Functions

- **GlonassL2CaDIIPICAidTracking** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override
  - Returns "GLONASS\_L2\_CA\_DLL\_PLL\_C\_Aid\_Tracking".
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_channel](#) (unsigned int channel) override  
*Set tracking channel unique ID.*
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [start\\_tracking](#) () override
- void [stop\\_tracking](#) () override  
*Stop running tracking.*

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()  
*Start the flow of samples if needed.*

### 10.170.1 Detailed Description

This class implements a code DLL + carrier PLL tracking loop.  
Definition at line 43 of file [glonass\\_l2\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking.h](#).

### 10.170.2 Member Function Documentation

#### 10.170.2.1 connect()

```
void GlonassL2CaDllPllCAidTracking::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.170.2.2 disconnect()

```
void GlonassL2CaDllPllCAidTracking::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.170.2.3 get\_left\_block()

```
gr::basic_block_sptr GlonassL2CaDllPllCAidTracking::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.170.2.4 get\_right\_block()

```
gr::basic_block_sptr GlonassL2CaDllPllCAidTracking::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.170.2.5 implementation()

```
std::string GlonassL2CaDllPllCAidTracking::implementation () [inline], [override], [virtual]
```

Returns "GLONASS\_L2\_CA\_DLL\_PLL\_C\_Aid\_Tracking".  
Implements [GNSSBlockInterface](#).  
Definition at line 60 of file [glonass\\_l2\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking.h](#).

**10.170.2.6 item\_size()**

```
size_t GlonassL2CaDllPllCAidTracking::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 65 of file [glonass\\_l2\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking.h](#).

**10.170.2.7 role()**

```
std::string GlonassL2CaDllPllCAidTracking::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 54 of file [glonass\\_l2\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking.h](#).

**10.170.2.8 set\_channel()**

```
void GlonassL2CaDllPllCAidTracking::set_channel (
    unsigned int channel) [override], [virtual]
```

Set tracking channel unique ID.

Implements [TrackingInterface](#).

**10.170.2.9 set\_gnss\_synchro()**

```
void GlonassL2CaDllPllCAidTracking::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [TrackingInterface](#).

**10.170.2.10 start\_tracking()**

```
void GlonassL2CaDllPllCAidTracking::start_tracking () [override], [virtual]
```

Implements [TrackingInterface](#).

**10.170.2.11 stop\_tracking()**

```
void GlonassL2CaDllPllCAidTracking::stop_tracking () [override], [virtual]
```

Stop running tracking.

Implements [TrackingInterface](#).

The documentation for this class was generated from the following file:

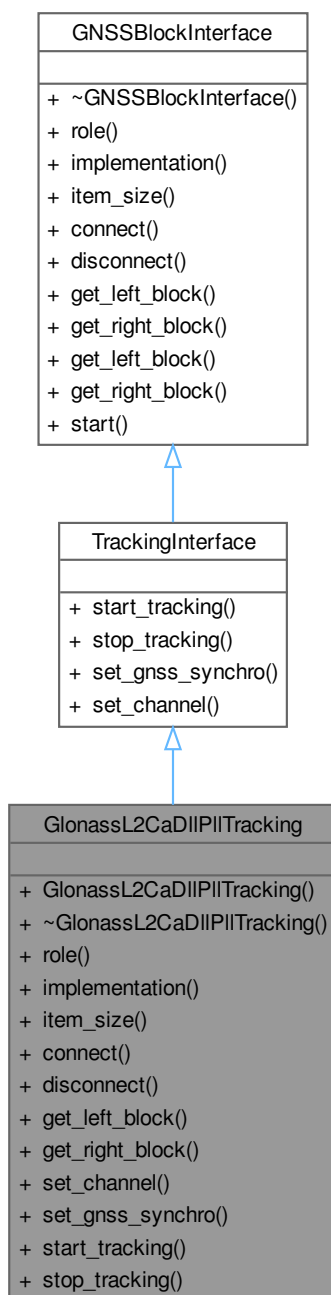
- [glonass\\_l2\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking.h](#)

**10.171 GlonassL2CaDllPllCAidTracking Class Reference**

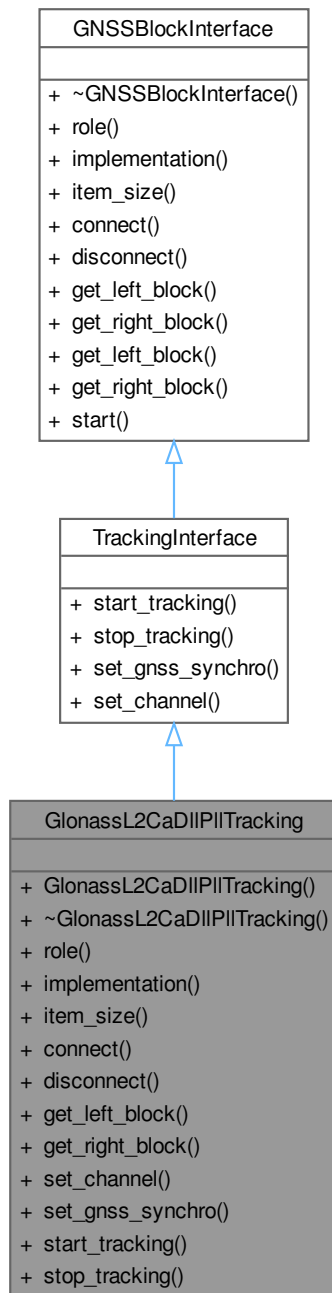
This class implements a code DLL + carrier PLL tracking loop.

```
#include <glonass_l2_ca_dll_pll_tracking.h>
```

Inheritance diagram for GlonassL2CaDIIPITracking:



Collaboration diagram for GlonassL2CaDIIPIITracking:



## Public Member Functions

- **GlonassL2CaDIIPIITracking** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override
  - Returns "GLONASS\_L1\_CA\_DLL\_PLL\_Tracking".
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_channel](#) (unsigned int channel) override  
*Set tracking channel unique ID.*
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [start\\_tracking](#) () override
- void [stop\\_tracking](#) () override  
*Stop running tracking.*

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()  
*Start the flow of samples if needed.*

### 10.171.1 Detailed Description

This class implements a code DLL + carrier PLL tracking loop.  
Definition at line 42 of file [glonass\\_l2\\_ca\\_dll\\_pll\\_tracking.h](#).

### 10.171.2 Member Function Documentation

#### 10.171.2.1 connect()

```
void GlonassL2CaDllPllTracking::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.171.2.2 disconnect()

```
void GlonassL2CaDllPllTracking::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.171.2.3 get\_left\_block()

```
gr::basic_block_sptr GlonassL2CaDllPllTracking::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.171.2.4 get\_right\_block()

```
gr::basic_block_sptr GlonassL2CaDllPllTracking::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.171.2.5 implementation()

```
std::string GlonassL2CaDllPllTracking::implementation () [inline], [override], [virtual]
```

Returns "GLONASS\_L1\_CA\_DLL\_PLL\_Tracking".  
Implements [GNSSBlockInterface](#).  
Definition at line 59 of file [glonass\\_l2\\_ca\\_dll\\_pll\\_tracking.h](#).

**10.171.2.6 item\_size()**

```
size_t GlonassL2CaDllPllTracking::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 64 of file [glonass\\_l2\\_ca\\_dll\\_pll\\_tracking.h](#).

**10.171.2.7 role()**

```
std::string GlonassL2CaDllPllTracking::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 53 of file [glonass\\_l2\\_ca\\_dll\\_pll\\_tracking.h](#).

**10.171.2.8 set\_channel()**

```
void GlonassL2CaDllPllTracking::set_channel (
    unsigned int channel) [override], [virtual]
```

Set tracking channel unique ID.

Implements [TrackingInterface](#).

**10.171.2.9 set\_gnss\_synchro()**

```
void GlonassL2CaDllPllTracking::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [TrackingInterface](#).

**10.171.2.10 start\_tracking()**

```
void GlonassL2CaDllPllTracking::start_tracking () [override], [virtual]
```

Implements [TrackingInterface](#).

**10.171.2.11 stop\_tracking()**

```
void GlonassL2CaDllPllTracking::stop_tracking () [override], [virtual]
```

Stop running tracking.

Implements [TrackingInterface](#).

The documentation for this class was generated from the following file:

- [glonass\\_l2\\_ca\\_dll\\_pll\\_tracking.h](#)

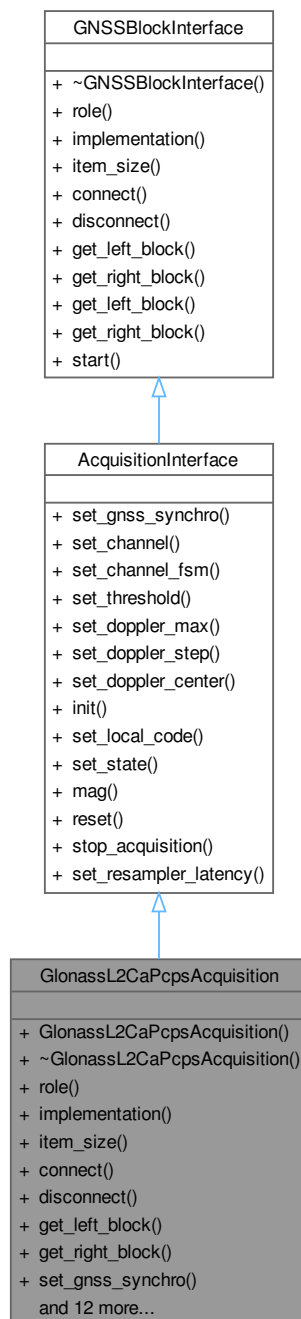
**10.172 GlonassL2CaPcpsAcquisition Class Reference**

This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GLONASS L2 C/A signals.

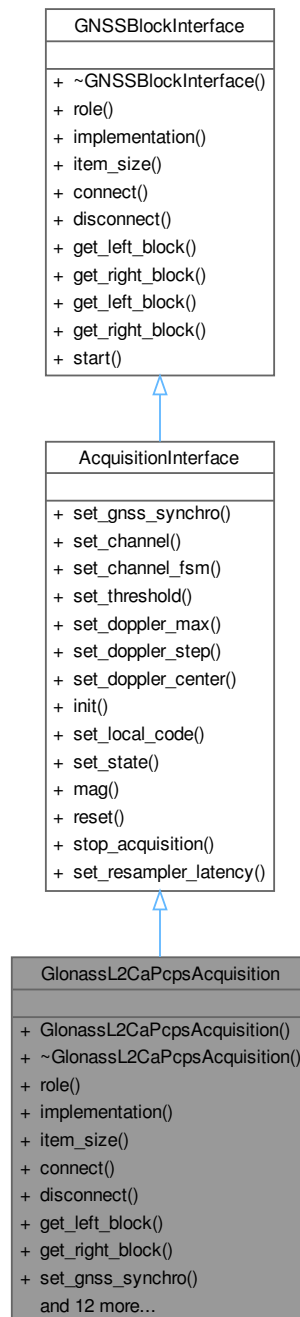
```
#include <glonass_l2_ca_pcps_acquisition.h>
```



Inheritance diagram for GlonassL2CaPcpsAcquisition:



Collaboration diagram for GlonassL2CaPcpsAcquisition:



## Public Member Functions

- **GlonassL2CaPcpsAcquisition** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override
  - Returns "GLONASS\_L2\_CA\_PCPS\_Acquisition".*
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [set\\_channel](#) (unsigned int channel) override  
*Set acquisition channel unique ID.*
- void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm) override  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold) override  
*Set statistics threshold of PCPS algorithm.*
- void [set\\_doppler\\_max](#) (unsigned int doppler\_max) override  
*Set maximum Doppler off grid search.*
- void [set\\_doppler\\_step](#) (unsigned int doppler\_step) override  
*Set Doppler steps for the grid search.*
- void [init](#) () override  
*Initializes acquisition algorithm.*
- void [set\\_local\\_code](#) () override  
*Sets local code for GLONASS L2/CA PCPS acquisition algorithm.*
- signed int [mag](#) () override  
*Returns the maximum peak of grid search.*
- void [reset](#) () override  
*Restart acquisition algorithm.*
- void [set\\_state](#) (int state) override  
*If state = 1, it forces the block to start acquiring from the first sample.*
- void [stop\\_acquisition](#) () override  
*Stop running acquisition.*
- void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples \_\_attribute\_\_((unused))) override

### Public Member Functions inherited from [AcquisitionInterface](#)

- virtual void [set\\_doppler\\_center](#) (int doppler\_center \_\_attribute\_\_((unused)))
- virtual void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples)=0

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

## 10.172.1 Detailed Description

This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GLONASS L2 C/A signals. Definition at line 45 of file [glonass\\_l2\\_ca\\_pcps\\_acquisition.h](#).

## 10.172.2 Member Function Documentation

### 10.172.2.1 [connect\(\)](#)

```
void GlonassL2CaPcpsAcquisition::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.172.2.2 disconnect()

```
void GlonassL2CaPcpsAcquisition::disconnect (  
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.172.2.3 get\_left\_block()

```
gr::basic_block_sptr GlonassL2CaPcpsAcquisition::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.172.2.4 get\_right\_block()

```
gr::basic_block_sptr GlonassL2CaPcpsAcquisition::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.172.2.5 implementation()

```
std::string GlonassL2CaPcpsAcquisition::implementation () [inline], [override], [virtual]
```

Returns "GLONASS\_L2\_CA\_PCPS\_Acquisition".

Implements [GNSSBlockInterface](#).

Definition at line 64 of file [glonass\\_l2\\_ca\\_pcps\\_acquisition.h](#).

#### 10.172.2.6 init()

```
void GlonassL2CaPcpsAcquisition::init () [override], [virtual]
```

Initializes acquisition algorithm.

Implements [AcquisitionInterface](#).

#### 10.172.2.7 item\_size()

```
size_t GlonassL2CaPcpsAcquisition::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 69 of file [glonass\\_l2\\_ca\\_pcps\\_acquisition.h](#).

#### 10.172.2.8 mag()

```
signed int GlonassL2CaPcpsAcquisition::mag () [override], [virtual]
```

Returns the maximum peak of grid search.

Implements [AcquisitionInterface](#).

#### 10.172.2.9 reset()

```
void GlonassL2CaPcpsAcquisition::reset () [override], [virtual]
```

Restart acquisition algorithm.

Implements [AcquisitionInterface](#).

#### 10.172.2.10 role()

```
std::string GlonassL2CaPcpsAcquisition::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 56 of file [glonass\\_l2\\_ca\\_pcps\\_acquisition.h](#).

#### 10.172.2.11 set\_channel()

```
void GlonassL2CaPcpsAcquisition::set_channel (  
    unsigned int channel) [inline], [override], [virtual]
```

Set acquisition channel unique ID.

Implements [AcquisitionInterface](#).

Definition at line 89 of file [glonass\\_l2\\_ca\\_pcps\\_acquisition.h](#).

**10.172.2.12 set\_channel\_fsm()**

```
void GlonassL2CaPcpsAcquisition::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline], [override], [virtual]
```

Set channel fsm associated to this acquisition instance.

Implements [AcquisitionInterface](#).

Definition at line 98 of file [glonass\\_l2\\_ca\\_pcps\\_acquisition.h](#).

**10.172.2.13 set\_doppler\_max()**

```
void GlonassL2CaPcpsAcquisition::set_doppler_max (
    unsigned int doppler_max) [override], [virtual]
```

Set maximum Doppler off grid search.

Implements [AcquisitionInterface](#).

**10.172.2.14 set\_doppler\_step()**

```
void GlonassL2CaPcpsAcquisition::set_doppler_step (
    unsigned int doppler_step) [override], [virtual]
```

Set Doppler steps for the grid search.

Implements [AcquisitionInterface](#).

**10.172.2.15 set\_gnss\_synchro()**

```
void GlonassL2CaPcpsAcquisition::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [AcquisitionInterface](#).

**10.172.2.16 set\_local\_code()**

```
void GlonassL2CaPcpsAcquisition::set_local_code () [override], [virtual]
```

Sets local code for GLONASS L2/CA PCPS acquisition algorithm.

Implements [AcquisitionInterface](#).

**10.172.2.17 set\_resampler\_latency()**

```
void GlonassL2CaPcpsAcquisition::set_resampler_latency (
    uint32_t latency_samples __attribute__((unused))) [inline], [override]
```

Definition at line 149 of file [glonass\\_l2\\_ca\\_pcps\\_acquisition.h](#).

**10.172.2.18 set\_state()**

```
void GlonassL2CaPcpsAcquisition::set_state (
    int state) [override], [virtual]
```

If state = 1, it forces the block to start acquiring from the first sample.

Implements [AcquisitionInterface](#).

**10.172.2.19 set\_threshold()**

```
void GlonassL2CaPcpsAcquisition::set_threshold (
    float threshold) [override], [virtual]
```

Set statistics threshold of PCPS algorithm.

Implements [AcquisitionInterface](#).

**10.172.2.20 stop\_acquisition()**

```
void GlonassL2CaPcpsAcquisition::stop_acquisition () [override], [virtual]
```

Stop running acquisition.

Implements [AcquisitionInterface](#).

The documentation for this class was generated from the following file:

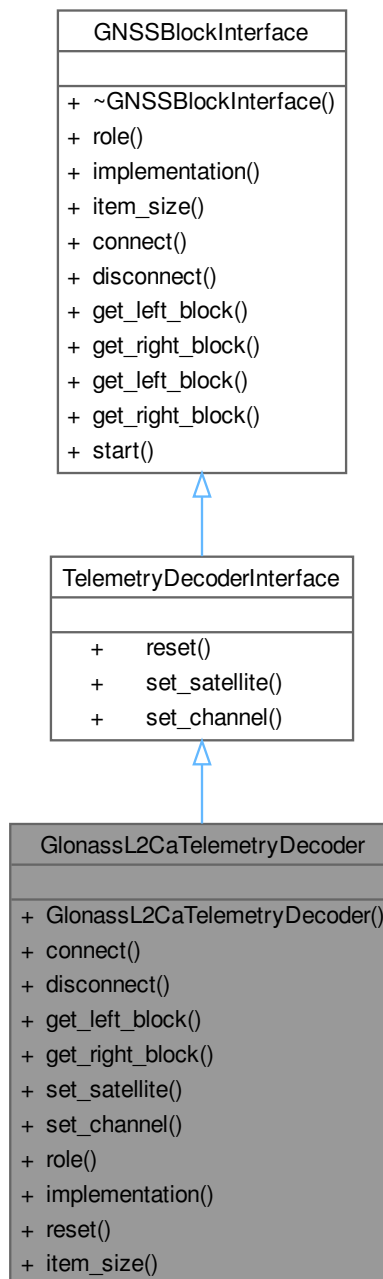
- [glonass\\_l2\\_ca\\_pcps\\_acquisition.h](#)

## 10.173 GlonassL2CaTelemetryDecoder Class Reference

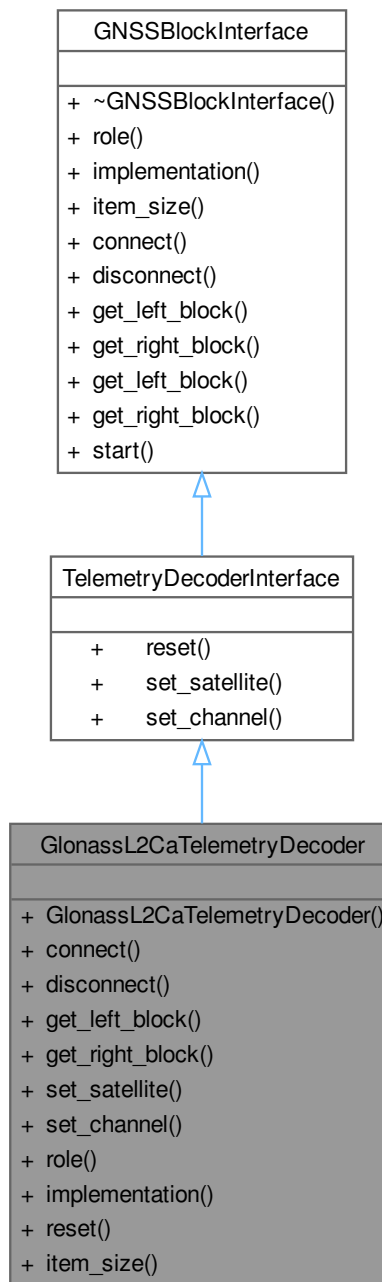
This class implements a NAV data decoder for GLONASS L2 C/A.

```
#include <glonass_l2_ca_telemetry_decoder.h>
```

Inheritance diagram for GlonassL2CaTelemetryDecoder:



Collaboration diagram for GlonassL2CaTelemetryDecoder:



### Public Member Functions

- **GlonassL2CaTelemetryDecoder** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override

- void [set\\_satellite](#) (const [Gnss\\_Satellite](#) &satellite) override
  - void [set\\_channel](#) (int channel) override
  - std::string [role](#) () override
  - std::string [implementation](#) () override
- Returns "GLONASS\_L2\_CA\_Telemetry\_Decoder".*
- void [reset](#) () override
  - size\_t [item\\_size](#) () override

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.173.1 Detailed Description

This class implements a NAV data decoder for GLONASS L2 C/A.

Definition at line 42 of file [glonass\\_l2\\_ca\\_telemetry\\_decoder.h](#).

### 10.173.2 Member Function Documentation

#### 10.173.2.1 [connect\(\)](#)

```
void GlonassL2CaTelemetryDecoder::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.173.2.2 [disconnect\(\)](#)

```
void GlonassL2CaTelemetryDecoder::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.173.2.3 [get\\_left\\_block\(\)](#)

```
gr::basic_block_sptr GlonassL2CaTelemetryDecoder::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.173.2.4 [get\\_right\\_block\(\)](#)

```
gr::basic_block_sptr GlonassL2CaTelemetryDecoder::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.173.2.5 [implementation\(\)](#)

```
std::string GlonassL2CaTelemetryDecoder::implementation () [inline], [override], [virtual]
```

Returns "GLONASS\_L2\_CA\_Telemetry\_Decoder".

Implements [GNSSBlockInterface](#).

Definition at line 65 of file [glonass\\_l2\\_ca\\_telemetry\\_decoder.h](#).

#### 10.173.2.6 [item\\_size\(\)](#)

```
size_t GlonassL2CaTelemetryDecoder::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 75 of file [glonass\\_l2\\_ca\\_telemetry\\_decoder.h](#).



### 10.173.2.7 reset()

```
void GlonassL2CaTelemetryDecoder::reset () [inline], [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

Definition at line 70 of file [glonass\\_l2\\_ca\\_telemetry\\_decoder.h](#).

### 10.173.2.8 role()

```
std::string GlonassL2CaTelemetryDecoder::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 59 of file [glonass\\_l2\\_ca\\_telemetry\\_decoder.h](#).

### 10.173.2.9 set\_channel()

```
void GlonassL2CaTelemetryDecoder::set_channel (  
    int channel) [inline], [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

Definition at line 57 of file [glonass\\_l2\\_ca\\_telemetry\\_decoder.h](#).

### 10.173.2.10 set\_satellite()

```
void GlonassL2CaTelemetryDecoder::set_satellite (  
    const Gnss\_Satellite & satellite) [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

The documentation for this class was generated from the following file:

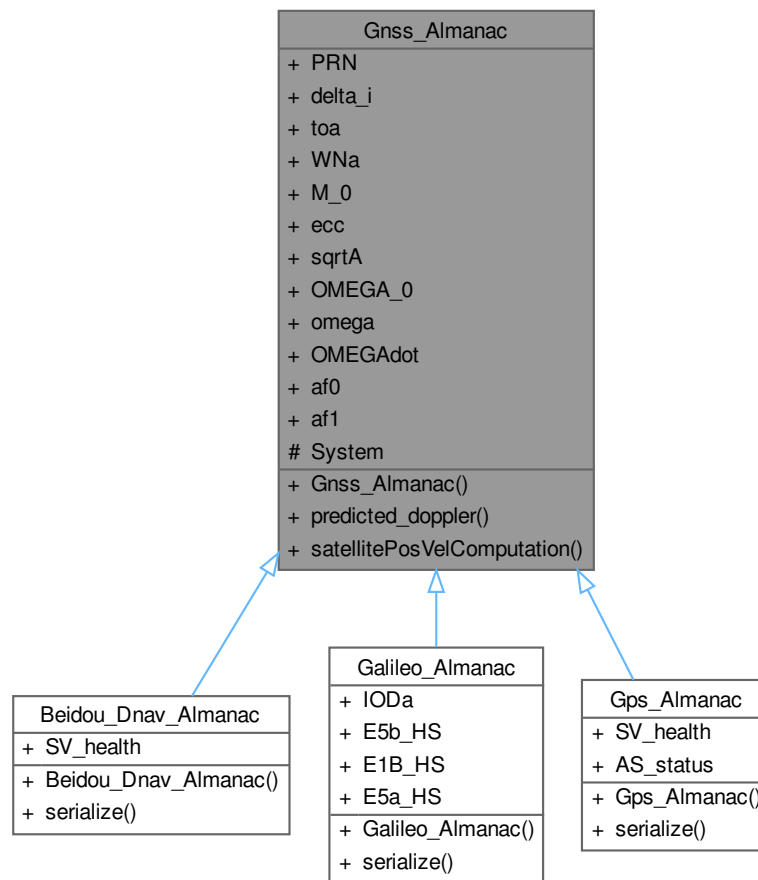
- [glonass\\_l2\\_ca\\_telemetry\\_decoder.h](#)

## 10.174 Gnss\_Almanac Class Reference

Base class for GNSS almanac storage.

```
#include <gnss_almanac.h>
```

Inheritance diagram for Gnss\_Almanac:



Collaboration diagram for Gnss\_Almanac:



### Public Member Functions

- [Gnss\\_Almanac](#) ()=default
- double [predicted\\_doppler](#) (double rx\_time\_s, double lat, double lon, double h, double ve, double vn, double vu, int band) const  
*Computes prediction of the Doppler shift for a given time and receiver's position and velocity.*
- void [satellitePosVelComputation](#) (double transmitTime, std::array< double, 7 > &pos\_vel\_dtr) const  
*Computes satellite Position and Velocity, in ECEF, for a given time (expressed in seconds of week)*

### Public Attributes

- uint32\_t [PRN](#) {}  
*SV PRN NUMBER.*
- double [delta\\_i](#) {}  
*Inclination Angle at Reference Time (relative to  $i_0 = 0.30$  semi-circles)*
- int32\_t [toa](#) {}  
*Almanac data reference time of week [s].*
- int32\_t [WNa](#) {}  
*Almanac week number.*
- double [M\\_0](#) {}  
*Mean Anomaly at Reference Time [semi-circles].*
- double [ecc](#) {}  
*Eccentricity [dimensionless].*
- double [sqrtA](#) {}

- *Square Root of the Semi-Major Axis [sqrt(m)].*
- double `OMEGA_0` {}  
*Longitude of Ascending Node of Orbit Plane at Weekly Epoch [semi-circles].*
- double `omega` {}  
*Argument of Perigee [semi-circles].*
- double `OMEGAdot` {}  
*Rate of Right Ascension [semi-circles/s].*
- double `af0` {}  
*Coefficient 0 of code phase offset model [s].*
- double `af1` {}  
*Coefficient 1 of code phase offset model [s/s].*

### Protected Attributes

- char `System` {}  
*Character ID of the GNSS system. 'G': GPS. 'E': Galileo. 'B': BeiDou.*

## 10.174.1 Detailed Description

Base class for GNSS almanac storage.

Definition at line 33 of file [gnss\\_almanac.h](#).

## 10.174.2 Constructor & Destructor Documentation

### 10.174.2.1 Gnss\_Almanac()

```
Gnss_Almanac::Gnss_Almanac () [default]
```

Default constructor

## 10.174.3 Member Function Documentation

### 10.174.3.1 predicted\_doppler()

```
double Gnss_Almanac::predicted_doppler (
    double rx_time_s,
    double lat,
    double lon,
    double h,
    double ve,
    double vn,
    double vu,
    int band) const
```

Computes prediction of the Doppler shift for a given time and receiver's position and velocity.

$$f_d = -\mathbf{v} \frac{\mathbf{x}^T}{|\mathbf{x}|} \frac{f_L}{c}$$

where:

$$\mathbf{v} = \mathbf{v}_{sat} - \mathbf{v}_{rx}$$

$$\mathbf{x} = \mathbf{x}_{sat} - \mathbf{x}_{rx}$$

$$|\mathbf{x}| = \sqrt{\mathbf{x}\mathbf{x}^T}$$

### Parameters

in	<code>rx_time_s</code>	Time of Week in seconds
in	<code>lat</code>	Receiver's latitude in degrees

## Parameters

in	<i>lon</i>	Receiver's longitude in degrees
in	<i>h</i>	Receiver's height in meters
in	<i>ve</i>	Receiver's velocity in the East direction [m/s]
in	<i>vn</i>	Receiver's velocity in the North direction [m/s]
in	<i>vu</i>	Receiver's velocity in the Up direction [m/s]
in	<i>band</i>	Signal band for which the Doppler will be computed (1: L1 C/A, E1B, BI1; 2: L2C, BI2; 3: BI3; 5: L5/E5a; 6: E6B; 7: E5b; 8: E5a+E5b)

**10.174.3.2 satellitePosVelComputation()**

```
void Gnss_Almanac::satellitePosVelComputation (
    double transmitTime,
    std::array< double, 7 > & pos_vel_dtr) const
```

Computes satellite Position and Velocity, in ECEF, for a given time (expressed in seconds of week)

**10.174.4 Member Data Documentation****10.174.4.1 af0**

```
double Gnss_Almanac::af0 {}
```

Coefficient 0 of code phase offset model [s].

Definition at line 91 of file [gnss\\_almanac.h](#).

**10.174.4.2 af1**

```
double Gnss_Almanac::af1 {}
```

Coefficient 1 of code phase offset model [s/s].

Definition at line 92 of file [gnss\\_almanac.h](#).

**10.174.4.3 delta\_i**

```
double Gnss_Almanac::delta_i {}
```

Inclination Angle at Reference Time (relative to  $i_0 = 0.30$  semi-circles)

Definition at line 82 of file [gnss\\_almanac.h](#).

**10.174.4.4 ecc**

```
double Gnss_Almanac::ecc {}
```

Eccentricity [dimensionless].

Definition at line 86 of file [gnss\\_almanac.h](#).

**10.174.4.5 M\_0**

```
double Gnss_Almanac::M_0 {}
```

Mean Anomaly at Reference Time [semi-circles].

Definition at line 85 of file [gnss\\_almanac.h](#).

**10.174.4.6 omega**

```
double Gnss_Almanac::omega {}
```

Argument of Perigee [semi-circles].

Definition at line 89 of file [gnss\\_almanac.h](#).

#### 10.174.4.7 OMEGA\_0

```
double Gnss_Almanac::OMEGA_0 {}
```

Longitude of Ascending Node of Orbit Plane at Weekly Epoch [semi-circles].

Definition at line 88 of file [gnss\\_almanac.h](#).

#### 10.174.4.8 OMEGAdot

```
double Gnss_Almanac::OMEGAdot {}
```

Rate of Right Ascension [semi-circles/s].

Definition at line 90 of file [gnss\\_almanac.h](#).

#### 10.174.4.9 PRN

```
uint32_t Gnss_Almanac::PRN {}
```

SV PRN NUMBER.

Definition at line 81 of file [gnss\\_almanac.h](#).

#### 10.174.4.10 sqrtA

```
double Gnss_Almanac::sqrtA {}
```

Square Root of the Semi-Major Axis [sqrt(m)].

Definition at line 87 of file [gnss\\_almanac.h](#).

#### 10.174.4.11 System

```
char Gnss_Almanac::System {} [protected]
```

Character ID of the GNSS system. 'G': GPS. 'E': Galileo. 'B': BeiDou.

Definition at line 95 of file [gnss\\_almanac.h](#).

Referenced by [Beidou\\_Dnav\\_Almanac::Beidou\\_Dnav\\_Almanac\(\)](#), [Galileo\\_Almanac::Galileo\\_Almanac\(\)](#), and [Gps\\_Almanac::Gps\\_Almanac\(\)](#).

#### 10.174.4.12 toa

```
int32_t Gnss_Almanac::toa {}
```

Almanac data reference time of week [s].

Definition at line 83 of file [gnss\\_almanac.h](#).

#### 10.174.4.13 WNa

```
int32_t Gnss_Almanac::WNa {}
```

Almanac week number.

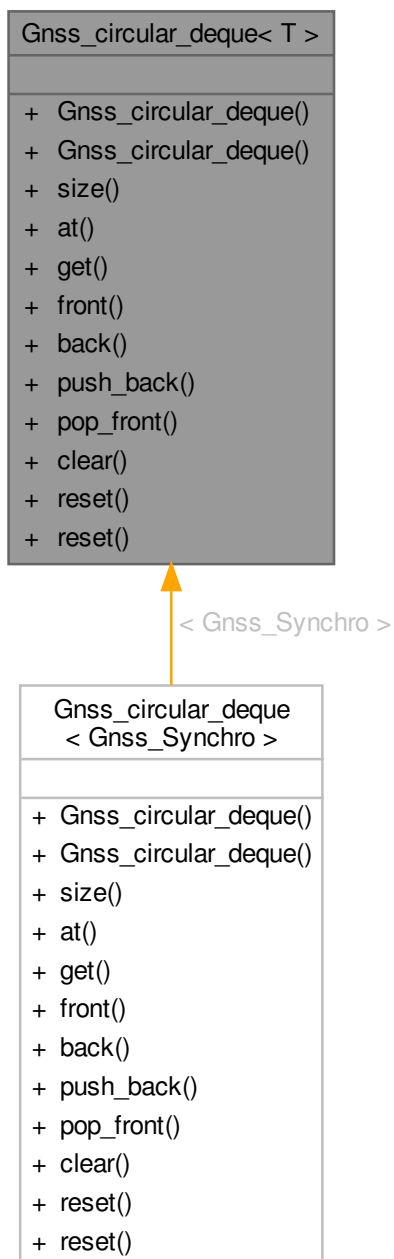
Definition at line 84 of file [gnss\\_almanac.h](#).

The documentation for this class was generated from the following file:

- [gnss\\_almanac.h](#)

## 10.175 Gnss\_circular\_deque< T > Class Template Reference

Inheritance diagram for Gnss\_circular\_deque< T >:



Collaboration diagram for `Gnss_circular_deque< T >`:

Gnss_circular_deque< T >
<ul style="list-style-type: none"> <li>+ Gnss_circular_deque()</li> <li>+ Gnss_circular_deque()</li> <li>+ size()</li> <li>+ at()</li> <li>+ get()</li> <li>+ front()</li> <li>+ back()</li> <li>+ push_back()</li> <li>+ pop_front()</li> <li>+ clear()</li> <li>+ reset()</li> <li>+ reset()</li> </ul>

## Public Member Functions

- [Gnss\\_circular\\_deque](#) ()  
*Default constructor.*
- [Gnss\\_circular\\_deque](#) (unsigned int max\_size, unsigned int nchann)  
*nchann = number of channels; max\_size = channel capacity*
- unsigned int [size](#) (unsigned int ch) const  
*Returns the number of available elements in a channel.*
- T & [at](#) (unsigned int ch, unsigned int pos)  
*Returns a reference to an element with bound checking.*
- const T & [get](#) (unsigned int ch, unsigned int pos) const  
*Returns a const reference to an element without bound checking.*
- T & [front](#) (unsigned int ch)  
*Returns a reference to the first element in the deque.*
- T & [back](#) (unsigned int ch)  
*Returns a reference to the last element in the deque.*
- void [push\\_back](#) (unsigned int ch, const T &new\_data)  
*Inserts an element at the end of the deque.*
- void [pop\\_front](#) (unsigned int ch)  
*Removes the first element of the deque.*
- void [clear](#) (unsigned int ch)  
*Removes all the elements of the deque (Sets size to 0). Capacity is not modified.*
- void [reset](#) (unsigned int max\_size, unsigned int nchann)  
*Removes all the elements in all the channels. Re-sets the number of channels and their capacity.*
- void [reset](#) ()  
*Removes all the channels (Sets nchann to 0)*



### 10.175.1 Detailed Description

```
template<class T>
class Gnss_circular_deque< T >
```

Definition at line 31 of file [gnss\\_circular\\_deque.h](#).

The documentation for this class was generated from the following file:

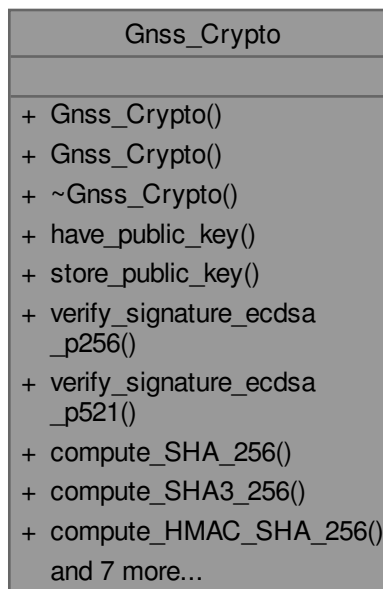
- [gnss\\_circular\\_deque.h](#)

## 10.176 Gnss\_Crypto Class Reference

Class implementing cryptographic functions for Navigation Message Authentication.

```
#include <gnss_crypto.h>
```

Collaboration diagram for Gnss\_Crypto:



### Public Member Functions

- [Gnss\\_Crypto](#) ()  
*Default constructor.*
- [Gnss\\_Crypto](#) (const std::string &certFilePath, const std::string &merkleTreePath)
- [~Gnss\\_Crypto](#) ()  
*Default destructor.*
- bool [have\\_public\\_key](#) () const  
*Returns true if the ECDSA Public Key is already loaded.*
- bool [store\\_public\\_key](#) (const std::string &pubKeyFilePath) const
- bool [verify\\_signature\\_ecdsa\\_p256](#) (const std::vector< uint8\_t > &message, const std::vector< uint8\_t > &signature) const  
*Verify ECDSA-P256 signature (message in plain hex, signature in raw format)*
- bool [verify\\_signature\\_ecdsa\\_p521](#) (const std::vector< uint8\_t > &message, const std::vector< uint8\_t > &signature) const

- Verify ECDSA-P521 signature (message in plain hex, signature in raw format)*

  - `std::vector< uint8_t > compute_SHA_256` (const `std::vector< uint8_t > &input`) const

*Computes SHA-256 hash.*
- `std::vector< uint8_t > compute_SHA3_256` (const `std::vector< uint8_t > &input`) const

*Computes SHA3-256 hash.*
- `std::vector< uint8_t > compute_HMAC_SHA_256` (const `std::vector< uint8_t > &key`, const `std::vector< uint8_t > &input`) const

*Computes HMAC-SHA-256 message authentication code.*
- `std::vector< uint8_t > compute_CMAC_AES` (const `std::vector< uint8_t > &key`, const `std::vector< uint8_t > &input`) const

*Computes CMAC-AES message authentication code.*
- `std::vector< uint8_t > get_merkle_root` () const

*Gets the Merkle Tree root node ( $x_{4,0}$ )*
- `std::string get_public_key_type` () const

*Gets the ECDSA Public Key type (ECDSA P-256 / ECDSA P-521 / Unknown)*
- void `set_public_key` (const `std::vector< uint8_t > &publickey`)

*Sets the ECDSA Public Key (publickey compressed format)*
- void `set_public_key_type` (const `std::string &public_key_type`)

*Sets the ECDSA Public Key type (ECDSA P-256 / ECDSA P-521)*
- void `set_merkle_root` (const `std::vector< uint8_t > &v`)

*Sets the Merkle Tree root node  $x(x_{4,0})$*
- void `read_merkle_xml` (const `std::string &merkleFilePath`)

### 10.176.1 Detailed Description

Class implementing cryptographic functions for Navigation Message Authentication.  
Definition at line 41 of file [gnss\\_crypto.h](#).

### 10.176.2 Constructor & Destructor Documentation

#### 10.176.2.1 Gnss\_Crypto() [1/2]

`Gnss_Crypto::Gnss_Crypto ()`  
Default constructor.

#### 10.176.2.2 Gnss\_Crypto() [2/2]

`Gnss_Crypto::Gnss_Crypto (`  
`const std::string & certFilePath,`  
`const std::string & merkleTreePath)`

Constructor with a .crt or .pem file for the ECDSA Public Key and a XML file for the Merkle Tree root. Files can be downloaded by registering at <https://www.gsc-europa.eu/>

#### 10.176.2.3 ~Gnss\_Crypto()

`Gnss_Crypto::~Gnss_Crypto ()`  
Default destructor.

### 10.176.3 Member Function Documentation

#### 10.176.3.1 compute\_CMAC\_AES()

`std::vector< uint8_t > Gnss_Crypto::compute_CMAC_AES (`  
`const std::vector< uint8_t > & key,`  
`const std::vector< uint8_t > & input) const`

Computes CMAC-AES message authentication code.

**10.176.3.2 compute\_HMAC\_SHA\_256()**

```
std::vector< uint8_t > Gnss_Crypto::compute_HMAC_SHA_256 (
    const std::vector< uint8_t > & key,
    const std::vector< uint8_t > & input) const
```

Computes HMAC-SHA-256 message authentication code.

**10.176.3.3 compute\_SHA3\_256()**

```
std::vector< uint8_t > Gnss_Crypto::compute_SHA3_256 (
    const std::vector< uint8_t > & input) const
```

Computes SHA3-256 hash.

**10.176.3.4 compute\_SHA\_256()**

```
std::vector< uint8_t > Gnss_Crypto::compute_SHA_256 (
    const std::vector< uint8_t > & input) const
```

Computes SHA-256 hash.

**10.176.3.5 get\_merkle\_root()**

```
std::vector< uint8_t > Gnss_Crypto::get_merkle_root () const
```

Gets the Merkle Tree root node ( $x_{4,0}$ )

**10.176.3.6 get\_public\_key\_type()**

```
std::string Gnss_Crypto::get_public_key_type () const
```

Gets the ECDSA Public Key type (ECDSA P-256 / ECDSA P-521 / Unknown)

**10.176.3.7 have\_public\_key()**

```
bool Gnss_Crypto::have_public_key () const
```

Returns true if the ECDSA Public Key is already loaded.

**10.176.3.8 set\_merkle\_root()**

```
void Gnss_Crypto::set_merkle_root (
    const std::vector< uint8_t > & v)
```

Sets the Merkle Tree root node  $x(x_{4,0})$

**10.176.3.9 set\_public\_key()**

```
void Gnss_Crypto::set_public_key (
    const std::vector< uint8_t > & publickey)
```

Sets the ECDSA Public Key (publickey compressed format)

**10.176.3.10 set\_public\_key\_type()**

```
void Gnss_Crypto::set_public_key_type (
    const std::string & public_key_type)
```

Sets the ECDSA Public Key type (ECDSA P-256 / ECDSA P-521)

**10.176.3.11 store\_public\_key()**

```
bool Gnss_Crypto::store_public_key (
    const std::string & pubKeyFilePath) const
```

Stores the ECDSA Public Key in a .pem file, which is read in a following run if the .crt file is not found

### 10.176.3.12 verify\_signature\_ecdsa\_p256()

```
bool Gnss_Crypto::verify_signature_ecdsa_p256 (
    const std::vector< uint8_t > & message,
    const std::vector< uint8_t > & signature) const
```

Verify ECDSA-P256 signature (message in plain hex, signature in raw format)

### 10.176.3.13 verify\_signature\_ecdsa\_p521()

```
bool Gnss_Crypto::verify_signature_ecdsa_p521 (
    const std::vector< uint8_t > & message,
    const std::vector< uint8_t > & signature) const
```

Verify ECDSA-P521 signature (message in plain hex, signature in raw format)  
The documentation for this class was generated from the following file:

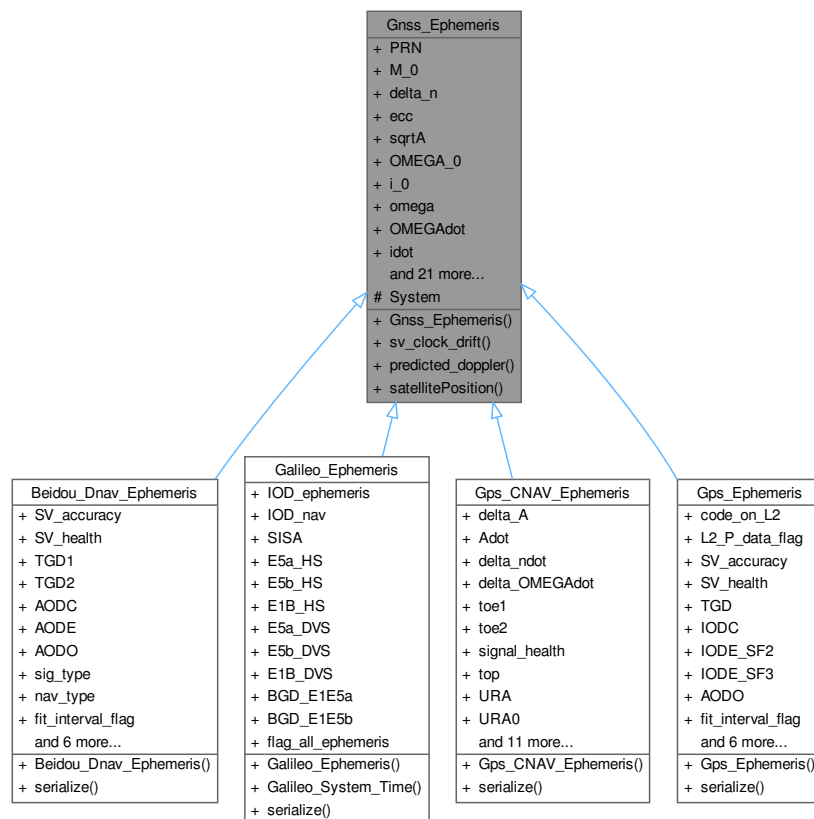
- [gnss\\_crypto.h](#)

## 10.177 Gnss\_Ephemeris Class Reference

Base class for GNSS ephemeris storage.

```
#include <gnss_ephemeris.h>
```

Inheritance diagram for Gnss\_Ephemeris:



Collaboration diagram for Gnss\_Ephemeris:

Gnss_Ephemeris
+ PRN
+ M_0
+ delta_n
+ ecc
+ sqrtA
+ OMEGA_0
+ i_0
+ omega
+ OMEGAdot
+ idot
and 21 more...
# System
+ Gnss_Ephemeris()
+ sv_clock_drift()
+ predicted_doppler()
+ satellitePosition()

### Public Member Functions

- double [sv\\_clock\\_drift](#) (double transmitTime)  
*Sets (satClkDrift) and (dtr), and returns the clock drift in seconds according to the User Algorithm for SV Clock Correction (IS-GPS-200M, 20.3.3.3.3.1, and Galileo OS SIS ICD, 5.1.4).*
- double [predicted\\_doppler](#) (double rx\_time\_s, double lat, double lon, double h, double ve, double vn, double vu, int band) const  
*Computes prediction of the Doppler shift for a given time and receiver's position and velocity.*
- void [satellitePosition](#) (double transmitTime)  
*Computes the ECEF SV coordinates and ECEF velocity.*

### Public Attributes

- uint32\_t [PRN](#) {}  
*SV ID.*
- double [M\\_0](#) {}  
*Mean anomaly at reference time [rad].*
- double [delta\\_n](#) {}  
*Mean motion difference from computed value [rad/sec].*
- double [ecc](#) {}  
*Eccentricity.*
- double [sqrtA](#) {}  
*Square root of the semi-major axis [meters<sup>1/2</sup>].*
- double [OMEGA\\_0](#) {}

- Longitude of ascending node of orbital plane at weekly epoch [rad].*
- double `i_0` {}
- Inclination angle at reference time [rad].*
- double `omega` {}
- Argument of perigee [rad].*
- double `OMEGAdot` {}
- Rate of right ascension [rad/sec].*
- double `idot` {}
- Rate of inclination angle [rad/sec].*
- double `Cuc` {}
- Amplitude of the cosine harmonic correction term to the argument of latitude [rad].*
- double `Cus` {}
- Amplitude of the sine harmonic correction term to the argument of latitude [rad].*
- double `Crc` {}
- Amplitude of the cosine harmonic correction term to the orbit radius [meters].*
- double `Crs` {}
- Amplitude of the sine harmonic correction term to the orbit radius [meters].*
- double `Cic` {}
- Amplitude of the cosine harmonic correction term to the angle of inclination [rad].*
- double `Cis` {}
- Amplitude of the sine harmonic correction term to the angle of inclination [rad].*
- int32\_t `toe` {}
- Ephemeris reference time [s].*
- int32\_t `toc` {}
- Clock correction data reference Time of Week [sec].*
- double `af0` {}
- SV clock bias correction coefficient [s].*
- double `af1` {}
- SV clock drift correction coefficient [s/s].*
- double `af2` {}
- SV clock drift rate correction coefficient [s/s<sup>2</sup>].*
- double `satClkDrift` {}
- SV clock drift.*
- double `dtr` {}
- Relativistic clock correction term.*
- int32\_t `WN` {}
- Week number.*
- int32\_t `tow` {}
- Time of Week.*
- double `satpos_X` {}
- Earth-fixed coordinate x of the satellite [m]. Intersection of the IERS Reference Meridian (IRM) and the plane passing through the origin and normal to the Z-axis.*
- double `satpos_Y` {}
- Earth-fixed coordinate y of the satellite [m]. Completes a right-handed, Earth-Centered, Earth-Fixed orthogonal coordinate system.*
- double `satpos_Z` {}
- Earth-fixed coordinate z of the satellite [m]. The direction of the IERS (International Earth Rotation and Reference Systems Service) Reference Pole (IRP).*
- double `satvel_X` {}
- Earth-fixed velocity coordinate x of the satellite [m].*
- double `satvel_Y` {}
- Earth-fixed velocity coordinate y of the satellite [m].*
- double `satvel_Z` {}
- Earth-fixed velocity coordinate z of the satellite [m].*

## Protected Attributes

- char [System](#) {}

Character ID of the GNSS system. 'G': GPS. 'E': Galileo. 'B': BeiDou.

## 10.177.1 Detailed Description

Base class for GNSS ephemeris storage.

Definition at line 28 of file [gnss\\_ephemeris.h](#).

## 10.177.2 Member Function Documentation

### 10.177.2.1 predicted\_doppler()

```
double Gnss_Ephemeris::predicted_doppler (
    double rx_time_s,
    double lat,
    double lon,
    double h,
    double ve,
    double vn,
    double vu,
    int band) const
```

Computes prediction of the Doppler shift for a given time and receiver's position and velocity.

$$f_d = -\mathbf{v} \frac{\mathbf{x}^T}{|\mathbf{x}|} \frac{f_L}{c}$$

where:

$$\mathbf{v} = \mathbf{v}_{sat} - \mathbf{v}_{rx}$$

$$\mathbf{x} = \mathbf{x}_{sat} - \mathbf{x}_{rx}$$

$$|\mathbf{x}| = \sqrt{\mathbf{x}\mathbf{x}^T}$$

### Parameters

in	<i>rx_time_s</i>	Time of Week in seconds
in	<i>lat</i>	Receiver's latitude in degrees
in	<i>lon</i>	Receiver's longitude in degrees
in	<i>h</i>	Receiver's height in meters
in	<i>ve</i>	Receiver's velocity in the East direction [m/s]
in	<i>vn</i>	Receiver's velocity in the North direction [m/s]
in	<i>vu</i>	Receiver's velocity in the Up direction [m/s]
in	<i>band</i>	Signal band for which the Doppler will be computed (1: L1 C/A, E1B, BI1; 2: L2C, BI2; 3: BI3; 5: L5/E5a; 6: E6B; 7: E5b; 8: E5a+E5b)

### 10.177.2.2 satellitePosition()

```
void Gnss_Ephemeris::satellitePosition (
    double transmitTime)
```

Computes the ECEF SV coordinates and ECEF velocity.

### 10.177.2.3 sv\_clock\_drift()

```
double Gnss_Ephemeris::sv_clock_drift (
    double transmitTime)
```

Sets (*satClkDrift*) and (*dtr*), and returns the clock drift in seconds according to the User Algorithm for SV Clock Correction (IS-GPS-200M, 20.3.3.3.1, and Galileo OS SIS ICD, 5.1.4).

## 10.177.3 Member Data Documentation

### 10.177.3.1 af0

```
double Gnss_Ephemeris::af0 {}
```

SV clock bias correction coefficient [s].

Definition at line 90 of file [gnss\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), [Beidou\\_Dnav\\_Ephemeris::serialize\(\)](#), [Galileo\\_Ephemeris::serialize\(\)](#), [Gps\\_CNAV\\_Ephemeris::serialize\(\)](#), and [Gps\\_Ephemeris::serialize\(\)](#).

### 10.177.3.2 af1

```
double Gnss_Ephemeris::af1 {}
```

SV clock drift correction coefficient [s/s].

Definition at line 91 of file [gnss\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), [Beidou\\_Dnav\\_Ephemeris::serialize\(\)](#), [Galileo\\_Ephemeris::serialize\(\)](#), [Gps\\_CNAV\\_Ephemeris::serialize\(\)](#), and [Gps\\_Ephemeris::serialize\(\)](#).

### 10.177.3.3 af2

```
double Gnss_Ephemeris::af2 {}
```

SV clock drift rate correction coefficient [s/s<sup>2</sup>].

Definition at line 92 of file [gnss\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), [Beidou\\_Dnav\\_Ephemeris::serialize\(\)](#), [Galileo\\_Ephemeris::serialize\(\)](#), [Gps\\_CNAV\\_Ephemeris::serialize\(\)](#), and [Gps\\_Ephemeris::serialize\(\)](#).

### 10.177.3.4 Cic

```
double Gnss_Ephemeris::Cic {}
```

Amplitude of the cosine harmonic correction term to the angle of inclination [rad].

Definition at line 84 of file [gnss\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), [Beidou\\_Dnav\\_Ephemeris::serialize\(\)](#), [Galileo\\_Ephemeris::serialize\(\)](#), [Gps\\_CNAV\\_Ephemeris::serialize\(\)](#), and [Gps\\_Ephemeris::serialize\(\)](#).

### 10.177.3.5 Cis

```
double Gnss_Ephemeris::Cis {}
```

Amplitude of the sine harmonic correction term to the angle of inclination [rad].

Definition at line 85 of file [gnss\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), [Beidou\\_Dnav\\_Ephemeris::serialize\(\)](#), [Galileo\\_Ephemeris::serialize\(\)](#), [Gps\\_CNAV\\_Ephemeris::serialize\(\)](#), and [Gps\\_Ephemeris::serialize\(\)](#).

### 10.177.3.6 Crc

```
double Gnss_Ephemeris::Crc {}
```

Amplitude of the cosine harmonic correction term to the orbit radius [meters].

Definition at line 82 of file [gnss\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), [Beidou\\_Dnav\\_Ephemeris::serialize\(\)](#), [Galileo\\_Ephemeris::serialize\(\)](#), [Gps\\_CNAV\\_Ephemeris::serialize\(\)](#), and [Gps\\_Ephemeris::serialize\(\)](#).



### 10.177.3.7 Crs

```
double Gnss_Ephemeris::Crs {}
```

Amplitude of the sine harmonic correction term to the orbit radius [meters].

Definition at line 83 of file [gnss\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), [Beidou\\_Dnav\\_Ephemeris::serialize\(\)](#), [Galileo\\_Ephemeris::serialize\(\)](#), [Gps\\_CNAV\\_Ephemeris::serialize\(\)](#), and [Gps\\_Ephemeris::serialize\(\)](#).

### 10.177.3.8 Cuc

```
double Gnss_Ephemeris::Cuc {}
```

Amplitude of the cosine harmonic correction term to the argument of latitude [rad].

Definition at line 80 of file [gnss\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), [Beidou\\_Dnav\\_Ephemeris::serialize\(\)](#), [Galileo\\_Ephemeris::serialize\(\)](#), [Gps\\_CNAV\\_Ephemeris::serialize\(\)](#), and [Gps\\_Ephemeris::serialize\(\)](#).

### 10.177.3.9 Cus

```
double Gnss_Ephemeris::Cus {}
```

Amplitude of the sine harmonic correction term to the argument of latitude [rad].

Definition at line 81 of file [gnss\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), [Beidou\\_Dnav\\_Ephemeris::serialize\(\)](#), [Galileo\\_Ephemeris::serialize\(\)](#), [Gps\\_CNAV\\_Ephemeris::serialize\(\)](#), and [Gps\\_Ephemeris::serialize\(\)](#).

### 10.177.3.10 delta\_n

```
double Gnss_Ephemeris::delta_n {}
```

Mean motion difference from computed value [rad/sec].

Definition at line 72 of file [gnss\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), [Beidou\\_Dnav\\_Ephemeris::serialize\(\)](#), [Galileo\\_Ephemeris::serialize\(\)](#), [Gps\\_CNAV\\_Ephemeris::serialize\(\)](#), and [Gps\\_Ephemeris::serialize\(\)](#).

### 10.177.3.11 dtr

```
double Gnss_Ephemeris::dtr {}
```

Relativistic clock correction term.

Definition at line 95 of file [gnss\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), [Beidou\\_Dnav\\_Ephemeris::serialize\(\)](#), [Galileo\\_Ephemeris::serialize\(\)](#), [Gps\\_CNAV\\_Ephemeris::serialize\(\)](#), and [Gps\\_Ephemeris::serialize\(\)](#).

### 10.177.3.12 ecc

```
double Gnss_Ephemeris::ecc {}
```

Eccentricity.

Definition at line 73 of file [gnss\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), [Beidou\\_Dnav\\_Ephemeris::serialize\(\)](#), [Galileo\\_Ephemeris::serialize\(\)](#), [Gps\\_CNAV\\_Ephemeris::serialize\(\)](#), and [Gps\\_Ephemeris::serialize\(\)](#).

### 10.177.3.13 i\_0

```
double Gnss_Ephemeris::i_0 {}
```

Inclination angle at reference time [rad].

Definition at line 76 of file [gnss\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), [Beidou\\_Dnav\\_Ephemeris::serialize\(\)](#), [Galileo\\_Ephemeris::serialize\(\)](#), [Gps\\_CNAV\\_Ephemeris::serialize\(\)](#), and [Gps\\_Ephemeris::serialize\(\)](#).

### 10.177.3.14 idot

```
double Gnss_Ephemeris::idot {}
```

Rate of inclination angle [rad/sec].

Definition at line 79 of file [gnss\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), [Beidou\\_Dnav\\_Ephemeris::serialize\(\)](#), [Galileo\\_Ephemeris::serialize\(\)](#), [Gps\\_CNAV\\_Ephemeris::serialize\(\)](#), and [Gps\\_Ephemeris::serialize\(\)](#).

#### 10.177.3.15 M\_0

```
double Gnss_Ephemeris::M_0 {}
```

Mean anomaly at reference time [rad].

Definition at line 71 of file [gnss\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), [Beidou\\_Dnav\\_Ephemeris::serialize\(\)](#), [Galileo\\_Ephemeris::serialize\(\)](#), [Gps\\_CNAV\\_Ephemeris::serialize\(\)](#), and [Gps\\_Ephemeris::serialize\(\)](#).

#### 10.177.3.16 omega

```
double Gnss_Ephemeris::omega {}
```

Argument of perigee [rad].

Definition at line 77 of file [gnss\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), [Beidou\\_Dnav\\_Ephemeris::serialize\(\)](#), [Galileo\\_Ephemeris::serialize\(\)](#), [Gps\\_CNAV\\_Ephemeris::serialize\(\)](#), and [Gps\\_Ephemeris::serialize\(\)](#).

#### 10.177.3.17 OMEGA\_0

```
double Gnss_Ephemeris::OMEGA_0 {}
```

Longitude of ascending node of orbital plane at weekly epoch [rad].

Definition at line 75 of file [gnss\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), [Beidou\\_Dnav\\_Ephemeris::serialize\(\)](#), [Galileo\\_Ephemeris::serialize\(\)](#), [Gps\\_CNAV\\_Ephemeris::serialize\(\)](#), and [Gps\\_Ephemeris::serialize\(\)](#).

#### 10.177.3.18 OMEGAdot

```
double Gnss_Ephemeris::OMEGAdot {}
```

Rate of right ascension [rad/sec].

Definition at line 78 of file [gnss\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), [Beidou\\_Dnav\\_Ephemeris::serialize\(\)](#), [Galileo\\_Ephemeris::serialize\(\)](#), [Gps\\_CNAV\\_Ephemeris::serialize\(\)](#), and [Gps\\_Ephemeris::serialize\(\)](#).

#### 10.177.3.19 PRN

```
uint32_t Gnss_Ephemeris::PRN {}
```

SV ID.

Definition at line 70 of file [gnss\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), [Beidou\\_Dnav\\_Ephemeris::serialize\(\)](#), [Galileo\\_Ephemeris::serialize\(\)](#), [Gps\\_CNAV\\_Ephemeris::serialize\(\)](#), and [Gps\\_Ephemeris::serialize\(\)](#).

#### 10.177.3.20 satClkDrift

```
double Gnss_Ephemeris::satClkDrift {}
```

SV clock drift.

Definition at line 94 of file [gnss\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), [Beidou\\_Dnav\\_Ephemeris::serialize\(\)](#), [Galileo\\_Ephemeris::serialize\(\)](#), [Gps\\_CNAV\\_Ephemeris::serialize\(\)](#), and [Gps\\_Ephemeris::serialize\(\)](#).

#### 10.177.3.21 satpos\_X

```
double Gnss_Ephemeris::satpos_X {}
```

Earth-fixed coordinate x of the satellite [m]. Intersection of the IERS Reference Meridian (IRM) and the plane passing through the origin and normal to the Z-axis.

Definition at line 102 of file [gnss\\_ephemeris.h](#).

### 10.177.3.22 satpos\_Y

```
double Gnss_Ephemeris::satpos_Y {}
```

Earth-fixed coordinate y of the satellite [m]. Completes a right-handed, Earth-Centered, Earth-Fixed orthogonal coordinate system.

Definition at line 103 of file [gnss\\_ephemeris.h](#).

### 10.177.3.23 satpos\_Z

```
double Gnss_Ephemeris::satpos_Z {}
```

Earth-fixed coordinate z of the satellite [m]. The direction of the IERS (International Earth Rotation and Reference Systems Service) Reference Pole (IRP).

Definition at line 104 of file [gnss\\_ephemeris.h](#).

### 10.177.3.24 satvel\_X

```
double Gnss_Ephemeris::satvel_X {}
```

Earth-fixed velocity coordinate x of the satellite [m].

Definition at line 107 of file [gnss\\_ephemeris.h](#).

### 10.177.3.25 satvel\_Y

```
double Gnss_Ephemeris::satvel_Y {}
```

Earth-fixed velocity coordinate y of the satellite [m].

Definition at line 108 of file [gnss\\_ephemeris.h](#).

### 10.177.3.26 satvel\_Z

```
double Gnss_Ephemeris::satvel_Z {}
```

Earth-fixed velocity coordinate z of the satellite [m].

Definition at line 109 of file [gnss\\_ephemeris.h](#).

### 10.177.3.27 sqrtA

```
double Gnss_Ephemeris::sqrtA {}
```

Square root of the semi-major axis [meters<sup>1/2</sup>].

Definition at line 74 of file [gnss\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), [Beidou\\_Dnav\\_Ephemeris::serialize\(\)](#), [Galileo\\_Ephemeris::serialize\(\)](#), [Gps\\_CNAV\\_Ephemeris::serialize\(\)](#), and [Gps\\_Ephemeris::serialize\(\)](#).

### 10.177.3.28 System

```
char Gnss_Ephemeris::System {} [protected]
```

Character ID of the GNSS system. 'G': GPS. 'E': Galileo. 'B': BeiDou.

Definition at line 112 of file [gnss\\_ephemeris.h](#).

Referenced by [Gps\\_CNAV\\_Ephemeris::Gps\\_CNAV\\_Ephemeris\(\)](#).

### 10.177.3.29 toc

```
int32_t Gnss_Ephemeris::toc {}
```

Clock correction data reference Time of Week [sec].

Definition at line 89 of file [gnss\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), [Beidou\\_Dnav\\_Ephemeris::serialize\(\)](#), [Galileo\\_Ephemeris::serialize\(\)](#), [Gps\\_CNAV\\_Ephemeris::serialize\(\)](#), and [Gps\\_Ephemeris::serialize\(\)](#).

### 10.177.3.30 toe

```
int32_t Gnss_Ephemeris::toe {}
```

Ephemeris reference time [s].

Definition at line 86 of file [gnss\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), [Beidou\\_Dnav\\_Ephemeris::serialize\(\)](#), [Galileo\\_Ephemeris::serialize\(\)](#), [Gps\\_CNAV\\_Ephemeris::serialize\(\)](#), and [Gps\\_Ephemeris::serialize\(\)](#).

#### 10.177.3.31 tow

```
int32_t Gnss_Ephemeris::tow {}
```

Time of Week.

Definition at line 99 of file [gnss\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), [Beidou\\_Dnav\\_Ephemeris::serialize\(\)](#), [Galileo\\_Ephemeris::serialize\(\)](#), [Gps\\_CNAV\\_Ephemeris::serialize\(\)](#), and [Gps\\_Ephemeris::serialize\(\)](#).

#### 10.177.3.32 WN

```
int32_t Gnss_Ephemeris::WN {}
```

Week number.

Definition at line 98 of file [gnss\\_ephemeris.h](#).

Referenced by [Serdes\\_Galileo\\_Eph::readProtobuffer\(\)](#), [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), [Beidou\\_Dnav\\_Ephemeris::serialize\(\)](#), [Galileo\\_Ephemeris::serialize\(\)](#), [Gps\\_CNAV\\_Ephemeris::serialize\(\)](#), and [Gps\\_Ephemeris::serialize\(\)](#).

The documentation for this class was generated from the following file:

- [gnss\\_ephemeris.h](#)

## 10.178 Gnss\_Satellite Class Reference

This class represents a GNSS satellite.

```
#include <gnss_satellite.h>
```

Collaboration diagram for Gnss\_Satellite:

Gnss_Satellite
<ul style="list-style-type: none"> <li>+ Gnss_Satellite()</li> <li>+ Gnss_Satellite()</li> <li>+ ~Gnss_Satellite()</li> <li>+ Gnss_Satellite()</li> <li>+ operator=()</li> <li>+ Gnss_Satellite()</li> <li>+ operator=()</li> <li>+ update_PRN()</li> <li>+ get_PRN()</li> <li>+ get_rf_link()</li> <li>+ get_system()</li> <li>+ get_system_short()</li> <li>+ get_block()</li> <li>+ what_block()</li> </ul>

**Public Member Functions**

- [Gnss\\_Satellite](#) ()=default  
*Default Constructor.*
- [Gnss\\_Satellite](#) (const std::string &system\_, uint32\_t PRN\_)  
*Concrete GNSS satellite Constructor.*
- [~Gnss\\_Satellite](#) ()=default  
*Default Destructor.*
- [Gnss\\_Satellite](#) (const [Gnss\\_Satellite](#) &other) noexcept  
*Copy constructor.*
- [Gnss\\_Satellite](#) & [operator=](#) (const [Gnss\\_Satellite](#) &) noexcept  
*Copy assignment operator.*
- [Gnss\\_Satellite](#) ([Gnss\\_Satellite](#) &&other) noexcept  
*Move constructor.*
- [Gnss\\_Satellite](#) & [operator=](#) ([Gnss\\_Satellite](#) &&other) noexcept  
*Move assignment operator.*
- void [update\\_PRN](#) (uint32\_t PRN)  
*Updates the PRN Number when information is decoded, only applies to GLONASS GNAV messages.*
- uint32\_t [get\\_PRN](#) () const  
*Gets satellite's PRN.*
- int32\_t [get\\_rf\\_link](#) () const  
*Gets the satellite's rf link.*
- std::string [get\\_system](#) () const  
*Gets the satellite system {"GPS", "GLONASS", "SBAS", "Galileo", "Beidou"}.*
- std::string [get\\_system\\_short](#) () const  
*Gets the satellite system {"G", "R", "SBAS", "E", "C"}.*
- std::string [get\\_block](#) () const  
*Gets the satellite block. If GPS, returns {"IIA", "IIR", "IIR-M", "IIF"}.*
- std::string [what\\_block](#) (const std::string &system\_, uint32\_t PRN\_)  
*Gets the block of a given satellite.*

**Friends**

- bool [operator==](#) (const [Gnss\\_Satellite](#) &, const [Gnss\\_Satellite](#) &)  
*operator== for comparison*
- std::ostream & [operator<<](#) (std::ostream &, const [Gnss\\_Satellite](#) &)  
*operator<< for pretty printing*

**10.178.1 Detailed Description**

This class represents a GNSS satellite.

It contains information about the space vehicles currently operational of GPS, Glonass, SBAS and Galileo constellations.

Definition at line 39 of file [gnss\\_satellite.h](#).

**10.178.2 Constructor & Destructor Documentation****10.178.2.1 Gnss\_Satellite() [1/4]**

```
Gnss_Satellite::Gnss_Satellite () [default]
```

Default Constructor.

Referenced by [Gnss\\_Satellite\(\)](#), [Gnss\\_Satellite\(\)](#), [operator<<](#), [operator=\(\)](#), [operator=\(\)](#), and [operator==](#).

**10.178.2.2 Gnss\_Satellite()** [2/4]

```
Gnss_Satellite::Gnss_Satellite (
    const std::string & system_,
    uint32_t PRN_)
Concrete GNSS satellite Constructor.
```

**10.178.2.3 ~Gnss\_Satellite()**

```
Gnss_Satellite::~Gnss_Satellite () [default]
Default Destructor.
```

**10.178.2.4 Gnss\_Satellite()** [3/4]

```
Gnss_Satellite::Gnss_Satellite (
    const Gnss_Satellite & other) [noexcept]
```

Copy constructor.

References [Gnss\\_Satellite\(\)](#).

Here is the call graph for this function:

**10.178.2.5 Gnss\_Satellite()** [4/4]

```
Gnss_Satellite::Gnss_Satellite (
    Gnss_Satellite && other) [noexcept]
```

Move constructor.

References [Gnss\\_Satellite\(\)](#).

Here is the call graph for this function:

**10.178.3 Member Function Documentation****10.178.3.1 get\_block()**

```
std::string Gnss_Satellite::get_block () const
Gets the satellite block. If GPS, returns {"IIA", "IIR", "IIR-M", "IIF"}.
```

### 10.178.3.2 get\_PRN()

```
uint32_t Gnss_Satellite::get_PRN () const
```

Gets satellite's PRN.

### 10.178.3.3 get\_rf\_link()

```
int32_t Gnss_Satellite::get_rf_link () const
```

Gets the satellite's rf link.

### 10.178.3.4 get\_system()

```
std::string Gnss_Satellite::get_system () const
```

Gets the satellite system {"GPS", "GLONASS", "SBAS", "Galileo", "Beidou"}.

### 10.178.3.5 get\_system\_short()

```
std::string Gnss_Satellite::get_system_short () const
```

Gets the satellite system {"G", "R", "SBAS", "E", "C"}.

### 10.178.3.6 operator=() [1/2]

```
Gnss_Satellite & Gnss_Satellite::operator= (  
    const Gnss_Satellite & ) [noexcept]
```

Copy assignment operator.

References [Gnss\\_Satellite\(\)](#).

Here is the call graph for this function:



### 10.178.3.7 operator=() [2/2]

```
Gnss_Satellite & Gnss_Satellite::operator= (  
    Gnss_Satellite && other) [noexcept]
```

Move assignment operator.

References [Gnss\\_Satellite\(\)](#).

Here is the call graph for this function:



### 10.178.3.8 update\_PRN()

```
void Gnss_Satellite::update_PRN (
    uint32_t PRN)
```

Updates the PRN Number when information is decoded, only applies to GLONASS GNAV messages.

### 10.178.3.9 what\_block()

```
std::string Gnss_Satellite::what_block (
    const std::string & system_,
    uint32_t PRN_)
```

Gets the block of a given satellite.

## 10.178.4 Friends And Related Symbol Documentation

### 10.178.4.1 operator<<

```
std::ostream & operator<< (
    std::ostream & ,
    const Gnss_Satellite & ) [friend]
```

operator<< for pretty printing

References [Gnss\\_Satellite\(\)](#).

### 10.178.4.2 operator==

```
bool operator== (
    const Gnss_Satellite & ,
    const Gnss_Satellite & ) [friend]
```

operator== for comparison

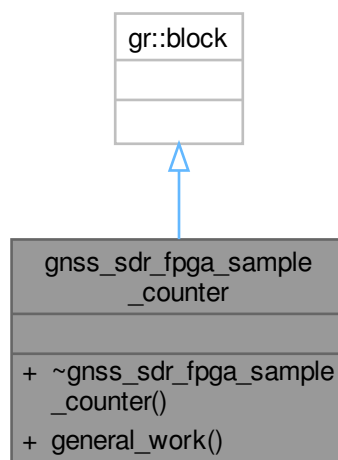
References [Gnss\\_Satellite\(\)](#).

The documentation for this class was generated from the following file:

- [gnss\\_satellite.h](#)

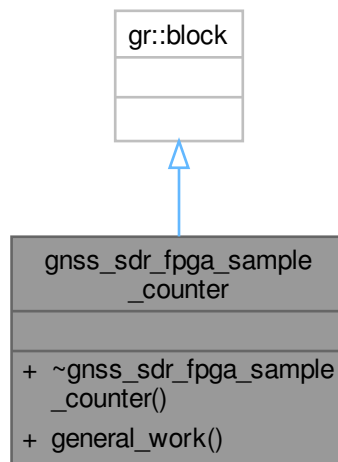
## 10.179 gnss\_sdr\_fpga\_sample\_counter Class Reference

Inheritance diagram for gnss\_sdr\_fpga\_sample\_counter:





Collaboration diagram for gnss\_sdr\_fpga\_sample\_counter:



### Public Member Functions

- int **general\_work** (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)

### Friends

- gnss\_sdr\_fpga\_sample\_counter\_sptr **gnss\_sdr\_make\_fpga\_sample\_counter** (double \_fs, int32\_t \_↔ interval\_ms)

### 10.179.1 Detailed Description

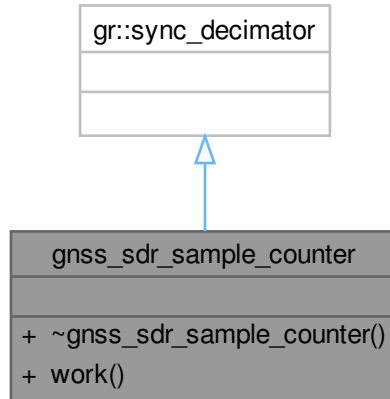
Definition at line 40 of file [gnss\\_sdr\\_fpga\\_sample\\_counter.h](#).

The documentation for this class was generated from the following file:

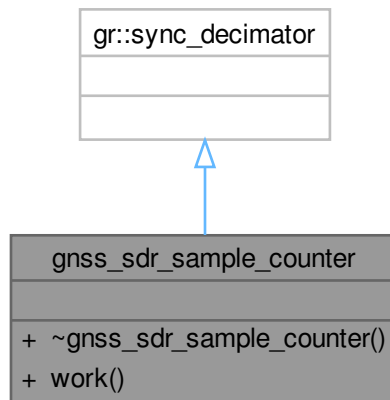
- [gnss\\_sdr\\_fpga\\_sample\\_counter.h](#)

## 10.180 gnss\_sdr\_sample\_counter Class Reference

Inheritance diagram for gnss\_sdr\_sample\_counter:



Collaboration diagram for gnss\_sdr\_sample\_counter:



### Public Member Functions

- `int` **work** (`int` noutput\_items, `gr_vector_const_void_star` &input\_items, `gr_vector_void_star` &output\_items)

### Friends

- `gnss_sdr_sample_counter_sptr` **gnss\_sdr\_make\_sample\_counter** (`double` \_fs, `int32_t` \_interval\_ms, `size_t` \_t\_size)



- bool [save\\_gal\\_ephemeris\\_map\\_xml](#) (const std::string &file\_name, std::map< int, [Galileo\\_Ephemeris](#) > eph\_map)  
*Save Galileo ephemeris map to XML file.*
- bool [load\\_gnav\\_ephemeris\\_xml](#) (const std::string &file\_name)  
*Read GLONASS GNAV ephemeris map from XML file.*
- bool [save\\_gnav\\_ephemeris\\_map\\_xml](#) (const std::string &file\_name, std::map< int, [Glonass\\_Gnav\\_Ephemeris](#) > eph\_map)  
*Save GLONASS GNAV ephemeris map to XML file.*
- bool [load\\_utc\\_xml](#) (const std::string &file\_name)  
*Read GPS utc model from XML file.*
- bool [save\\_utc\\_xml](#) (const std::string &file\_name, [Gps\\_Utc\\_Model](#) &utc)  
*Save UTC model map to XML file.*
- bool [load\\_cnav\\_utc\\_xml](#) (const std::string &file\_name)  
*Read CNAV GPS utc model from XML file.*
- bool [save\\_cnav\\_utc\\_xml](#) (const std::string &file\_name, [Gps\\_CNAV\\_Utc\\_Model](#) &utc)  
*Save CNAV UTC model map to XML file.*
- bool [load\\_gal\\_utc\\_xml](#) (const std::string &file\_name)  
*Read Galileo utc model from XML file.*
- bool [save\\_gal\\_utc\\_xml](#) (const std::string &file\_name, [Galileo\\_Utc\\_Model](#) &utc)  
*Save Galileo UTC model map to XML file.*
- bool [load\\_gal\\_almanac\\_xml](#) (const std::string &file\_name)  
*Read Galileo almanac map from XML file.*
- bool [save\\_gal\\_almanac\\_xml](#) (const std::string &file\_name, std::map< int, [Galileo\\_Almanac](#) > galileo\_↔ almanac\_map\_to\_save)  
*Save Galileo almanac map to XML file.*
- bool [load\\_gps\\_almanac\\_xml](#) (const std::string &file\_name)  
*Read GPS almanac map from XML file.*
- bool [save\\_gps\\_almanac\\_xml](#) (const std::string &file\_name, std::map< int, [Gps\\_Almanac](#) > gps\_almanac\_↔ \_map\_to\_save)  
*Save GPS almanac map to XML file.*
- bool [load\\_iono\\_xml](#) (const std::string &file\_name)  
*Read iono from XML file.*
- bool [save\\_iono\\_xml](#) (const std::string &file\_name, [Gps\\_Iono](#) &iono)  
*Save iono map to XML file.*
- bool [load\\_gal\\_iono\\_xml](#) (const std::string &file\_name)  
*Read Galileo iono from XML file.*
- bool [save\\_gal\\_iono\\_xml](#) (const std::string &file\_name, [Galileo\\_Iono](#) &iono)  
*Save Galileo iono map to XML file.*
- bool [load\\_glo\\_utc\\_xml](#) (const std::string &file\_name)  
*Read Glonass utc model from XML file.*
- bool [save\\_glo\\_utc\\_xml](#) (const std::string &file\_name, [Glonass\\_Gnav\\_Utc\\_Model](#) &utc)  
*Save Glonass UTC model map to XML file.*
- bool [load\\_ref\\_time\\_xml](#) (const std::string &file\_name)  
*Read ref time from XML file.*
- bool [save\\_ref\\_time\\_xml](#) (const std::string &file\_name, [Agnss\\_Ref\\_Time](#) &ref\_time\_map)  
*Save ref time map to XML file.*
- bool [load\\_ref\\_location\\_xml](#) (const std::string &file\_name)  
*Read ref location from XML file.*
- bool [save\\_ref\\_location\\_xml](#) (const std::string &file\_name, [Agnss\\_Ref\\_Location](#) &ref\_location)  
*Save ref location map to XML file.*
- void **print\_assistance** ()

**Public Attributes**

- `std::string` [server\\_name](#)
- `int` [server\\_port](#)
- `int` [request](#)
- `std::map< int, Gps\_Ephemeris >` [gps\\_ephemeris\\_map](#)
- `std::map< int, Galileo\_Ephemeris >` [gal\\_ephemeris\\_map](#)
- `std::map< int, Gps\_CNAV\_Ephemeris >` [gps\\_cnav\\_ephemeris\\_map](#)
- `std::map< int, Glonass\_Gnav\_Ephemeris >` [glonass\\_gnav\\_ephemeris\\_map](#)
- `std::map< int, Gps\_Almanac >` [gps\\_almanac\\_map](#)
- `std::map< int, Galileo\_Almanac >` [gal\\_almanac\\_map](#)
- [Gps\\_Iono](#) [gps\\_iono](#)
- [Galileo\\_Iono](#) [gal\\_iono](#)
- [Agnss\\_Ref\\_Time](#) [gps\\_time](#)
- [Gps\\_Utc\\_Model](#) [gps\\_utc](#)
- [Galileo\\_Utc\\_Model](#) [gal\\_utc](#)
- [Gps\\_CNAV\\_Utc\\_Model](#) [gps\\_cnav\\_utc](#)
- [Glonass\\_Gnav\\_Utc\\_Model](#) [glo\\_gnav\\_utc](#)
- [Agnss\\_Ref\\_Location](#) [gps\\_ref\\_loc](#)
- `std::map< int, Gps\_Acq\_Assist >` [gps\\_acq\\_map](#)

**10.181.1 Detailed Description**

class that implements a C++ interface to external Secure User Location Protocol (SUPL) client library..  
 Definition at line 55 of file [gnss\\_sdr\\_supl\\_client.h](#).

**10.181.2 Member Function Documentation****10.181.2.1 load\_cnav\_ephemeris\_xml()**

```
bool Gnss_Sdr_Supl_Client::load_cnav_ephemeris_xml (
    const std::string & file_name)
```

Read GPS CNAV ephemeris map from XML file.

**10.181.2.2 load\_cnav\_utc\_xml()**

```
bool Gnss_Sdr_Supl_Client::load_cnav_utc_xml (
    const std::string & file_name)
```

Read CNAV GPS utc model from XML file.

**10.181.2.3 load\_ephemeris\_xml()**

```
bool Gnss_Sdr_Supl_Client::load_ephemeris_xml (
    const std::string & file_name)
```

Read GPS NAV ephemeris map from XML file.

**10.181.2.4 load\_gal\_almanac\_xml()**

```
bool Gnss_Sdr_Supl_Client::load_gal_almanac_xml (
    const std::string & file_name)
```

Read Galileo almanac map from XML file.

**10.181.2.5 load\_gal\_ephemeris\_xml()**

```
bool Gnss_Sdr_Supl_Client::load_gal_ephemeris_xml (
    const std::string & file_name)
```

Read Galileo ephemeris map from XML file.

**10.181.2.6 load\_gal\_iono\_xml()**

```
bool Gnss_Sdr_Supl_Client::load_gal_iono_xml (
    const std::string & file_name)
```

Read Galileo iono from XML file.

**10.181.2.7 load\_gal\_utc\_xml()**

```
bool Gnss_Sdr_Supl_Client::load_gal_utc_xml (
    const std::string & file_name)
```

Read Galileo utc model from XML file.

**10.181.2.8 load\_glo\_utc\_xml()**

```
bool Gnss_Sdr_Supl_Client::load_glo_utc_xml (
    const std::string & file_name)
```

Read Glonass utc model from XML file.

**10.181.2.9 load\_gnav\_ephemeris\_xml()**

```
bool Gnss_Sdr_Supl_Client::load_gnav_ephemeris_xml (
    const std::string & file_name)
```

Read GLONASS GNAV ephemeris map from XML file.

**10.181.2.10 load\_gps\_almanac\_xml()**

```
bool Gnss_Sdr_Supl_Client::load_gps_almanac_xml (
    const std::string & file_name)
```

Read GPS almanac map from XML file.

**10.181.2.11 load\_iono\_xml()**

```
bool Gnss_Sdr_Supl_Client::load_iono_xml (
    const std::string & file_name)
```

Read iono from XML file.

**10.181.2.12 load\_ref\_location\_xml()**

```
bool Gnss_Sdr_Supl_Client::load_ref_location_xml (
    const std::string & file_name)
```

Read ref location from XML file.

**10.181.2.13 load\_ref\_time\_xml()**

```
bool Gnss_Sdr_Supl_Client::load_ref_time_xml (
    const std::string & file_name)
```

Read ref time from XML file.

**10.181.2.14 load\_utc\_xml()**

```
bool Gnss_Sdr_Supl_Client::load_utc_xml (
    const std::string & file_name)
```

Read GPS utc model from XML file.

**10.181.2.15 save\_cnav\_ephemeris\_map\_xml()**

```
bool Gnss_Sdr_Supl_Client::save_cnav_ephemeris_map_xml (
    const std::string & file_name,
    std::map< int, Gps\_CNAV\_Ephemeris > eph_map)
```

Save GPS CNAV ephemeris map to XML file.

**10.181.2.16 save\_cnav\_utc\_xml()**

```
bool Gnss_Sdr_Supl_Client::save_cnav_utc_xml (
    const std::string & file_name,
    Gps_CNAV_Utc_Model & utc)
```

Save CNAV UTC model map to XML file.

**10.181.2.17 save\_ephemeris\_map\_xml()**

```
bool Gnss_Sdr_Supl_Client::save_ephemeris_map_xml (
    const std::string & file_name,
    std::map< int, Gps_Ephemeris > eph_map)
```

Save ephemeris map to XML file.

**10.181.2.18 save\_gal\_almanac\_xml()**

```
bool Gnss_Sdr_Supl_Client::save_gal_almanac_xml (
    const std::string & file_name,
    std::map< int, Galileo_Almanac > galileo_almanac_map_to_save)
```

Save Galileo almanac map to XML file.

**10.181.2.19 save\_gal\_ephemeris\_map\_xml()**

```
bool Gnss_Sdr_Supl_Client::save_gal_ephemeris_map_xml (
    const std::string & file_name,
    std::map< int, Galileo_Ephemeris > eph_map)
```

Save Galileo ephemeris map to XML file.

**10.181.2.20 save\_gal\_iono\_xml()**

```
bool Gnss_Sdr_Supl_Client::save_gal_iono_xml (
    const std::string & file_name,
    Galileo_Iono & iono)
```

Save Galileo iono map to XML file.

**10.181.2.21 save\_gal\_utc\_xml()**

```
bool Gnss_Sdr_Supl_Client::save_gal_utc_xml (
    const std::string & file_name,
    Galileo_Utc_Model & utc)
```

Save Galileo UTC model map to XML file.

**10.181.2.22 save\_glo\_utc\_xml()**

```
bool Gnss_Sdr_Supl_Client::save_glo_utc_xml (
    const std::string & file_name,
    Glonass_Gnav_Utc_Model & utc)
```

Save Glonass UTC model map to XML file.

**10.181.2.23 save\_gnav\_ephemeris\_map\_xml()**

```
bool Gnss_Sdr_Supl_Client::save_gnav_ephemeris_map_xml (
    const std::string & file_name,
    std::map< int, Glonass_Gnav_Ephemeris > eph_map)
```

Save GLONASS GNAV ephemeris map to XML file.

**10.181.2.24 save\_gps\_almanac\_xml()**

```
bool Gnss_Sdr_Supl_Client::save_gps_almanac_xml (
    const std::string & file_name,
    std::map< int, Gps_Almanac > gps_almanac_map_to_save)
```

Save GPS almanac map to XML file.

**10.181.2.25 save\_iono\_xml()**

```
bool Gnss_Sdr_Supl_Client::save_iono_xml (
    const std::string & file_name,
    Gps_Iono & iono)
```

Save iono map to XML file.

**10.181.2.26 save\_ref\_location\_xml()**

```
bool Gnss_Sdr_Supl_Client::save_ref_location_xml (
    const std::string & file_name,
    Agnss_Ref_Location & ref_location)
```

Save ref location map to XML file.

**10.181.2.27 save\_ref\_time\_xml()**

```
bool Gnss_Sdr_Supl_Client::save_ref_time_xml (
    const std::string & file_name,
    Agnss_Ref_Time & ref_time_map)
```

Save ref time map to XML file.

**10.181.2.28 save\_utc\_xml()**

```
bool Gnss_Sdr_Supl_Client::save_utc_xml (
    const std::string & file_name,
    Gps_Utc_Model & utc)
```

Save UTC model map to XML file.

**10.181.3 Member Data Documentation****10.181.3.1 gal\_almanac\_map**

std::map<int, Galileo\_Almanac> Gnss\_Sdr\_Supl\_Client::gal\_almanac\_map  
Definition at line 73 of file [gnss\\_sdr\\_supl\\_client.h](#).

**10.181.3.2 gal\_ephemeris\_map**

std::map<int, Galileo\_Ephemeris> Gnss\_Sdr\_Supl\_Client::gal\_ephemeris\_map  
Definition at line 67 of file [gnss\\_sdr\\_supl\\_client.h](#).

**10.181.3.3 gal\_iono**

Galileo\_Iono Gnss\_Sdr\_Supl\_Client::gal\_iono  
Definition at line 77 of file [gnss\\_sdr\\_supl\\_client.h](#).

**10.181.3.4 gal\_utc**

Galileo\_Utc\_Model Gnss\_Sdr\_Supl\_Client::gal\_utc  
Definition at line 82 of file [gnss\\_sdr\\_supl\\_client.h](#).

**10.181.3.5 glo\_gnav\_utc**

Glonass\_Gnav\_Utc\_Model Gnss\_Sdr\_Supl\_Client::glo\_gnav\_utc  
Definition at line 84 of file [gnss\\_sdr\\_supl\\_client.h](#).



### 10.181.3.6 glonass\_gnav\_ephemeris\_map

`std::map<int, Glonass\_Gnav\_Ephemeris> Gnss_Sdr_Supl_Client::glonass_gnav_ephemeris_map`  
Definition at line 69 of file [gnss\\_sdr\\_supl\\_client.h](#).

### 10.181.3.7 gps\_acq\_map

`std::map<int, Gps\_Acq\_Assist> Gnss_Sdr_Supl_Client::gps_acq_map`  
Definition at line 88 of file [gnss\\_sdr\\_supl\\_client.h](#).

### 10.181.3.8 gps\_almanac\_map

`std::map<int, Gps\_Almanac> Gnss_Sdr_Supl_Client::gps_almanac_map`  
Definition at line 72 of file [gnss\\_sdr\\_supl\\_client.h](#).

### 10.181.3.9 gps\_cnav\_ephemeris\_map

`std::map<int, Gps\_CNAV\_Ephemeris> Gnss_Sdr_Supl_Client::gps_cnav_ephemeris_map`  
Definition at line 68 of file [gnss\\_sdr\\_supl\\_client.h](#).

### 10.181.3.10 gps\_cnav\_utc

`Gps\_CNAV\_Utc\_Model Gnss_Sdr_Supl_Client::gps_cnav_utc`  
Definition at line 83 of file [gnss\\_sdr\\_supl\\_client.h](#).

### 10.181.3.11 gps\_ephemeris\_map

`std::map<int, Gps\_Ephemeris> Gnss_Sdr_Supl_Client::gps_ephemeris_map`  
Definition at line 66 of file [gnss\\_sdr\\_supl\\_client.h](#).

### 10.181.3.12 gps\_iono

`Gps\_Iono Gnss_Sdr_Supl_Client::gps_iono`  
Definition at line 76 of file [gnss\\_sdr\\_supl\\_client.h](#).

### 10.181.3.13 gps\_ref\_loc

`Gnss\_Ref\_Location Gnss_Sdr_Supl_Client::gps_ref_loc`  
Definition at line 86 of file [gnss\\_sdr\\_supl\\_client.h](#).

### 10.181.3.14 gps\_time

`Gnss\_Ref\_Time Gnss_Sdr_Supl_Client::gps_time`  
Definition at line 79 of file [gnss\\_sdr\\_supl\\_client.h](#).

### 10.181.3.15 gps\_utc

`Gps\_Utc\_Model Gnss_Sdr_Supl_Client::gps_utc`  
Definition at line 81 of file [gnss\\_sdr\\_supl\\_client.h](#).

### 10.181.3.16 request

`int Gnss_Sdr_Supl_Client::request`  
Definition at line 64 of file [gnss\\_sdr\\_supl\\_client.h](#).

### 10.181.3.17 server\_name

`std::string Gnss_Sdr_Supl_Client::server_name`  
Definition at line 62 of file [gnss\\_sdr\\_supl\\_client.h](#).

### 10.181.3.18 server\_port

```
int Gnss_Sdr_Supl_Client::server_port
```

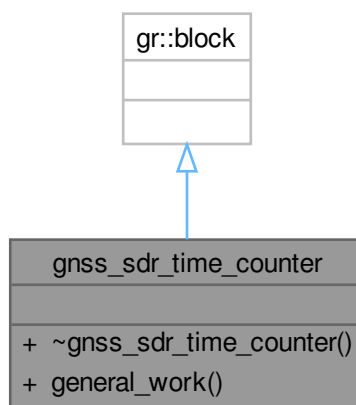
Definition at line 63 of file [gnss\\_sdr\\_supl\\_client.h](#).

The documentation for this class was generated from the following file:

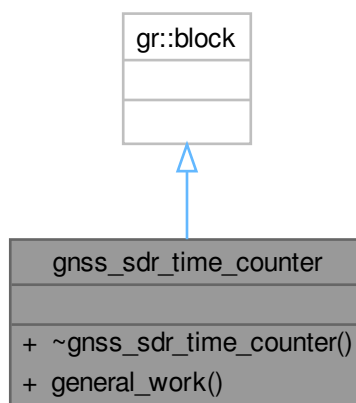
- [gnss\\_sdr\\_supl\\_client.h](#)

## 10.182 gnss\_sdr\_time\_counter Class Reference

Inheritance diagram for gnss\_sdr\_time\_counter:



Collaboration diagram for gnss\_sdr\_time\_counter:



**Public Member Functions**

- int **general\_work** (int noutput\_items \_\_attribute\_\_((unused)), gr\_vector\_int &ninput\_items \_\_attribute\_\_((unused)), gr\_vector\_const\_void\_star &input\_items \_\_attribute\_\_((unused)), gr\_vector\_void\_star &output\_items)

**Friends**

- gnss\_sdr\_time\_counter\_sptr **gnss\_sdr\_make\_time\_counter** ()

**10.182.1 Detailed Description**

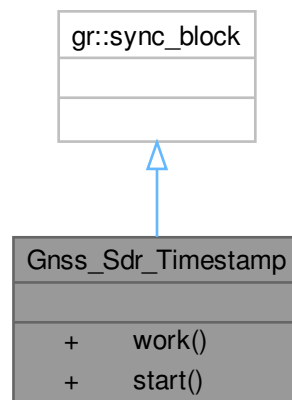
Definition at line 38 of file [gnss\\_sdr\\_time\\_counter.h](#).

The documentation for this class was generated from the following file:

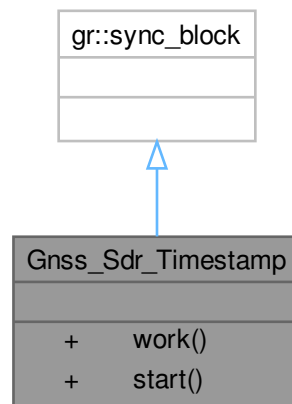
- [gnss\\_sdr\\_time\\_counter.h](#)

**10.183 Gnss\_Sdr\_Timestamp Class Reference**

Inheritance diagram for Gnss\_Sdr\_Timestamp:



Collaboration diagram for Gnss\_Sdr\_Timestamp:



### Public Member Functions

- int **work** (int noutput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)
- bool **start** ()

### Friends

- gnss\_shared\_ptr< [Gnss\\_Sdr\\_Timestamp](#) > **gnss\_sdr\_make\_Timestamp** (size\_t sizeof\_stream\_item, std::string timestamp\_file, double clock\_offset\_ms, int items\_to\_samples)

## 10.183.1 Detailed Description

Definition at line 46 of file [gnss\\_sdr\\_timestamp.h](#).

The documentation for this class was generated from the following file:

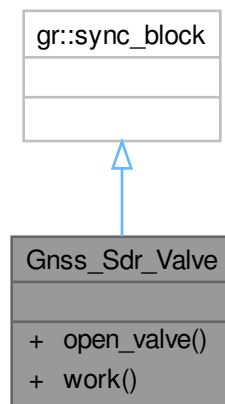
- [gnss\\_sdr\\_timestamp.h](#)

## 10.184 Gnss\_Sdr\_Valve Class Reference

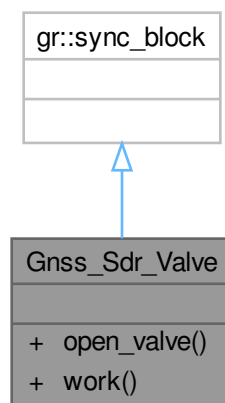
Implementation of a GNU Radio block that sends a STOP message to the control queue right after a specific number of samples have passed through it.

```
#include <gnss_sdr_valve.h>
```

Inheritance diagram for Gnss\_Sdr\_Valve:



Collaboration diagram for Gnss\_Sdr\_Valve:



### Public Member Functions

- void **open\_valve** ()
- int **work** (int noutput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)

### Friends

- gnss\_shared\_ptr< [Gnss\\_Sdr\\_Valve](#) > **gnss\_sdr\_make\_valve** (size\_t sizeof\_stream\_item, uint64\_t nitems, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)
- gnss\_shared\_ptr< [Gnss\\_Sdr\\_Valve](#) > **gnss\_sdr\_make\_valve** (size\_t sizeof\_stream\_item, uint64\_t nitems, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue, bool stop\_flowgraph)

### 10.184.1 Detailed Description

Implementation of a GNU Radio block that sends a STOP message to the control queue right after a specific number of samples have passed through it.

Definition at line 54 of file [gnss\\_sdr\\_valve.h](#).

The documentation for this class was generated from the following file:

- [gnss\\_sdr\\_valve.h](#)

## 10.185 Gnss\_Signal Class Reference

This class represents a GNSS signal.

`#include <gnss_signal.h>`

Collaboration diagram for Gnss\_Signal:

Gnss_Signal
<ul style="list-style-type: none"> <li>+ Gnss_Signal()</li> <li>+ Gnss_Signal()</li> <li>+ Gnss_Signal()</li> <li>+ ~Gnss_Signal()</li> <li>+ get_signal_str()</li> <li>+ get_satellite()</li> </ul>

### Public Member Functions

- **Gnss\_Signal** (const std::string &signal\_)
- **Gnss\_Signal** (const [Gnss\\_Satellite](#) &satellite\_, const std::string &signal\_)
- std::string **get\_signal\_str** () const  
*Get the satellite signal {"1C" for GPS L1 C/A, "2S" for GPS L2C (M), "L5" for GPS L5, "1G" for GLONASS L1 C/A, "1B" for Galileo E1B, "5X" for Galileo E5a.*
- [Gnss\\_Satellite](#) **get\_satellite** () const  
*Get the [Gnss\\_Satellite](#) associated to the signal.*

### Friends

- bool **operator==** (const [Gnss\\_Signal](#) &, const [Gnss\\_Signal](#) &)  
*operator== for comparison*
- std::ostream & **operator<<** (std::ostream &, const [Gnss\\_Signal](#) &)  
*operator<< for pretty printing*

### 10.185.1 Detailed Description

This class represents a GNSS signal.

It contains information about the space vehicle and the specific signal.

Definition at line 37 of file [gnss\\_signal.h](#).

## 10.185.2 Member Function Documentation

### 10.185.2.1 `get_satellite()`

`Gnss_Satellite` `Gnss_Signal::get_satellite () const`

Get the `Gnss_Satellite` associated to the signal.

### 10.185.2.2 `get_signal_str()`

`std::string` `Gnss_Signal::get_signal_str () const`

Get the satellite signal {"1C" for GPS L1 C/A, "2S" for GPS L2C (M), "L5" for GPS L5, "1G" for GLONASS L1 C/A, "1B" for Galileo E1B, "5X" for Galileo E5a.

## 10.185.3 Friends And Related Symbol Documentation

### 10.185.3.1 `operator<<`

```
std::ostream & operator<< (
    std::ostream & ,
    const Gnss_Signal & ) [friend]
```

`operator<<` for pretty printing

### 10.185.3.2 `operator==`

```
bool operator== (
    const Gnss_Signal & ,
    const Gnss_Signal & ) [friend]
```

`operator==` for comparison

The documentation for this class was generated from the following file:

- [gnss\\_signal.h](#)

## 10.186 Gnss\_Synchro Class Reference

This is the class that contains the information that is shared by the processing blocks.

```
#include <gnss_synchro.h>
```

Collaboration diagram for Gnss\_Synchro:

Gnss_Synchro
<ul style="list-style-type: none"> <li>+ System</li> <li>+ Signal</li> <li>+ PRN</li> <li>+ Channel_ID</li> <li>+ Acq_delay_samples</li> <li>+ Acq_doppler_hz</li> <li>+ Acq_samplestamp_samples</li> <li>+ Acq_doppler_step</li> <li>+ fs</li> <li>+ Prompt_I</li> <li>and 16 more...</li> </ul>
<ul style="list-style-type: none"> <li>+ Gnss_Synchro()</li> <li>+ ~Gnss_Synchro()</li> <li>+ Gnss_Synchro()</li> <li>+ operator=()</li> <li>+ Gnss_Synchro()</li> <li>+ operator=()</li> <li>+ serialize()</li> </ul>

### Public Member Functions

- [Gnss\\_Synchro](#) ()=default  
*Default constructor.*
- [~Gnss\\_Synchro](#) ()=default  
*Default destructor.*
- [Gnss\\_Synchro](#) (const [Gnss\\_Synchro](#) &other) noexcept=default  
*Copy constructor.*
- [Gnss\\_Synchro](#) & [operator=](#) (const [Gnss\\_Synchro](#) &rhs) noexcept  
*Copy assignment operator.*
- [Gnss\\_Synchro](#) ([Gnss\\_Synchro](#) &&other) noexcept=default  
*Move constructor.*
- [Gnss\\_Synchro](#) & [operator=](#) ([Gnss\\_Synchro](#) &&other) noexcept  
*Move assignment operator.*
- template<class Archive>  
void [serialize](#) (Archive &ar, const unsigned int version)  
*This member function serializes and restores [Gnss\\_Synchro](#) objects from a byte stream.*

### Public Attributes

- char [System](#) {}  
*Set by Channel::set\_signal(Gnss\_Signal gnss\_signal)*



- char [Signal](#) [3] {}  
*Set by Channel::set\_signal(Gnss\_Signal gnss\_signal)*
- uint32\_t [PRN](#) {}  
*Set by Channel::set\_signal(Gnss\_Signal gnss\_signal)*
- int32\_t [Channel\\_ID](#) {}  
*Set by Channel constructor.*
- double [Acq\\_delay\\_samples](#) {}  
*Set by Acquisition processing block.*
- double [Acq\\_doppler\\_hz](#) {}  
*Set by Acquisition processing block.*
- uint64\_t [Acq\\_samplestamp\\_samples](#) {}  
*Set by Acquisition processing block.*
- uint32\_t [Acq\\_doppler\\_step](#) {}  
*Set by Acquisition processing block.*
- int64\_t [fs](#) {}  
*Set by Tracking processing block.*
- double [Prompt\\_I](#) {}  
*Set by Tracking processing block.*
- double [Prompt\\_Q](#) {}  
*Set by Tracking processing block.*
- double [CN0\\_dB\\_hz](#) {}  
*Set by Tracking processing block.*
- double [Carrier\\_Doppler\\_hz](#) {}  
*Set by Tracking processing block.*
- double [Carrier\\_phase\\_rads](#) {}  
*Set by Tracking processing block.*
- double [Code\\_phase\\_samples](#) {}  
*Set by Tracking processing block.*
- uint64\_t [Tracking\\_sample\\_counter](#) {}  
*Set by Tracking processing block.*
- int32\_t [correlation\\_length\\_ms](#) {}  
*Set by Tracking processing block.*
- uint32\_t [TOW\\_at\\_current\\_symbol\\_ms](#) {}  
*Set by Telemetry Decoder processing block.*
- double [Pseudorange\\_m](#) {}  
*Set by Observables processing block.*
- double [RX\\_time](#) {}  
*Set by Observables processing block.*
- double [interp\\_TOW\\_ms](#) {}  
*Set by Observables processing block.*
- bool [Flag\\_valid\\_acquisition](#) {}  
*Set by Acquisition processing block.*
- bool [Flag\\_valid\\_symbol\\_output](#) {}  
*Set by Tracking processing block.*
- bool [Flag\\_valid\\_word](#) {}  
*Set by Telemetry Decoder processing block.*
- bool [Flag\\_valid\\_pseudorange](#) {}  
*Set by Observables processing block.*
- bool [Flag\\_PLL\\_180\\_deg\\_phase\\_locked](#) {}  
*Set by Telemetry Decoder processing block.*

### 10.186.1 Detailed Description

This is the class that contains the information that is shared by the processing blocks.  
Definition at line 38 of file [gnss\\_synchro.h](#).

### 10.186.2 Constructor & Destructor Documentation

#### 10.186.2.1 Gnss\_Synchro() [1/3]

```
Gnss_Synchro::Gnss_Synchro () [default]
```

Default constructor.

Referenced by [Gnss\\_Synchro\(\)](#), [Gnss\\_Synchro\(\)](#), [operator=\(\)](#), and [operator=\(\)](#).

#### 10.186.2.2 ~Gnss\_Synchro()

```
Gnss_Synchro::~Gnss_Synchro () [default]
```

Default destructor.

#### 10.186.2.3 Gnss\_Synchro() [2/3]

```
Gnss_Synchro::Gnss_Synchro (
    const Gnss_Synchro & other) [default], [noexcept]
```

Copy constructor.

References [Gnss\\_Synchro\(\)](#).

Here is the call graph for this function:



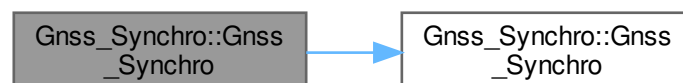
#### 10.186.2.4 Gnss\_Synchro() [3/3]

```
Gnss_Synchro::Gnss_Synchro (
    Gnss_Synchro && other) [default], [noexcept]
```

Move constructor.

References [Gnss\\_Synchro\(\)](#).

Here is the call graph for this function:



### 10.186.3 Member Function Documentation

#### 10.186.3.1 operator=() [1/2]

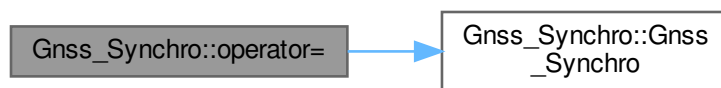
```
Gnss_Synchro & Gnss_Synchro::operator= (
    const Gnss_Synchro & rhs) [inline], [noexcept]
```

Copy assignment operator.

Definition at line 87 of file [gnss\\_synchro.h](#).

References [Acq\\_delay\\_samples](#), [Acq\\_doppler\\_hz](#), [Acq\\_doppler\\_step](#), [Acq\\_samplestamp\\_samples](#), [Carrier\\_Doppler\\_hz](#), [Carrier\\_phase\\_rads](#), [Channel\\_ID](#), [CN0\\_dB\\_hz](#), [Code\\_phase\\_samples](#), [correlation\\_length\\_ms](#), [Flag\\_PLL\\_180\\_deg\\_phase\\_locked](#), [Flag\\_valid\\_acquisition](#), [Flag\\_valid\\_pseudorange](#), [Flag\\_valid\\_symbol\\_output](#), [Flag\\_valid\\_word](#), [fs](#), [Gnss\\_Synchro\(\)](#), [interp\\_TOW\\_ms](#), [PRN](#), [Prompt\\_I](#), [Prompt\\_Q](#), [Pseudorange\\_m](#), [RX\\_time](#), [Signal](#), [System](#), [TOW\\_at\\_current\\_symbol\\_ms](#), and [Tracking\\_sample\\_counter](#).

Here is the call graph for this function:



#### 10.186.3.2 operator=() [2/2]

```
Gnss_Synchro & Gnss_Synchro::operator= (
    Gnss_Synchro && other) [inline], [noexcept]
```

Move assignment operator.

Definition at line 128 of file [gnss\\_synchro.h](#).

References [Acq\\_delay\\_samples](#), [Acq\\_doppler\\_hz](#), [Acq\\_doppler\\_step](#), [Acq\\_samplestamp\\_samples](#), [Carrier\\_Doppler\\_hz](#), [Carrier\\_phase\\_rads](#), [Channel\\_ID](#), [CN0\\_dB\\_hz](#), [Code\\_phase\\_samples](#), [correlation\\_length\\_ms](#), [Flag\\_PLL\\_180\\_deg\\_phase\\_locked](#), [Flag\\_valid\\_acquisition](#), [Flag\\_valid\\_pseudorange](#), [Flag\\_valid\\_symbol\\_output](#), [Flag\\_valid\\_word](#), [fs](#), [Gnss\\_Synchro\(\)](#), [interp\\_TOW\\_ms](#), [PRN](#), [Prompt\\_I](#), [Prompt\\_Q](#), [Pseudorange\\_m](#), [RX\\_time](#), [Signal](#), [System](#), [TOW\\_at\\_current\\_symbol\\_ms](#), and [Tracking\\_sample\\_counter](#).

Here is the call graph for this function:



#### 10.186.3.3 serialize()

```
template<class Archive>
void Gnss_Synchro::serialize (
    Archive & ar,
    const unsigned int version) [inline]
```

This member function serializes and restores [Gnss\\_Synchro](#) objects from a byte stream.

Definition at line 200 of file [gnss\\_synchro.h](#).

References [Acq\\_delay\\_samples](#), [Acq\\_doppler\\_hz](#), [Acq\\_doppler\\_step](#), [Acq\\_samplestamp\\_samples](#), [Carrier\\_Doppler\\_hz](#), [Carrier\\_phase\\_rads](#), [Channel\\_ID](#), [CN0\\_dB\\_hz](#), [Code\\_phase\\_samples](#), [correlation\\_length\\_ms](#), [Flag\\_PLL\\_180\\_deg\\_phase\\_locked](#), [Flag\\_valid\\_acquisition](#), [Flag\\_valid\\_pseudorange](#), [Flag\\_valid\\_symbol\\_output](#), [Flag\\_valid\\_word](#), [fs](#), [interp\\_TOW\\_ms](#), [PRN](#), [Prompt\\_I](#), [Prompt\\_Q](#), [Pseudorange\\_m](#), [RX\\_time](#), [Signal](#), [System](#), [TOW\\_at\\_current\\_symbol\\_ms](#), and [Tracking\\_sample\\_counter](#).

## 10.186.4 Member Data Documentation

### 10.186.4.1 Acq\_delay\_samples

```
double Gnss_Synchro::Acq_delay_samples {}
```

Set by Acquisition processing block.

Definition at line 52 of file [gnss\\_synchro.h](#).

Referenced by [operator=\(\)](#), [operator=\(\)](#), [Serdes\\_Gnss\\_Synchro::readProtobuffer\(\)](#), and [serialize\(\)](#).

### 10.186.4.2 Acq\_doppler\_hz

```
double Gnss_Synchro::Acq_doppler_hz {}
```

Set by Acquisition processing block.

Definition at line 53 of file [gnss\\_synchro.h](#).

Referenced by [operator=\(\)](#), [operator=\(\)](#), [Serdes\\_Gnss\\_Synchro::readProtobuffer\(\)](#), and [serialize\(\)](#).

### 10.186.4.3 Acq\_doppler\_step

```
uint32_t Gnss_Synchro::Acq_doppler_step {}
```

Set by Acquisition processing block.

Definition at line 55 of file [gnss\\_synchro.h](#).

Referenced by [operator=\(\)](#), [operator=\(\)](#), [Serdes\\_Gnss\\_Synchro::readProtobuffer\(\)](#), and [serialize\(\)](#).

### 10.186.4.4 Acq\_samplestamp\_samples

```
uint64_t Gnss_Synchro::Acq_samplestamp_samples {}
```

Set by Acquisition processing block.

Definition at line 54 of file [gnss\\_synchro.h](#).

Referenced by [operator=\(\)](#), [operator=\(\)](#), [Serdes\\_Gnss\\_Synchro::readProtobuffer\(\)](#), and [serialize\(\)](#).

### 10.186.4.5 Carrier\_Doppler\_hz

```
double Gnss_Synchro::Carrier_Doppler_hz {}
```

Set by Tracking processing block.

Definition at line 62 of file [gnss\\_synchro.h](#).

Referenced by [operator=\(\)](#), [operator=\(\)](#), [Serdes\\_Gnss\\_Synchro::readProtobuffer\(\)](#), and [serialize\(\)](#).

### 10.186.4.6 Carrier\_phase\_rads

```
double Gnss_Synchro::Carrier_phase_rads {}
```

Set by Tracking processing block.

Definition at line 63 of file [gnss\\_synchro.h](#).

Referenced by [operator=\(\)](#), [operator=\(\)](#), [Serdes\\_Gnss\\_Synchro::readProtobuffer\(\)](#), and [serialize\(\)](#).

### 10.186.4.7 Channel\_ID

```
int32_t Gnss_Synchro::Channel_ID {}
```

Set by [Channel](#) constructor.

Definition at line 49 of file [gnss\\_synchro.h](#).

Referenced by [operator=\(\)](#), [operator=\(\)](#), [Serdes\\_Gnss\\_Synchro::readProtobuffer\(\)](#), and [serialize\(\)](#).

#### 10.186.4.8 CN0\_dB\_hz

```
double Gnss_Synchro::CN0_dB_hz {}
```

Set by Tracking processing block.

Definition at line 61 of file [gnss\\_synchro.h](#).

Referenced by [operator=\(\)](#), [operator=\(\)](#), [Serdes\\_Gnss\\_Synchro::readProtobuffer\(\)](#), and [serialize\(\)](#).

#### 10.186.4.9 Code\_phase\_samples

```
double Gnss_Synchro::Code_phase_samples {}
```

Set by Tracking processing block.

Definition at line 64 of file [gnss\\_synchro.h](#).

Referenced by [operator=\(\)](#), [operator=\(\)](#), [Serdes\\_Gnss\\_Synchro::readProtobuffer\(\)](#), and [serialize\(\)](#).

#### 10.186.4.10 correlation\_length\_ms

```
int32_t Gnss_Synchro::correlation_length_ms {}
```

Set by Tracking processing block.

Definition at line 66 of file [gnss\\_synchro.h](#).

Referenced by [operator=\(\)](#), [operator=\(\)](#), [Serdes\\_Gnss\\_Synchro::readProtobuffer\(\)](#), and [serialize\(\)](#).

#### 10.186.4.11 Flag\_PLL\_180\_deg\_phase\_locked

```
bool Gnss_Synchro::Flag_PLL_180_deg_phase_locked {}
```

Set by Telemetry Decoder processing block.

Definition at line 81 of file [gnss\\_synchro.h](#).

Referenced by [operator=\(\)](#), [operator=\(\)](#), [Serdes\\_Gnss\\_Synchro::readProtobuffer\(\)](#), and [serialize\(\)](#).

#### 10.186.4.12 Flag\_valid\_acquisition

```
bool Gnss_Synchro::Flag_valid_acquisition {}
```

Set by Acquisition processing block.

Definition at line 77 of file [gnss\\_synchro.h](#).

Referenced by [operator=\(\)](#), [operator=\(\)](#), [Serdes\\_Gnss\\_Synchro::readProtobuffer\(\)](#), and [serialize\(\)](#).

#### 10.186.4.13 Flag\_valid\_pseudorange

```
bool Gnss_Synchro::Flag_valid_pseudorange {}
```

Set by Observables processing block.

Definition at line 80 of file [gnss\\_synchro.h](#).

Referenced by [operator=\(\)](#), [operator=\(\)](#), [Serdes\\_Gnss\\_Synchro::readProtobuffer\(\)](#), and [serialize\(\)](#).

#### 10.186.4.14 Flag\_valid\_symbol\_output

```
bool Gnss_Synchro::Flag_valid_symbol_output {}
```

Set by Tracking processing block.

Definition at line 78 of file [gnss\\_synchro.h](#).

Referenced by [operator=\(\)](#), [operator=\(\)](#), [Serdes\\_Gnss\\_Synchro::readProtobuffer\(\)](#), and [serialize\(\)](#).

#### 10.186.4.15 Flag\_valid\_word

```
bool Gnss_Synchro::Flag_valid_word {}
```

Set by Telemetry Decoder processing block.

Definition at line 79 of file [gnss\\_synchro.h](#).

Referenced by [operator=\(\)](#), [operator=\(\)](#), [Serdes\\_Gnss\\_Synchro::readProtobuffer\(\)](#), and [serialize\(\)](#).

#### 10.186.4.16 fs

```
int64_t Gnss_Synchro::fs {}
```

Set by Tracking processing block.

Definition at line 58 of file [gnss\\_synchro.h](#).

Referenced by [operator=\(\)](#), [operator=\(\)](#), [Serdes\\_Gnss\\_Synchro::readProtobuffer\(\)](#), and [serialize\(\)](#).

#### 10.186.4.17 interp\_TOW\_ms

```
double Gnss_Synchro::interp_TOW_ms {}
```

Set by Observables processing block.

Definition at line 74 of file [gnss\\_synchro.h](#).

Referenced by [operator=\(\)](#), [operator=\(\)](#), [Serdes\\_Gnss\\_Synchro::readProtobuffer\(\)](#), and [serialize\(\)](#).

#### 10.186.4.18 PRN

```
uint32_t Gnss_Synchro::PRN {}
```

Set by Channel::set\_signal(Gnss\_Signal gnss\_signal)

Definition at line 48 of file [gnss\\_synchro.h](#).

Referenced by [operator=\(\)](#), [operator=\(\)](#), [Serdes\\_Gnss\\_Synchro::readProtobuffer\(\)](#), and [serialize\(\)](#).

#### 10.186.4.19 Prompt\_I

```
double Gnss_Synchro::Prompt_I {}
```

Set by Tracking processing block.

Definition at line 59 of file [gnss\\_synchro.h](#).

Referenced by [operator=\(\)](#), [operator=\(\)](#), [Serdes\\_Gnss\\_Synchro::readProtobuffer\(\)](#), and [serialize\(\)](#).

#### 10.186.4.20 Prompt\_Q

```
double Gnss_Synchro::Prompt_Q {}
```

Set by Tracking processing block.

Definition at line 60 of file [gnss\\_synchro.h](#).

Referenced by [operator=\(\)](#), [operator=\(\)](#), [Serdes\\_Gnss\\_Synchro::readProtobuffer\(\)](#), and [serialize\(\)](#).

#### 10.186.4.21 Pseudorange\_m

```
double Gnss_Synchro::Pseudorange_m {}
```

Set by Observables processing block.

Definition at line 72 of file [gnss\\_synchro.h](#).

Referenced by [operator=\(\)](#), [operator=\(\)](#), [Serdes\\_Gnss\\_Synchro::readProtobuffer\(\)](#), and [serialize\(\)](#).

#### 10.186.4.22 RX\_time

```
double Gnss_Synchro::RX_time {}
```

Set by Observables processing block.

Definition at line 73 of file [gnss\\_synchro.h](#).

Referenced by [operator=\(\)](#), [operator=\(\)](#), [Serdes\\_Gnss\\_Synchro::readProtobuffer\(\)](#), and [serialize\(\)](#).

#### 10.186.4.23 Signal

```
char Gnss_Synchro::Signal[3] {}
```

Set by Channel::set\_signal(Gnss\_Signal gnss\_signal)

Definition at line 47 of file [gnss\\_synchro.h](#).

Referenced by [operator=\(\)](#), [operator=\(\)](#), [Serdes\\_Gnss\\_Synchro::readProtobuffer\(\)](#), and [serialize\(\)](#).

#### 10.186.4.24 System

```
char Gnss_Synchro::System {}
```

Set by Channel::set\_signal(Gnss\_Signal gnss\_signal)

Definition at line 46 of file [gnss\\_synchro.h](#).

Referenced by [operator=\(\)](#), [operator=\(\)](#), [Serdes\\_Gnss\\_Synchro::readProtobuffer\(\)](#), and [serialize\(\)](#).

**10.186.4.25 TOW\_at\_current\_symbol\_ms**

```
uint32_t Gnss_Synchro::TOW_at_current_symbol_ms {}
```

Set by Telemetry Decoder processing block.

Definition at line 69 of file [gnss\\_synchro.h](#).

Referenced by [operator=\(\)](#), [operator=\(\)](#), [Serdes\\_Gnss\\_Synchro::readProtobuffer\(\)](#), and [serialize\(\)](#).

**10.186.4.26 Tracking\_sample\_counter**

```
uint64_t Gnss_Synchro::Tracking_sample_counter {}
```

Set by Tracking processing block.

Definition at line 65 of file [gnss\\_synchro.h](#).

Referenced by [operator=\(\)](#), [operator=\(\)](#), [Serdes\\_Gnss\\_Synchro::readProtobuffer\(\)](#), and [serialize\(\)](#).

The documentation for this class was generated from the following file:

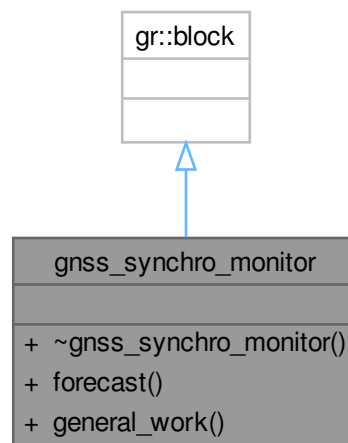
- [gnss\\_synchro.h](#)

**10.187 gnss\_synchro\_monitor Class Reference**

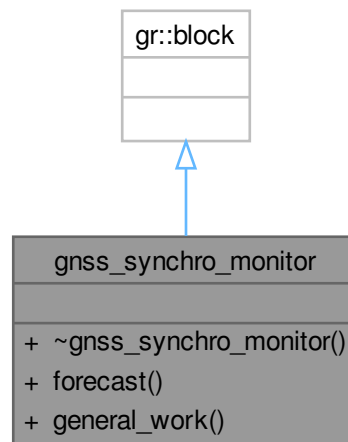
This class implements a monitoring block which allows sending a data stream with the receiver internal parameters ([Gnss\\_Synchro](#) objects) to local or remote clients over UDP.

```
#include <gnss_synchro_monitor.h>
```

Inheritance diagram for `gnss_synchro_monitor`:



Collaboration diagram for `gnss_synchro_monitor`:



### Public Member Functions

- `~gnss_synchro_monitor()` = default  
*Default destructor.*
- void **forecast** (int noutput\_items, gr\_vector\_int &ninput\_items\_required)
- int **general\_work** (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)

### Friends

- `gnss_synchro_monitor_sptr gnss_synchro_make_monitor` (int n\_channels, int decimation\_factor, const std::vector< std::string > &udp\_ports, const std::vector< std::string > &udp\_addresses, bool enable\_protobuf)

## 10.187.1 Detailed Description

This class implements a monitoring block which allows sending a data stream with the receiver internal parameters (`Gnss_Synchro` objects) to local or remote clients over UDP.

Definition at line 53 of file `gnss_synchro_monitor.h`.

## 10.187.2 Constructor & Destructor Documentation

### 10.187.2.1 ~gnss\_synchro\_monitor()

```
gnss_synchro_monitor::~gnss_synchro_monitor () [default]
```

Default destructor.

The documentation for this class was generated from the following file:

- `gnss_synchro_monitor.h`

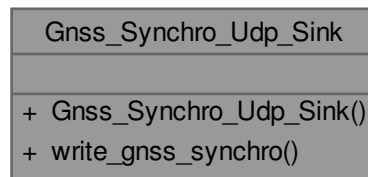
## 10.188 Gnss\_Synchro\_Udp\_Sink Class Reference

This class sends serialized `Gnss_Synchro` objects over UDP to one or multiple endpoints.

```
#include <gnss_synchro_udp_sink.h>
```



Collaboration diagram for Gnss\_Synchro\_Udp\_Sink:



### Public Member Functions

- **Gnss\_Synchro\_Udp\_Sink** (const std::vector< std::string > &addresses, const std::vector< std::string > &ports, bool enable\_protobuf)
- bool **write\_gnss\_synchro** (const std::vector< [Gnss\\_Synchro](#) > &stocks)

### 10.188.1 Detailed Description

This class sends serialized [Gnss\\_Synchro](#) objects over UDP to one or multiple endpoints.

Definition at line 45 of file [gnss\\_synchro\\_udp\\_sink.h](#).

The documentation for this class was generated from the following file:

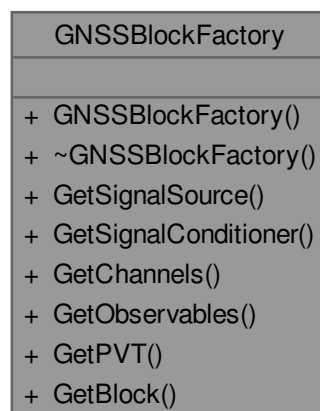
- [gnss\\_synchro\\_udp\\_sink.h](#)

## 10.189 GNSSBlockFactory Class Reference

Class that produces all kinds of GNSS blocks.

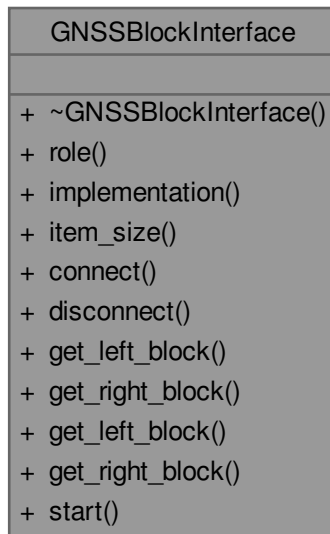
#include <gnss\_block\_factory.h>

Collaboration diagram for GNSSBlockFactory:





Collaboration diagram for GNSSBlockInterface:



### Public Member Functions

- virtual std::string [role](#) ()=0
- virtual std::string [implementation](#) ()=0
- virtual size\_t [item\\_size](#) ()=0
- virtual void [connect](#) (gr::top\_block\_sptr top\_block)=0
- virtual void [disconnect](#) (gr::top\_block\_sptr top\_block)=0
- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) ()=0
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) ()=0
- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.190.1 Detailed Description

This abstract class represents an interface to GNSS blocks.

Abstract class for GNSS block interfaces. Since all its methods are virtual, this class cannot be instantiated directly, and a subclass can only be instantiated directly if all inherited pure virtual methods have been implemented by that class or a parent class.

Definition at line 68 of file [gnss\\_block\\_interface.h](#).

### 10.190.2 Member Function Documentation

#### 10.190.2.1 connect()

```
virtual void GNSSBlockInterface::connect (
    gr::top_block_sptr top_block) [pure virtual]
```

Implemented in [Channel](#), [FifoSignalSource](#), [GalileoE1DIIPIIVemlTrackingFpga](#), [GalileoE1PcpsAmbiguousAcquisitionFpga](#), [GalileoE5aDIIPIITrackingFpga](#), [GalileoE5aPcpsAcquisitionFpga](#), [GalileoE5bDIIPIITracking](#), [GalileoE5bPcpsAcquisition](#), [GalileoE5bPcpsAcquisitionFpga](#), [GalileoE5bTelemetryDecoder](#), [GalileoE6DIIPIITracking](#), [GalileoE6TelemetryDecoder](#), [GpsL1CaDIIPIITrackingFpga](#), [GpsL1CaPcpsAcquisitionFpga](#), [GpsL5DIIPIITrackingFpga](#), and [GpsL5IPcpsAcquisitionFpga](#).

**10.190.2.2 disconnect()**

```
virtual void GNSSBlockInterface::disconnect (
    gr::top_block_sptr top_block) [pure virtual]
```

Implemented in [GalileoE1DIIPIIVemlTrackingFpga](#), [GalileoE1PcpsAmbiguousAcquisitionFpga](#), [GalileoE5aDIIPIITrackingFpga](#), [GalileoE5aPcpsAcquisitionFpga](#), [GalileoE5bDIIPIITracking](#), [GalileoE5bPcpsAcquisition](#), [GalileoE5bPcpsAcquisitionFpga](#), [GalileoE5bTelemetryDecoder](#), [GalileoE6DIIPIITracking](#), [GalileoE6TelemetryDecoder](#), [GpsL1CaDIIPIITrackingFpga](#), [GpsL1CaPcpsAcquisitionFpga](#), [GpsL5DIIPIITrackingFpga](#), and [GpsL5iPcpsAcquisitionFpga](#).

**10.190.2.3 get\_left\_block() [1/2]**

```
virtual gr::basic_block_sptr GNSSBlockInterface::get_left_block () [pure virtual]
```

Implemented in [GalileoE1DIIPIIVemlTrackingFpga](#), [GalileoE1PcpsAmbiguousAcquisitionFpga](#), [GalileoE5aDIIPIITrackingFpga](#), [GalileoE5aPcpsAcquisitionFpga](#), [GalileoE5bDIIPIITracking](#), [GalileoE5bPcpsAcquisition](#), [GalileoE5bPcpsAcquisitionFpga](#), [GalileoE5bTelemetryDecoder](#), [GalileoE6DIIPIITracking](#), [GalileoE6TelemetryDecoder](#), [GpsL1CaDIIPIITrackingFpga](#), [GpsL1CaPcpsAcquisitionFpga](#), [GpsL5DIIPIITrackingFpga](#), and [GpsL5iPcpsAcquisitionFpga](#).

**10.190.2.4 get\_left\_block() [2/2]**

```
virtual gr::basic_block_sptr GNSSBlockInterface::get_left_block (
    int RF_channel) [inline], [virtual]
```

Definition at line 81 of file [gnss\\_block\\_interface.h](#).

**10.190.2.5 get\_right\_block() [1/2]**

```
virtual gr::basic_block_sptr GNSSBlockInterface::get_right_block () [pure virtual]
```

Implemented in [Channel](#), [GalileoE1DIIPIIVemlTrackingFpga](#), [GalileoE1PcpsAmbiguousAcquisitionFpga](#), [GalileoE5aDIIPIITrackingFpga](#), [GalileoE5aPcpsAcquisitionFpga](#), [GalileoE5bDIIPIITracking](#), [GalileoE5bPcpsAcquisition](#), [GalileoE5bPcpsAcquisitionFpga](#), [GalileoE5bTelemetryDecoder](#), [GalileoE6DIIPIITracking](#), [GalileoE6TelemetryDecoder](#), [GpsL1CaDIIPIITrackingFpga](#), [GpsL1CaPcpsAcquisitionFpga](#), [GpsL5DIIPIITrackingFpga](#), and [GpsL5iPcpsAcquisitionFpga](#).

**10.190.2.6 get\_right\_block() [2/2]**

```
virtual gr::basic_block_sptr GNSSBlockInterface::get_right_block (
    int RF_channel) [inline], [virtual]
```

Definition at line 89 of file [gnss\\_block\\_interface.h](#).

**10.190.2.7 implementation()**

```
virtual std::string GNSSBlockInterface::implementation () [pure virtual]
```

Implemented in [ArraySignalConditioner](#), [BeamformerFilter](#), [BeidouB1iPcpsAcquisition](#), [BeidouB1iTelemetryDecoder](#), [BeidouB3iPcpsAcquisition](#), [BeidouB3iTelemetryDecoder](#), [ByteToShort](#), [Channel](#), [CshortToGrComplex](#), [DirectResamplerConditioner](#), [FirFilter](#), [FreqXlatingFirFilter](#), [GalileoE1BTelemetryDecoder](#), [GalileoE1DIIPIIVemlTracking](#), [GalileoE1DIIPIIVemlTrackingFpga](#), [GalileoE1Pcps8msAmbiguousAcquisition](#), [GalileoE1PcpsAmbiguousAcquisition](#), [GalileoE1PcpsAmbiguousAcquisitionFpga](#), [GalileoE1PcpsCccwsrAmbiguousAcquisition](#), [GalileoE1PcpsQuickSyncAmbiguousAcquisition](#), [GalileoE1PcpsTongAmbiguousAcquisition](#), [GalileoE1TcpConnectorTracking](#), [GalileoE5aDIIPIITracking](#), [GalileoE5aDIIPIITrackingFpga](#), [GalileoE5aNoncoherentIQAcquisitionCaf](#), [GalileoE5aPcpsAcquisitionFpga](#), [GalileoE5aTelemetryDecoder](#), [GalileoE5bDIIPIITracking](#), [GalileoE5bPcpsAcquisition](#), [GalileoE5bPcpsAcquisitionFpga](#), [GalileoE5bTelemetryDecoder](#), [GalileoE6DIIPIITracking](#), [GalileoE6PcpsAcquisition](#), [GalileoE6TelemetryDecoder](#), [GenSignalSource](#), [GlonassL1CaDIIPIICaidTracking](#), [GlonassL1CaDIIPIITracking](#), [GlonassL1CaPcpsAcquisition](#), [GlonassL1CaTelemetryDecoder](#), [GlonassL2CaDIIPIICaidTracking](#), [GlonassL2CaDIIPIITracking](#), [GlonassL2CaPcpsAcquisition](#), [GlonassL2CaTelemetryDecoder](#), [GpsL1CaDIIPIITracking](#), [GpsL1CaDIIPIITrackingFpga](#), [GpsL1CaDIIPIITrackingGPU](#), [GpsL1CaGaussianTracking](#), [GpsL1CaKfTracking](#), [GpsL1CaPcpsAcquisition](#), [GpsL1CaPcpsAcquisitionFineDoppler](#), [GpsL1CaPcpsAcquisitionFpga](#), [GpsL1CaPcpsAssistedAcquisition](#), [GpsL1CaPcpsOpenCIAcquisition](#), [GpsL1CaPcpsQuickSyncAcquisition](#), [GpsL1CaPcpsTongAcquisition](#), [GpsL1CaTcpConnectorTracking](#), [GpsL1CaTelemetryDecoder](#), [GpsL2CTelemetryDecoder](#), [GpsL2MDIIPIITracking](#), [GpsL2MDIIPIITrackingFpga](#), [GpsL2MPcpsAcquisition](#), [GpsL2MPcpsAcquisitionFpga](#), [GpsL5DIIPIITracking](#), [GpsL5DIIPIITrackingFpga](#), [GpsL5iPcpsAcquisition](#), [GpsL5iPcpsAcquisitionFpga](#), [GpsL5TelemetryDecoder](#), [HybridObservables](#), [lbyteToCbyte](#), [lbyteToComplex](#), [lbyteToCshort](#), [lshortToComplex](#), [lshortToCshort](#), [NotchFilter](#), [NotchFilterLite](#), [Pass\\_Through](#), [PulseBlankingFilter](#), [Rtklib\\_Pvt](#), [SbasL1TelemetryDecoder](#), [SignalConditioner](#), and [SignalGenerator](#).

### 10.190.2.8 item\_size()

virtual size\_t GNSSBlockInterface::item\_size () [pure virtual]

Implemented in [FileSourceBase](#), [GalileoE1DIIPIIVemlTrackingFpga](#), [GalileoE1PcpsAmbiguousAcquisitionFpga](#), [GalileoE5aDIIPIITrackingFpga](#), [GalileoE5aPcpsAcquisitionFpga](#), [GalileoE5bPcpsAcquisition](#), [GalileoE5bPcpsAcquisitionFpga](#), [GpsL1CaDIIPIITrackingFpga](#), [GpsL1CaPcpsAcquisitionFpga](#), [GpsL5DIIPIITrackingFpga](#), [GpsL5iPcpsAcquisitionFpga](#), [HybridObservables](#), and [Rtklib\\_Pvt](#).

### 10.190.2.9 role()

virtual std::string GNSSBlockInterface::role () [pure virtual]

Implemented in [GalileoE1DIIPIIVemlTrackingFpga](#), [GalileoE1PcpsAmbiguousAcquisitionFpga](#), [GalileoE5aDIIPIITrackingFpga](#), [GalileoE5aPcpsAcquisitionFpga](#), [GalileoE5bPcpsAcquisition](#), [GalileoE5bPcpsAcquisitionFpga](#), [GpsL1CaDIIPIITrackingFpga](#), [GpsL1CaPcpsAcquisitionFpga](#), [GpsL5DIIPIITrackingFpga](#), and [GpsL5iPcpsAcquisitionFpga](#).

### 10.190.2.10 start()

virtual void GNSSBlockInterface::start () [inline], [virtual]

Start the flow of samples if needed.

Reimplemented in [DMASignalSourceFPGA](#).

Definition at line 101 of file [gnss\\_block\\_interface.h](#).

The documentation for this class was generated from the following file:

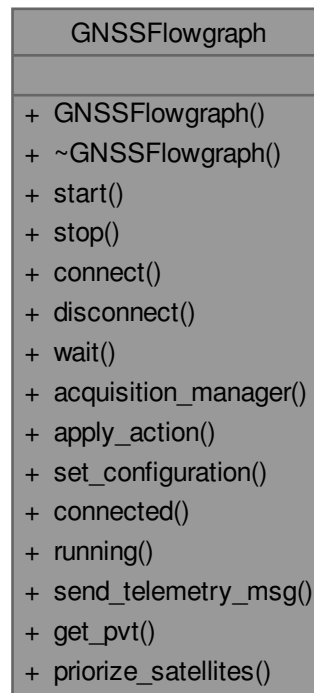
- [gnss\\_block\\_interface.h](#)

## 10.191 GNSSFlowgraph Class Reference

This class represents a GNSS flow graph.

```
#include <gnss_flowgraph.h>
```

Collaboration diagram for GNSSFlowgraph:



### Public Member Functions

- [GNSSFlowgraph](#) (std::shared\_ptr< [ConfigurationInterface](#) > configuration, std::shared\_ptr< [Concurrent\\_Queue](#)< pmt::pmt\_t > > queue)  
*Constructor that initializes the receiver flow graph.*
- [~GNSSFlowgraph](#) ()  
*Destructor.*
- void [start](#) ()  
*Start the flow graph.*
- void [stop](#) ()  
*Stop the flow graph.*
- void [connect](#) ()  
*Connects the defined blocks in the flow graph.*
- void [disconnect](#) ()  
*Disconnect the blocks in the flow graph.*
- void [wait](#) ()  
*Wait for a flowgraph to complete.*
- void [acquisition\\_manager](#) (unsigned int who)  
*Manage satellite acquisition.*
- void [apply\\_action](#) (unsigned int who, unsigned int what)  
*Applies an action to the flow graph.*
- void [set\\_configuration](#) (const std::shared\_ptr< [ConfigurationInterface](#) > &configuration)  
*Set flow graph configuration.*

- bool [connected](#) () const
- bool [running](#) () const
- bool [send\\_telemetry\\_msg](#) (const pmt::pmt\_t &msg)  
*Sends a GNU Radio asynchronous message from telemetry to PVT.*
- std::shared\_ptr< [PvtInterface](#) > [get\\_pvt](#) ()  
*Returns a smart pointer to the PVT object.*
- void [prioritize\\_satellites](#) (const std::vector< std::pair< int, [Gnss\\_Satellite](#) > > &visible\_satellites)  
*Prioritize visible satellites in the specified vector.*

### 10.191.1 Detailed Description

This class represents a GNSS flow graph.  
It contains a signal source, a signal conditioner, a set of channels, a PVT and an output filter.  
Definition at line 66 of file [gnss\\_flowgraph.h](#).

### 10.191.2 Constructor & Destructor Documentation

#### 10.191.2.1 GNSSFlowgraph()

```
GNSSFlowgraph::GNSSFlowgraph (
    std::shared_ptr< ConfigurationInterface > configuration,
    std::shared_ptr< Concurrent\_Queue< pmt::pmt_t > > queue)
```

Constructor that initializes the receiver flow graph.

#### 10.191.2.2 ~GNSSFlowgraph()

```
GNSSFlowgraph::~GNSSFlowgraph ()
```

Destructor.

### 10.191.3 Member Function Documentation

#### 10.191.3.1 acquisition\_manager()

```
void GNSSFlowgraph::acquisition_manager (
    unsigned int who)
```

Manage satellite acquisition.

##### Parameters

in	who	<a href="#">Channel ID</a>
----	-----	----------------------------

#### 10.191.3.2 apply\_action()

```
void GNSSFlowgraph::apply_action (
    unsigned int who,
    unsigned int what)
```

Applies an action to the flow graph.

##### Parameters

in	who	Who generated the action
in	what	What is the action. 0: acquisition failed; 1: acquisition success; 2: tracking lost

**10.191.3.3 connect()**

```
void GNSSFlowgraph::connect ()
```

Connects the defined blocks in the flow graph.

Signal Source > Signal conditioner > Channels >> Observables >> PVT > Output filter

**10.191.3.4 connected()**

```
bool GNSSFlowgraph::connected () const [inline]
```

Definition at line 130 of file [gnss\\_flowgraph.h](#).

**10.191.3.5 disconnect()**

```
void GNSSFlowgraph::disconnect ()
```

Disconnect the blocks in the flow graph.

**10.191.3.6 get\_pvt()**

```
std::shared_ptr< PvtInterface > GNSSFlowgraph::get_pvt () [inline]
```

Returns a smart pointer to the PVT object.

Definition at line 150 of file [gnss\\_flowgraph.h](#).

**10.191.3.7 prioritize\_satellites()**

```
void GNSSFlowgraph::prioritize_satellites (
    const std::vector< std::pair< int, Gnss_Satellite > > & visible_satellites)
```

Prioritize visible satellites in the specified vector.

**10.191.3.8 running()**

```
bool GNSSFlowgraph::running () const [inline]
```

Definition at line 135 of file [gnss\\_flowgraph.h](#).

**10.191.3.9 send\_telemetry\_msg()**

```
bool GNSSFlowgraph::send_telemetry_msg (
    const pmt::pmt_t & msg)
```

Sends a GNU Radio asynchronous message from telemetry to PVT.

It is used to assist the receiver with external ephemeris data

**10.191.3.10 set\_configuration()**

```
void GNSSFlowgraph::set_configuration (
    const std::shared_ptr< ConfigurationInterface > & configuration)
```

Set flow graph configuration.

**10.191.3.11 start()**

```
void GNSSFlowgraph::start ()
```

Start the flow graph.

**10.191.3.12 stop()**

```
void GNSSFlowgraph::stop ()
```

Stop the flow graph.



**10.191.3.13 wait()**

```
void GNSSFlowgraph::wait ()
```

Wait for a flowgraph to complete.

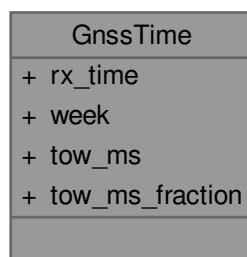
Flowgraphs complete when either (1) all blocks indicate that they are done, or (2) after [stop\(\)](#) has been called to request shutdown.

The documentation for this class was generated from the following file:

- [gnss\\_flowgraph.h](#)

**10.192 GnssTime Class Reference**

Collaboration diagram for GnssTime:

**Public Attributes**

- double [rx\\_time](#)
- int [week](#)
- int [tow\\_ms](#)
- double [tow\\_ms\\_fraction](#)

**10.192.1 Detailed Description**

Definition at line 23 of file [gnss\\_time.h](#).

**10.192.2 Member Data Documentation****10.192.2.1 rx\_time**

```
double GnssTime::rx_time
```

Definition at line 26 of file [gnss\\_time.h](#).

**10.192.2.2 tow\_ms**

```
int GnssTime::tow_ms
```

Definition at line 28 of file [gnss\\_time.h](#).

**10.192.2.3 tow\_ms\_fraction**

```
double GnssTime::tow_ms_fraction
```

Definition at line 29 of file [gnss\\_time.h](#).

#### 10.192.2.4 week

```
int Gnsstime::week
```

GPS week number (since January 1980)

Definition at line 27 of file [gnss\\_time.h](#).

The documentation for this class was generated from the following file:

- [gnss\\_time.h](#)

### 10.193 Gps\_Acq\_Assist Class Reference

This class is a storage for the GPS GSM RRLP acquisition assistance data as described in Digital cellular telecommunications system (Phase 2+); Location Services (LCS); Mobile Station (MS) - Serving Mobile Location Centre (SM-LC) Radio Resource LCS Protocol (RRLP) (3GPP TS 44.031 version 5.12.0 Release 5)

```
#include <gps_acq_assist.h>
```

Collaboration diagram for Gps\_Acq\_Assist:

Gps_Acq_Assist
+ PRN
+ tow
+ Doppler0
+ Doppler1
+ dopplerUncertainty
+ Code_Phase
+ Code_Phase_int
+ GPS_Bit_Number
+ Code_Phase_window
+ Azimuth
+ Elevation
+ Gps_Acq_Assist()

#### Public Member Functions

- [Gps\\_Acq\\_Assist](#) ()=default

#### Public Attributes

- uint32\_t [PRN](#) {}  
*SV PRN NUMBER.*
- double [tow](#) {}  
*Time Of Week assigned to the acquisition data.*
- double [Doppler0](#) {}  
*Doppler (0 order term) [Hz].*
- double [Doppler1](#) {}  
*Doppler (1 order term) [Hz].*
- double [dopplerUncertainty](#) {}

- Doppler Uncertainty [Hz].*
- double [Code\\_Phase](#) {}  
*Code phase [chips].*
- double [Code\\_Phase\\_int](#) {}  
*Integer Code Phase [1 C/A code period].*
- double [GPS\\_Bit\\_Number](#) {}  
*GPS Bit Number.*
- double [Code\\_Phase\\_window](#) {}  
*Code Phase search window [chips].*
- double [Azimuth](#) {}  
*Satellite Azimuth [deg].*
- double [Elevation](#) {}  
*Satellite Elevation [deg].*

### 10.193.1 Detailed Description

This class is a storage for the GPS GSM RRLT acquisition assistance data as described in Digital cellular telecommunications system (Phase 2+); Location Services (LCS); Mobile Station (MS) - Serving Mobile Location Centre (SM-LC) Radio Resource LCS Protocol (RRLP) (3GPP TS 44.031 version 5.12.0 Release 5)

Definition at line 37 of file [gps\\_acq\\_assist.h](#).

### 10.193.2 Constructor & Destructor Documentation

#### 10.193.2.1 Gps\_Acq\_Assist()

```
Gps_Acq_Assist::Gps_Acq_Assist () [default]
```

Default constructor

### 10.193.3 Member Data Documentation

#### 10.193.3.1 Azimuth

```
double Gps_Acq_Assist::Azimuth {}
```

Satellite Azimuth [deg].

Definition at line 54 of file [gps\\_acq\\_assist.h](#).

#### 10.193.3.2 Code\_Phase

```
double Gps_Acq_Assist::Code_Phase {}
```

Code phase [chips].

Definition at line 50 of file [gps\\_acq\\_assist.h](#).

#### 10.193.3.3 Code\_Phase\_int

```
double Gps_Acq_Assist::Code_Phase_int {}
```

Integer Code Phase [1 C/A code period].

Definition at line 51 of file [gps\\_acq\\_assist.h](#).

#### 10.193.3.4 Code\_Phase\_window

```
double Gps_Acq_Assist::Code_Phase_window {}
```

Code Phase search window [chips].

Definition at line 53 of file [gps\\_acq\\_assist.h](#).

#### 10.193.3.5 Doppler0

```
double Gps_Acq_Assist::Doppler0 {}
```

Doppler (0 order term) [Hz].

Definition at line 47 of file [gps\\_acq\\_assist.h](#).

#### 10.193.3.6 Doppler1

```
double Gps_Acq_Assist::Doppler1 {}
```

Doppler (1 order term) [Hz].  
Definition at line 48 of file [gps\\_acq\\_assist.h](#).

#### 10.193.3.7 dopplerUncertainty

```
double Gps_Acq_Assist::dopplerUncertainty {}
```

Doppler Uncertainty [Hz].  
Definition at line 49 of file [gps\\_acq\\_assist.h](#).

#### 10.193.3.8 Elevation

```
double Gps_Acq_Assist::Elevation {}
```

Satellite Elevation [deg].  
Definition at line 55 of file [gps\\_acq\\_assist.h](#).

#### 10.193.3.9 GPS\_Bit\_Number

```
double Gps_Acq_Assist::GPS_Bit_Number {}
```

GPS Bit Number.  
Definition at line 52 of file [gps\\_acq\\_assist.h](#).

#### 10.193.3.10 PRN

```
uint32_t Gps_Acq_Assist::PRN {}
```

SV PRN NUMBER.  
Definition at line 45 of file [gps\\_acq\\_assist.h](#).

#### 10.193.3.11 tow

```
double Gps_Acq_Assist::tow {}
```

Time Of Week assigned to the acquisition data.  
Definition at line 46 of file [gps\\_acq\\_assist.h](#).  
The documentation for this class was generated from the following file:

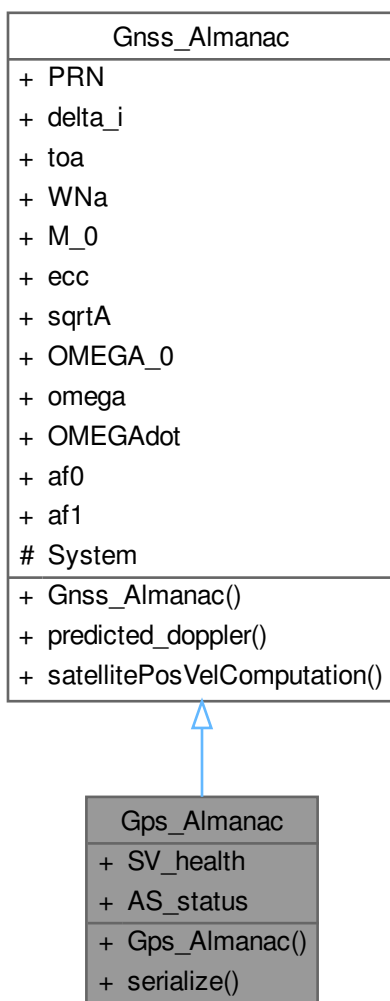
- [gps\\_acq\\_assist.h](#)

### 10.194 Gps\_Almanac Class Reference

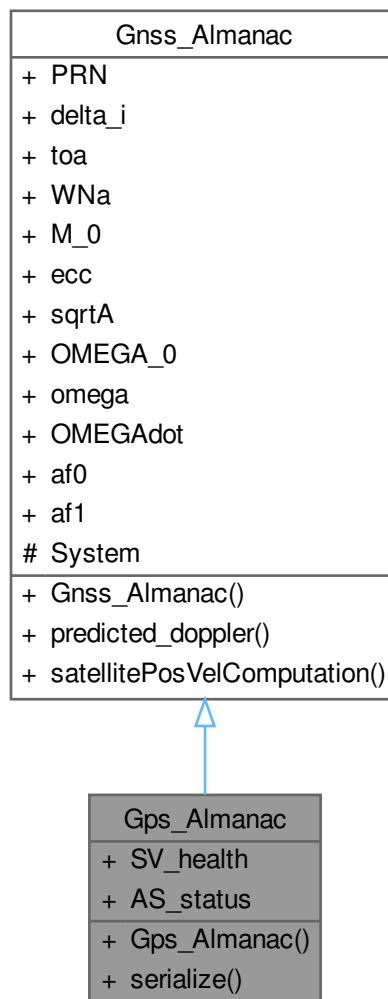
This class is a storage for the GPS SV ALMANAC data as described in IS-GPS-200M.  

```
#include <gps_almanac.h>
```

Inheritance diagram for Gps\_Almanac:



Collaboration diagram for Gps\_Almanac:



### Public Member Functions

- [Gps\\_Almanac](#) ()
- `template<class Archive>`  
void [serialize](#) (Archive &ar, const unsigned int version)

### Public Member Functions inherited from [Gnss\\_Almanac](#)

- [Gnss\\_Almanac](#) ()=default
- double [predicted\\_doppler](#) (double rx\_time\_s, double lat, double lon, double h, double ve, double vn, double vu, int band) const  
*Computes prediction of the Doppler shift for a given time and receiver's position and velocity.*
- void [satellitePosVelComputation](#) (double transmitTime, std::array< double, 7 > &pos\_vel\_dtr) const  
*Computes satellite Position and Velocity, in ECEF, for a given time (expressed in seconds of week)*

**Public Attributes**

- int32\_t [SV\\_health](#) {}  
*SV Health.*
- int32\_t [AS\\_status](#) {}  
*Anti-Spoofing Flags and SV Configuration.*

**Public Attributes inherited from [Gnss\\_Almanac](#)**

- uint32\_t [PRN](#) {}  
*SV PRN NUMBER.*
- double [delta\\_i](#) {}  
*Inclination Angle at Reference Time (relative to  $i_0 = 0.30$  semi-circles)*
- int32\_t [toa](#) {}  
*Almanac data reference time of week [s].*
- int32\_t [WNa](#) {}  
*Almanac week number.*
- double [M\\_0](#) {}  
*Mean Anomaly at Reference Time [semi-circles].*
- double [ecc](#) {}  
*Eccentricity [dimensionless].*
- double [sqrtA](#) {}  
*Square Root of the Semi-Major Axis [sqrt(m)].*
- double [OMEGA\\_0](#) {}  
*Longitude of Ascending Node of Orbit Plane at Weekly Epoch [semi-circles].*
- double [omega](#) {}  
*Argument of Perigee [semi-circles].*
- double [OMEGAdot](#) {}  
*Rate of Right Ascension [semi-circles/s].*
- double [af0](#) {}  
*Coefficient 0 of code phase offset model [s].*
- double [af1](#) {}  
*Coefficient 1 of code phase offset model [s/s].*

**Additional Inherited Members****Protected Attributes inherited from [Gnss\\_Almanac](#)**

- char [System](#) {}  
*Character ID of the GNSS system. 'G': GPS. 'E': Galileo. 'B': BeiDou.*

**10.194.1 Detailed Description**

This class is a storage for the GPS SV ALMANAC data as described in IS-GPS-200M.

See <https://www.gps.gov/technical/icwg/IS-GPS-200M.pdf> Appendix II  
Definition at line 35 of file [gps\\_almanac.h](#).

**10.194.2 Constructor & Destructor Documentation****10.194.2.1 Gps\_Almanac()**

```
Gps_Almanac::Gps_Almanac () [inline]
```

Default constructor

Definition at line 41 of file [gps\\_almanac.h](#).

References [Gnss\\_Almanac::System](#).

### 10.194.3 Member Function Documentation

#### 10.194.3.1 serialize()

```
template<class Archive>
void Gps_Almanac::serialize (
    Archive & ar,
    const unsigned int version) [inline]
```

Definition at line 51 of file [gps\\_almanac.h](#).

### 10.194.4 Member Data Documentation

#### 10.194.4.1 AS\_status

```
int32_t Gps_Almanac::AS_status {}
```

Anti-Spoofing Flags and SV Configuration.  
Definition at line 47 of file [gps\\_almanac.h](#).

#### 10.194.4.2 SV\_health

```
int32_t Gps_Almanac::SV_health {}
```

SV Health.  
Definition at line 46 of file [gps\\_almanac.h](#).  
The documentation for this class was generated from the following file:

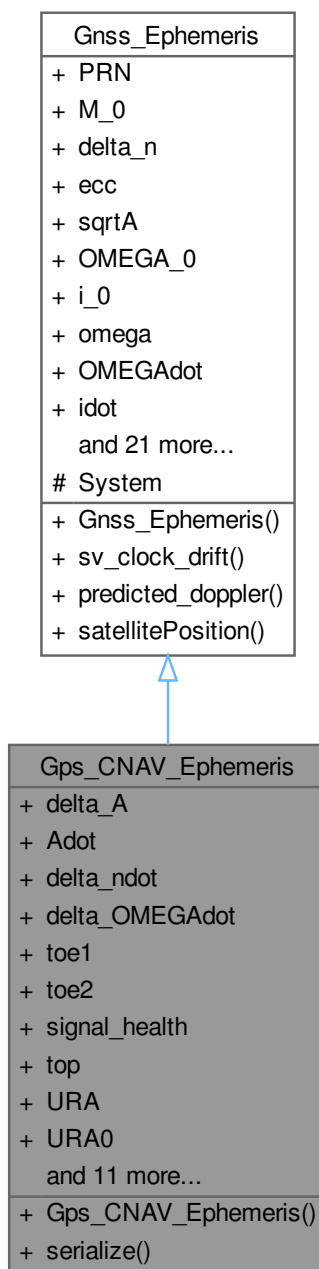
- [gps\\_almanac.h](#)

## 10.195 Gps\_CNAV\_Ephemeris Class Reference

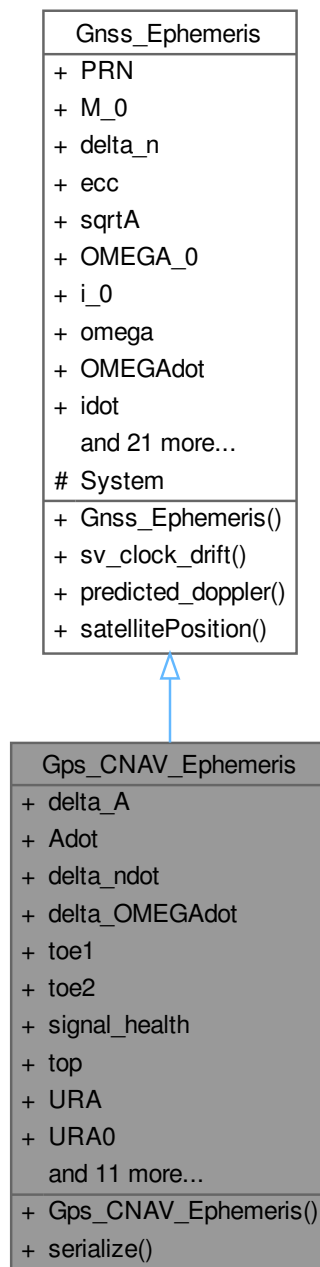
This is a storage class for the GPS CNAV ephemeris data as described in IS-GPS-200M.  
`#include <gps_cnav_ephemeris.h>`



Inheritance diagram for Gps\_CNAV\_Ephemeris:



Collaboration diagram for Gps\_CNAV\_Ephemeris:



## Public Member Functions

- [Gps\\_CNAV\\_Ephemeris](#) ()
- `template<class Archive>`  
void [serialize](#) (Archive &archive, const uint32\_t version)

*Serialize is a boost standard method to be called by the boost XML serialization. Here is used to save the ephemeris data on disk file.*

## Public Member Functions inherited from Gnss\_Ephemeris

- double [sv\\_clock\\_drift](#) (double transmitTime)  
*Sets (satClkDrift) and (dtr), and returns the clock drift in seconds according to the User Algorithm for SV Clock Correction (IS-GPS-200M, 20.3.3.3.3.1, and Galileo OS SIS ICD, 5.1.4).*
- double [predicted\\_doppler](#) (double rx\_time\_s, double lat, double lon, double h, double ve, double vn, double vu, int band) const  
*Computes prediction of the Doppler shift for a given time and receiver's position and velocity.*
- void [satellitePosition](#) (double transmitTime)  
*Computes the ECEF SV coordinates and ECEF velocity.*

## Public Attributes

- double [delta\\_A](#) {}  
*Semi-major axis difference at reference time.*
- double [Adot](#) {}  
*Change rate in semi-major axis.*
- double [delta\\_ndot](#) {}  
*Rate of mean motion difference from computed value.*
- double [delta\\_OMEGAdot](#) {}  
*Rate of Right Ascension difference [semi-circles/s].*
- int32\_t [toe1](#) {}  
*Ephemeris data reference time of week (Ref. 20.3.3.4.3 IS-GPS-200M) [s].*
- int32\_t [toe2](#) {}  
*Ephemeris data reference time of week (Ref. 20.3.3.4.3 IS-GPS-200M) [s].*
- int32\_t [signal\\_health](#) {}  
*Signal health (L1/L2/L5)*
- int32\_t [top](#) {}  
*Data predict time of week.*
- int32\_t [URA](#) {}  
*ED Accuracy Index.*
- double [URA0](#) {}  
*NED Accuracy Index.*
- double [URA1](#) {}  
*NED Accuracy Change Index.*
- double [URA2](#) {}  
*NED Accuracy Change Rate Index.*
- double [TGD](#) {}  
*Estimated Group Delay Differential: L1-L2 correction term only for the benefit of "L1 P(Y)" or "L2 P(Y)" s users [s].*
- double [ISCL1](#) {}
- double [ISCL2](#) {}
- double [ISCL5I](#) {}
- double [ISCL5Q](#) {}
- bool [integrity\\_status\\_flag](#) {}  
*If true, enhanced level of integrity assurance.*
- bool [l2c\\_phasing\\_flag](#) {}
- bool [alert\\_flag](#) {}  
*If true, indicates that the SV URA may be worse than indicated in d\_SV\_accuracy, use that SV at our own risk.*
- bool [antispoofing\\_flag](#) {}  
*If true, the AntiSpoofing mode is ON in that SV.*

## Public Attributes inherited from Gnss\_Ephemeris

- uint32\_t **PRN** {}  
SV ID.
- double **M\_0** {}  
Mean anomaly at reference time [rad].
- double **delta\_n** {}  
Mean motion difference from computed value [rad/sec].
- double **ecc** {}  
Eccentricity.
- double **sqrtA** {}  
Square root of the semi-major axis [meters<sup>1/2</sup>].
- double **OMEGA\_0** {}  
Longitude of ascending node of orbital plane at weekly epoch [rad].
- double **i\_0** {}  
Inclination angle at reference time [rad].
- double **omega** {}  
Argument of perigee [rad].
- double **OMEGAdot** {}  
Rate of right ascension [rad/sec].
- double **idot** {}  
Rate of inclination angle [rad/sec].
- double **Cuc** {}  
Amplitude of the cosine harmonic correction term to the argument of latitude [rad].
- double **Cus** {}  
Amplitude of the sine harmonic correction term to the argument of latitude [rad].
- double **Crc** {}  
Amplitude of the cosine harmonic correction term to the orbit radius [meters].
- double **Crs** {}  
Amplitude of the sine harmonic correction term to the orbit radius [meters].
- double **Cic** {}  
Amplitude of the cosine harmonic correction term to the angle of inclination [rad].
- double **Cis** {}  
Amplitude of the sine harmonic correction term to the angle of inclination [rad].
- int32\_t **toe** {}  
Ephemeris reference time [s].
- int32\_t **toc** {}  
Clock correction data reference Time of Week [sec].
- double **af0** {}  
SV clock bias correction coefficient [s].
- double **af1** {}  
SV clock drift correction coefficient [s/s].
- double **af2** {}  
SV clock drift rate correction coefficient [s/s<sup>2</sup>].
- double **satClkDrift** {}  
SV clock drift.
- double **dtr** {}  
Relativistic clock correction term.
- int32\_t **WN** {}  
Week number.
- int32\_t **tow** {}

*Time of Week.*

- double [satpos\\_X](#) {}

*Earth-fixed coordinate x of the satellite [m]. Intersection of the IERS Reference Meridian (IRM) and the plane passing through the origin and normal to the Z-axis.*

- double [satpos\\_Y](#) {}

*Earth-fixed coordinate y of the satellite [m]. Completes a right-handed, Earth-Centered, Earth-Fixed orthogonal coordinate system.*

- double [satpos\\_Z](#) {}

*Earth-fixed coordinate z of the satellite [m]. The direction of the IERS (International Earth Rotation and Reference Systems Service) Reference Pole (IRP).*

- double [satvel\\_X](#) {}

*Earth-fixed velocity coordinate x of the satellite [m].*

- double [satvel\\_Y](#) {}

*Earth-fixed velocity coordinate y of the satellite [m].*

- double [satvel\\_Z](#) {}

*Earth-fixed velocity coordinate z of the satellite [m].*

### Additional Inherited Members

### Protected Attributes inherited from [Gnss\\_Ephemeris](#)

- char [System](#) {}

*Character ID of the GNSS system. 'G': GPS. 'E': Galileo. 'B': BeiDou.*

## 10.195.1 Detailed Description

This is a storage class for the GPS CNAV ephemeris data as described in IS-GPS-200M.

See <https://www.gps.gov/technical/icwg/IS-GPS-200M.pdf> Appendix III

Definition at line 37 of file [gps\\_cnav\\_ephemeris.h](#).

## 10.195.2 Constructor & Destructor Documentation

### 10.195.2.1 Gps\_CNAV\_Ephemeris()

```
Gps_CNAV_Ephemeris::Gps_CNAV_Ephemeris () [inline]
```

Constructor

Definition at line 43 of file [gps\\_cnav\\_ephemeris.h](#).

References [Gnss\\_Ephemeris::System](#).

## 10.195.3 Member Function Documentation

### 10.195.3.1 serialize()

```
template<class Archive>
void Gps_CNAV_Ephemeris::serialize (
    Archive & archive,
    const uint32_t version) [inline]
```

Serialize is a boost standard method to be called by the boost XML serialization. Here is used to save the ephemeris data on disk file.

Definition at line 93 of file [gps\\_cnav\\_ephemeris.h](#).

References [Adot](#), [Gnss\\_Ephemeris::af0](#), [Gnss\\_Ephemeris::af1](#), [Gnss\\_Ephemeris::af2](#), [alert\\_flag](#), [antispoofing\\_flag](#), [Gnss\\_Ephemeris::Cic](#), [Gnss\\_Ephemeris::Cis](#), [Gnss\\_Ephemeris::Crc](#), [Gnss\\_Ephemeris::Crs](#), [Gnss\\_Ephemeris::Cuc](#), [Gnss\\_Ephemeris::Cus](#), [delta\\_A](#), [Gnss\\_Ephemeris::delta\\_n](#), [delta\\_OMEGAAdot](#), [Gnss\\_Ephemeris::dtr](#), [Gnss\\_Ephemeris::ecc](#), [Gnss\\_Ephemeris::i\\_0](#), [Gnss\\_Ephemeris::idot](#), [integrity\\_status\\_flag](#), [Gnss\\_Ephemeris::M\\_0](#), [Gnss\\_Ephemeris::omega](#), [Gnss\\_Ephemeris::OMEGA\\_0](#), [Gnss\\_Ephemeris::OMEGAAdot](#), [Gnss\\_Ephemeris::PRN](#), [Gnss\\_Ephemeris::satClkDrift](#), [Gnss\\_Ephemeris::sqrtA](#), [TGD](#), [Gnss\\_Ephemeris::toc](#), [Gnss\\_Ephemeris::toe](#), [toe1](#), [toe2](#), [Gnss\\_Ephemeris::tow](#), and [Gnss\\_Ephemeris::WN](#).

## 10.195.4 Member Data Documentation

### 10.195.4.1 Adot

```
double Gps_CNAV_Ephemeris::Adot {}
```

Change rate in semi-major axis.

Definition at line 49 of file [gps\\_cnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

### 10.195.4.2 alert\_flag

```
bool Gps_CNAV_Ephemeris::alert_flag {}
```

If true, indicates that the SV URA may be worse than indicated in d\_SV\_accuracy, use that SV at our own risk.

Definition at line 84 of file [gps\\_cnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

### 10.195.4.3 antispoofing\_flag

```
bool Gps_CNAV_Ephemeris::antispoofing_flag {}
```

If true, the AntiSpoofing mode is ON in that SV.

Definition at line 85 of file [gps\\_cnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

### 10.195.4.4 delta\_A

```
double Gps_CNAV_Ephemeris::delta_A {}
```

Semi-major axis difference at reference time.

Definition at line 48 of file [gps\\_cnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

### 10.195.4.5 delta\_ndot

```
double Gps_CNAV_Ephemeris::delta_ndot {}
```

Rate of mean motion difference from computed value.

Definition at line 50 of file [gps\\_cnav\\_ephemeris.h](#).

### 10.195.4.6 delta\_OMEGAdot

```
double Gps_CNAV_Ephemeris::delta_OMEGAdot {}
```

Rate of Right Ascension difference [semi-circles/s].

Definition at line 51 of file [gps\\_cnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

### 10.195.4.7 integrity\_status\_flag

```
bool Gps_CNAV_Ephemeris::integrity_status_flag {}
```

If true, enhanced level of integrity assurance.

If false, indicates that the conveying signal is provided with the legacy level of integrity assurance. That is, the probability that the instantaneous URE of the conveying signal exceeds 4.42 times the upper bound value of the current broadcast URA index, for more than 5.2 seconds, without an accompanying alert, is less than 1E-5 per hour.

If true, indicates that the conveying signal is provided with an enhanced level of integrity assurance. That is, the probability that the instantaneous URE of the conveying signal exceeds 5.73 times the upper bound value of the current broadcast URA index, for more than 5.2 seconds, without an accompanying alert, is less than 1E-8 per hour.

Definition at line 82 of file [gps\\_cnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

### 10.195.4.8 ISCL1

```
double Gps_CNAV_Ephemeris::ISCL1 {}
```

Definition at line 64 of file [gps\\_cnav\\_ephemeris.h](#).

#### 10.195.4.9 ISCL2

```
double Gps_CNAV_Ephemeris::ISCL2 {}
```

Definition at line 65 of file [gps\\_cnav\\_ephemeris.h](#).

#### 10.195.4.10 ISCL5I

```
double Gps_CNAV_Ephemeris::ISCL5I {}
```

Definition at line 66 of file [gps\\_cnav\\_ephemeris.h](#).

#### 10.195.4.11 ISCL5Q

```
double Gps_CNAV_Ephemeris::ISCL5Q {}
```

Definition at line 67 of file [gps\\_cnav\\_ephemeris.h](#).

#### 10.195.4.12 l2c\_phasing\_flag

```
bool Gps_CNAV_Ephemeris::l2c_phasing_flag {}
```

Definition at line 83 of file [gps\\_cnav\\_ephemeris.h](#).

#### 10.195.4.13 signal\_health

```
int32_t Gps_CNAV_Ephemeris::signal_health {}
```

Signal health (L1/L2/L5)

Definition at line 54 of file [gps\\_cnav\\_ephemeris.h](#).

#### 10.195.4.14 TGD

```
double Gps_CNAV_Ephemeris::TGD {}
```

Estimated Group Delay Differential: L1-L2 correction term only for the benefit of "L1 P(Y)" or "L2 P(Y)" s users [s].

Definition at line 63 of file [gps\\_cnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.195.4.15 toe1

```
int32_t Gps_CNAV_Ephemeris::toe1 {}
```

Ephemeris data reference time of week (Ref. 20.3.3.4.3 IS-GPS-200M) [s].

Definition at line 52 of file [gps\\_cnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.195.4.16 toe2

```
int32_t Gps_CNAV_Ephemeris::toe2 {}
```

Ephemeris data reference time of week (Ref. 20.3.3.4.3 IS-GPS-200M) [s].

Definition at line 53 of file [gps\\_cnav\\_ephemeris.h](#).

Referenced by [serialize\(\)](#).

#### 10.195.4.17 top

```
int32_t Gps_CNAV_Ephemeris::top {}
```

Data predict time of week.

Definition at line 55 of file [gps\\_cnav\\_ephemeris.h](#).

#### 10.195.4.18 URA

```
int32_t Gps_CNAV_Ephemeris::URA {}
```

ED Accuracy Index.

Definition at line 56 of file [gps\\_cnav\\_ephemeris.h](#).

**10.195.4.19 URA0**

```
double Gps_CNAV_Ephemeris::URA0 {}
```

NED Accuracy Index.

Definition at line 58 of file [gps\\_cnav\\_ephemeris.h](#).

**10.195.4.20 URA1**

```
double Gps_CNAV_Ephemeris::URA1 {}
```

NED Accuracy Change Index.

Definition at line 59 of file [gps\\_cnav\\_ephemeris.h](#).

**10.195.4.21 URA2**

```
double Gps_CNAV_Ephemeris::URA2 {}
```

NED Accuracy Change Rate Index.

Definition at line 60 of file [gps\\_cnav\\_ephemeris.h](#).

The documentation for this class was generated from the following file:

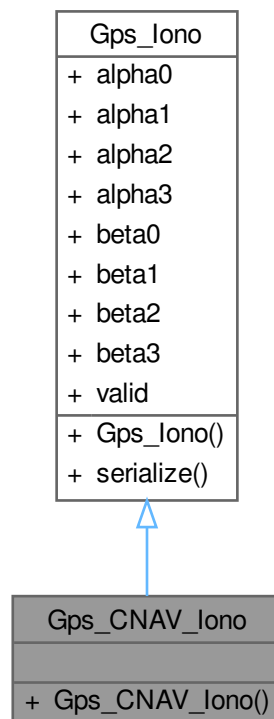
- [gps\\_cnav\\_ephemeris.h](#)

**10.196 Gps\_CNAV\_Iono Class Reference**

This class is a storage for the GPS IONOSPHERIC data as described in IS-GPS-200M.

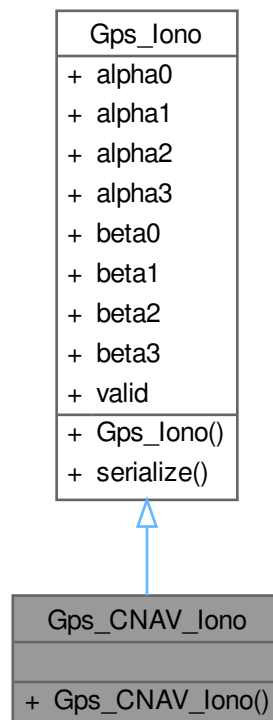
```
#include <gps_cnav_iono.h>
```

Inheritance diagram for Gps\_CNAV\_Iono:





Collaboration diagram for Gps\_CNAV\_Iono:



### Public Member Functions

- [Gps\\_CNAV\\_Iono](#) ()=default  
*Default constructor.*

### Public Member Functions inherited from [Gps\\_Iono](#)

- [Gps\\_Iono](#) ()=default  
*Default constructor.*
- template<class Archive>  
void [serialize](#) (Archive &archive, const unsigned int version)  
*Serialize is a boost standard method to be called by the boost XML serialization. Here is used to save the ephemeris data on disk file.*

### Additional Inherited Members

### Public Attributes inherited from [Gps\\_Iono](#)

- double [alpha0](#) {}  
*Coefficient 0 of a cubic equation representing the amplitude of the vertical delay [s].*
- double [alpha1](#) {}  
*Coefficient 1 of a cubic equation representing the amplitude of the vertical delay [s/semi-circle].*
- double [alpha2](#) {}  
*Coefficient 2 of a cubic equation representing the amplitude of the vertical delay [s(semi-circle)<sup>2</sup>].*

- double `alpha3` {}  
*Coefficient 3 of a cubic equation representing the amplitude of the vertical delay  $[s(\text{semi-circle})^3]$ .*
- double `beta0` {}  
*Coefficient 0 of a cubic equation representing the period of the model  $[s]$ .*
- double `beta1` {}  
*Coefficient 1 of a cubic equation representing the period of the model  $[s/\text{semi-circle}]$ .*
- double `beta2` {}  
*Coefficient 2 of a cubic equation representing the period of the model  $[s(\text{semi-circle})^2]$ .*
- double `beta3` {}  
*Coefficient 3 of a cubic equation representing the period of the model  $[s(\text{semi-circle})^3]$ .*
- bool `valid` {}  
*Valid flag.*

### 10.196.1 Detailed Description

This class is a storage for the GPS IONOSPHERIC data as described in IS-GPS-200M.

See <https://www.gps.gov/technical/icwg/IS-GPS-200M.pdf> Appendix III  
Definition at line 34 of file [gps\\_cnav\\_iono.h](#).

### 10.196.2 Constructor & Destructor Documentation

#### 10.196.2.1 Gps\_CNAV\_Iono()

```
Gps_CNAV_Iono::Gps_CNAV_Iono () [default]
```

Default constructor.

The documentation for this class was generated from the following file:

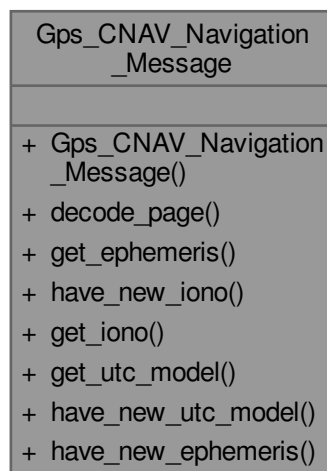
- [gps\\_cnav\\_iono.h](#)

## 10.197 Gps\_CNAV\_Navigation\_Message Class Reference

This class decodes a GPS CNAV Data message as described in IS-GPS-200M.

```
#include <gps_cnav_navigation_message.h>
```

Collaboration diagram for Gps\_CNAV\_Navigation\_Message:



## Public Member Functions

- [Gps\\_CNAV\\_Navigation\\_Message](#) ()
- void **decode\_page** (const std::bitset< GPS\_CNAV\_DATA\_PAGE\_BITS > &data\_bits)
- [Gps\\_CNAV\\_Ephemeris](#) **get\_ephemeris** () const  
*Obtain a GPS SV Ephemeris class filled with current SV data.*
- bool [have\\_new\\_iono](#) ()  
*Check if we have a new iono record stored in the GPS ephemeris class.*
- [Gps\\_CNAV\\_Iono](#) **get\_iono** () const  
*Obtain a GPS ionospheric correction parameters class filled with current SV data.*
- [Gps\\_CNAV\\_Utc\\_Model](#) **get\_utc\_model** ()  
*Obtain a GPS UTC model parameters class filled with current SV data.*
- bool [have\\_new\\_utc\\_model](#) ()
- bool [have\\_new\\_ephemeris](#) ()  
*Check if we have a new ephemeris stored in the GPS ephemeris class.*

### 10.197.1 Detailed Description

This class decodes a GPS CNAV Data message as described in IS-GPS-200M.

See <https://www.gps.gov/technical/icwg/IS-GPS-200M.pdf> Appendix III Definition at line 44 of file [gps\\_cnav\\_navigation\\_message.h](#).

### 10.197.2 Constructor & Destructor Documentation

#### 10.197.2.1 Gps\_CNAV\_Navigation\_Message()

`Gps_CNAV_Navigation_Message::Gps_CNAV_Navigation_Message ()`  
Default constructor

### 10.197.3 Member Function Documentation

#### 10.197.3.1 get\_ephemeris()

`Gps_CNAV_Ephemeris Gps_CNAV_Navigation_Message::get_ephemeris () const`  
Obtain a GPS SV Ephemeris class filled with current SV data.

#### 10.197.3.2 get\_iono()

`Gps_CNAV_Iono Gps_CNAV_Navigation_Message::get_iono () const`  
Obtain a GPS ionospheric correction parameters class filled with current SV data.

#### 10.197.3.3 get\_utc\_model()

`Gps_CNAV_Utc_Model Gps_CNAV_Navigation_Message::get_utc_model ()`  
Obtain a GPS UTC model parameters class filled with current SV data.

#### 10.197.3.4 have\_new\_ephemeris()

`bool Gps_CNAV_Navigation_Message::have_new_ephemeris ()`  
Check if we have a new ephemeris stored in the GPS ephemeris class.

#### 10.197.3.5 have\_new\_iono()

`bool Gps_CNAV_Navigation_Message::have_new_iono ()`  
Check if we have a new iono record stored in the GPS ephemeris class.

### 10.197.3.6 have\_new\_utc\_model()

```
bool Gps_CNAV_Navigation_Message::have_new_utc_model ()
```

\brief Check if we have a new GPS UTC model record stored in the GPS ephemeris class

The documentation for this class was generated from the following file:

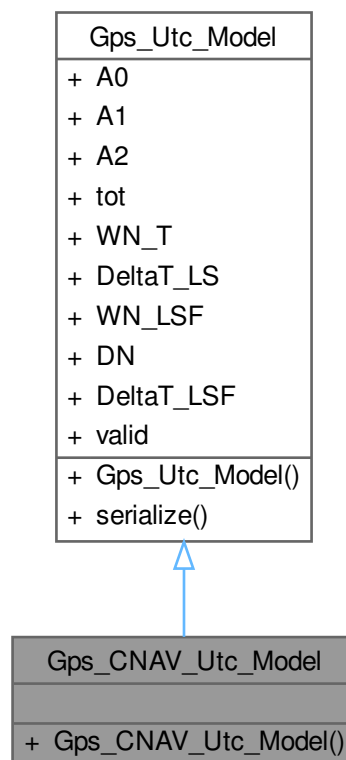
- [gps\\_cnav\\_navigation\\_message.h](#)

## 10.198 Gps\_CNAV\_Utc\_Model Class Reference

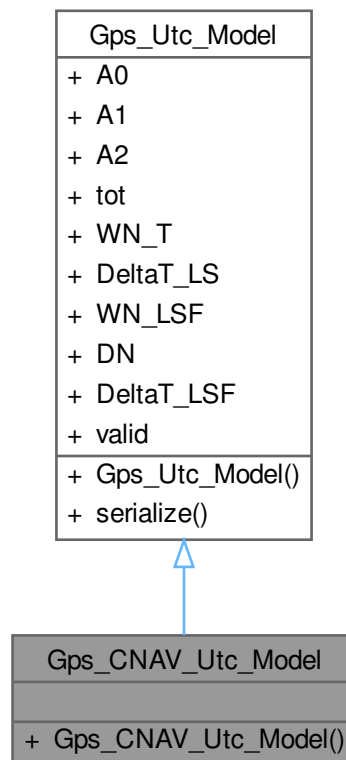
This class is a storage for the GPS UTC MODEL data as described in in IS-GPS-200M.

```
#include <gps_cnav_utc_model.h>
```

Inheritance diagram for Gps\_CNAV\_Utc\_Model:



Collaboration diagram for Gps\_CNAV\_Utc\_Model:



### Public Member Functions

- [Gps\\_CNAV\\_Utc\\_Model](#) ()=default  
*Default constructor.*

### Public Member Functions inherited from [Gps\\_Utc\\_Model](#)

- [Gps\\_Utc\\_Model](#) ()=default
- `template<class Archive>`  
void [serialize](#) (Archive &archive, const uint32\_t version)

### Additional Inherited Members

### Public Attributes inherited from [Gps\\_Utc\\_Model](#)

- double [A0](#) {}  
*Constant of a model that relates GPS and UTC time (ref. 20.3.3.5.2.4 IS-GPS-200M) [s].*
- double [A1](#) {}  
*1st order term of a model that relates GPS and UTC time (ref. 20.3.3.5.2.4 IS-GPS-200M) [s/s]*
- double [A2](#) {}  
*2nd order term of a model that relates GPS and UTC time (ref. 20.3.3.5.2.4 IS-GPS-200M) [s/s]*
- int32\_t [tot](#) {}  
*Reference time for UTC data (reference 20.3.4.5 and 20.3.3.5.2.4 IS-GPS-200M) [s].*

- int32\_t [WN\\_T](#) {}  
*UTC reference week number [weeks].*
- int32\_t [DeltaT\\_LS](#) {}  
*Delta time due to leap seconds [s]. Number of leap seconds since 6-Jan-1980 as transmitted by the GPS almanac.*
- int32\_t [WN\\_LSF](#) {}  
*Week number at the end of which the leap second becomes effective [weeks].*
- int32\_t [DN](#) {}  
*Day number (DN) at the end of which the leap second becomes effective [days].*
- int32\_t [DeltaT\\_LSF](#) {}  
*Scheduled future or recent past (relative to NAV message upload) value of the delta time due to leap seconds [s].*
- bool [valid](#) {}

### 10.198.1 Detailed Description

This class is a storage for the GPS UTC MODEL data as described in in IS-GPS-200M.

See <https://www.gps.gov/technical/icwg/IS-GPS-200M.pdf> Appendix III

Definition at line 34 of file [gps\\_cnav\\_utc\\_model.h](#).

### 10.198.2 Constructor & Destructor Documentation

#### 10.198.2.1 Gps\_CNAV\_Utc\_Model()

```
Gps_CNAV_Utc_Model::Gps_CNAV_Utc_Model () [default]
```

Default constructor.

The documentation for this class was generated from the following file:

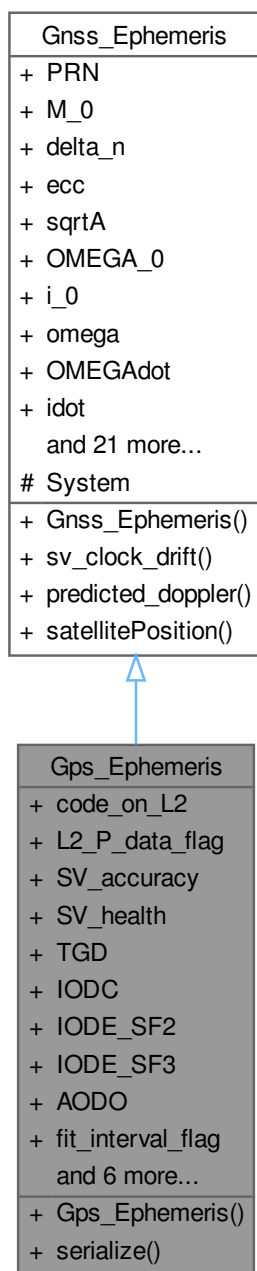
- [gps\\_cnav\\_utc\\_model.h](#)

## 10.199 Gps\_Ephemeris Class Reference

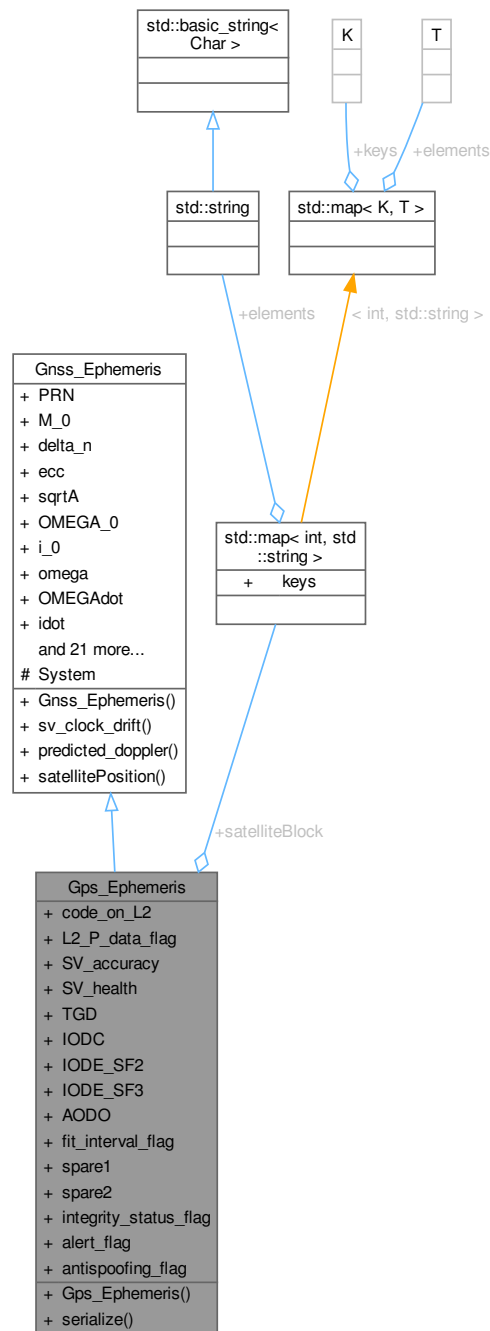
This class is a storage and orbital model functions for the GPS SV ephemeris data as described in IS-GPS-200M.

```
#include <gps_ephemeris.h>
```

Inheritance diagram for Gps\_Ephemeris:



Collaboration diagram for Gps\_Ephemeris:



## Public Member Functions

- [Gps\\_Ephemeris](#) ()
- `template<class Archive>`  
void [serialize](#) (Archive &archive, const uint32\_t version)

*Serialize is a boost standard method to be called by the boost XML serialization. Here is used to save the ephemeris data on disk file.*



## Public Member Functions inherited from Gnss\_Ephemeris

- double [sv\\_clock\\_drift](#) (double transmitTime)  
*Sets (satClkDrift) and (dtr), and returns the clock drift in seconds according to the User Algorithm for SV Clock Correction (IS-GPS-200M, 20.3.3.3.3.1, and Galileo OS SIS ICD, 5.1.4).*
- double [predicted\\_doppler](#) (double rx\_time\_s, double lat, double lon, double h, double ve, double vn, double vu, int band) const  
*Computes prediction of the Doppler shift for a given time and receiver's position and velocity.*
- void [satellitePosition](#) (double transmitTime)  
*Computes the ECEF SV coordinates and ECEF velocity.*

## Public Attributes

- int32\_t [code\\_on\\_L2](#) {}  
*If 1, P code ON in L2; if 2, C/A code ON in L2;.*
- bool [L2\\_P\\_data\\_flag](#) {}  
*When true, indicates that the NAV data stream was commanded OFF on the P-code of the L2 channel.*
- int32\_t [SV\\_accuracy](#) {}  
*User Range Accuracy (URA) index of the SV (reference paragraph 6.2.1) for the standard positioning service user (Ref 20.3.3.3.1.3 IS-GPS-200M)*
- int32\_t [SV\\_health](#) {}  
*Satellite health status.*
- double [TGD](#) {}  
*Estimated Group Delay Differential: L1-L2 correction term only for the benefit of "L1 P(Y)" or "L2 P(Y)" s users [s].*
- int32\_t [IODC](#) {}  
*Issue of Data, Clock.*
- int32\_t [IODE\\_SF2](#) {}  
*Issue of Data, Ephemeris (IODE), subframe 2.*
- int32\_t [IODE\\_SF3](#) {}  
*Issue of Data, Ephemeris (IODE), subframe 3.*
- int32\_t [AODO](#) {}  
*Age of Data Offset (AODO) term for the navigation message correction table (NMCT) contained in subframe 4 (reference paragraph 20.3.3.5.1.9) [s].*
- bool [fit\\_interval\\_flag](#) {}  
*indicates the curve-fit interval used by the CS (Block II/IIA/IIR/IIR-M/IIF) and SS (Block IIIA) in determining the ephemeris parameters, as follows: 0 = 4 hours, 1 = greater than 4 hours.*
- double [spare1](#) {}
- double [spare2](#) {}
- bool [integrity\\_status\\_flag](#) {}  
*If true, enhanced level of integrity assurance.*
- bool [alert\\_flag](#) {}  
*If true, indicates that the SV URA may be worse than indicated in d\_SV\_accuracy, use that SV at our own risk.*
- bool [antispoofing\\_flag](#) {}  
*If true, the AntiSpoofing mode is ON in that SV.*
- std::map< int, std::string > [satelliteBlock](#)  
*Map that stores to which block the PRN belongs <https://www.navcen.uscg.gov/?Do=constellation&Status>.*

## Public Attributes inherited from Gnss\_Ephemeris

- uint32\_t **PRN** {}  
SV ID.
- double **M\_0** {}  
Mean anomaly at reference time [rad].
- double **delta\_n** {}  
Mean motion difference from computed value [rad/sec].
- double **ecc** {}  
Eccentricity.
- double **sqrtA** {}  
Square root of the semi-major axis [meters<sup>1/2</sup>].
- double **OMEGA\_0** {}  
Longitude of ascending node of orbital plane at weekly epoch [rad].
- double **i\_0** {}  
Inclination angle at reference time [rad].
- double **omega** {}  
Argument of perigee [rad].
- double **OMEGAdot** {}  
Rate of right ascension [rad/sec].
- double **idot** {}  
Rate of inclination angle [rad/sec].
- double **Cuc** {}  
Amplitude of the cosine harmonic correction term to the argument of latitude [rad].
- double **Cus** {}  
Amplitude of the sine harmonic correction term to the argument of latitude [rad].
- double **Crc** {}  
Amplitude of the cosine harmonic correction term to the orbit radius [meters].
- double **Crs** {}  
Amplitude of the sine harmonic correction term to the orbit radius [meters].
- double **Cic** {}  
Amplitude of the cosine harmonic correction term to the angle of inclination [rad].
- double **Cis** {}  
Amplitude of the sine harmonic correction term to the angle of inclination [rad].
- int32\_t **toe** {}  
Ephemeris reference time [s].
- int32\_t **toc** {}  
Clock correction data reference Time of Week [sec].
- double **af0** {}  
SV clock bias correction coefficient [s].
- double **af1** {}  
SV clock drift correction coefficient [s/s].
- double **af2** {}  
SV clock drift rate correction coefficient [s/s<sup>2</sup>].
- double **satClkDrift** {}  
SV clock drift.
- double **dtr** {}  
Relativistic clock correction term.
- int32\_t **WN** {}  
Week number.
- int32\_t **tow** {}

*Time of Week.*

- double [satpos\\_X](#) {}

*Earth-fixed coordinate x of the satellite [m]. Intersection of the IERS Reference Meridian (IRM) and the plane passing through the origin and normal to the Z-axis.*

- double [satpos\\_Y](#) {}

*Earth-fixed coordinate y of the satellite [m]. Completes a right-handed, Earth-Centered, Earth-Fixed orthogonal coordinate system.*

- double [satpos\\_Z](#) {}

*Earth-fixed coordinate z of the satellite [m]. The direction of the IERS (International Earth Rotation and Reference Systems Service) Reference Pole (IRP).*

- double [satvel\\_X](#) {}

*Earth-fixed velocity coordinate x of the satellite [m].*

- double [satvel\\_Y](#) {}

*Earth-fixed velocity coordinate y of the satellite [m].*

- double [satvel\\_Z](#) {}

*Earth-fixed velocity coordinate z of the satellite [m].*

### Additional Inherited Members

### Protected Attributes inherited from [Gnss\\_Ephemeris](#)

- char [System](#) {}

*Character ID of the GNSS system. 'G': GPS. 'E': Galileo. 'B': BeiDou.*

## 10.199.1 Detailed Description

This class is a storage and orbital model functions for the GPS SV ephemeris data as described in IS-GPS-200M. See <https://www.gps.gov/technical/icwg/IS-GPS-200M.pdf> Appendix II Definition at line 40 of file [gps\\_ephemeris.h](#).

## 10.199.2 Constructor & Destructor Documentation

### 10.199.2.1 Gps\_Ephemeris()

```
Gps_Ephemeris::Gps_Ephemeris ()
```

Default constructor

## 10.199.3 Member Function Documentation

### 10.199.3.1 serialize()

```
template<class Archive>
void Gps_Ephemeris::serialize (
    Archive & archive,
    const uint32_t version) [inline]
```

Serialize is a boost standard method to be called by the boost XML serialization. Here is used to save the ephemeris data on disk file.

Definition at line 86 of file [gps\\_ephemeris.h](#).

References [Gnss\\_Ephemeris::af0](#), [Gnss\\_Ephemeris::af1](#), [Gnss\\_Ephemeris::af2](#), [alert\\_flag](#), [antispoofing\\_flag](#), [AODO](#), [Gnss\\_Ephemeris::Cic](#), [Gnss\\_Ephemeris::Cis](#), [code\\_on\\_L2](#), [Gnss\\_Ephemeris::Crc](#), [Gnss\\_Ephemeris::Crs](#), [Gnss\\_Ephemeris::Cuc](#), [Gnss\\_Ephemeris::Cus](#), [Gnss\\_Ephemeris::delta\\_n](#), [Gnss\\_Ephemeris::dtr](#), [Gnss\\_Ephemeris::ecc](#), [fit\\_interval\\_flag](#), [Gnss\\_Ephemeris::i\\_0](#), [Gnss\\_Ephemeris::idot](#), [integrity\\_status\\_flag](#), [IODC](#), [IODE\\_SF2](#), [IODE\\_SF3](#), [L2\\_P\\_data\\_flag](#), [Gnss\\_Ephemeris::M\\_0](#), [Gnss\\_Ephemeris::omega](#), [Gnss\\_Ephemeris::OMEGA\\_0](#), [Gnss\\_Ephemeris::OMEGAdot](#), [Gnss\\_Ephemeris::PRN](#), [Gnss\\_Ephemeris::satClkDrift](#), [Gnss\\_Ephemeris::sqrtA](#), [SV\\_accuracy](#), [SV\\_health](#), [TGD](#), [Gnss\\_Ephemeris::toc](#), [Gnss\\_Ephemeris::toe](#), [Gnss\\_Ephemeris::tow](#), and [Gnss\\_Ephemeris::WN](#).

## 10.199.4 Member Data Documentation

### 10.199.4.1 alert\_flag

```
bool Gps_Ephemeris::alert_flag {}
```

If true, indicates that the SV URA may be worse than indicated in d\_SV\_accuracy, use that SV at our own risk.

Definition at line 75 of file [gps\\_ephemeris.h](#).

Referenced by [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), and [serialize\(\)](#).

### 10.199.4.2 antispoofing\_flag

```
bool Gps_Ephemeris::antispoofing_flag {}
```

If true, the AntiSpoofing mode is ON in that SV.

Definition at line 76 of file [gps\\_ephemeris.h](#).

Referenced by [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), and [serialize\(\)](#).

### 10.199.4.3 AODO

```
int32_t Gps_Ephemeris::AODO {}
```

Age of Data Offset (AODO) term for the navigation message correction table (NMCT) contained in subframe 4 (reference paragraph 20.3.3.5.1.9) [s].

Definition at line 56 of file [gps\\_ephemeris.h](#).

Referenced by [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), and [serialize\(\)](#).

### 10.199.4.4 code\_on\_L2

```
int32_t Gps_Ephemeris::code_on_L2 {}
```

If 1, P code ON in L2; if 2, C/A code ON in L2;.

Definition at line 48 of file [gps\\_ephemeris.h](#).

Referenced by [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), and [serialize\(\)](#).

### 10.199.4.5 fit\_interval\_flag

```
bool Gps_Ephemeris::fit_interval_flag {}
```

indicates the curve-fit interval used by the CS (Block II/IIA/IIR/IIR-M/IIF) and SS (Block IIIA) in determining the ephemeris parameters, as follows: 0 = 4 hours, 1 = greater than 4 hours.

Definition at line 58 of file [gps\\_ephemeris.h](#).

Referenced by [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), and [serialize\(\)](#).

### 10.199.4.6 integrity\_status\_flag

```
bool Gps_Ephemeris::integrity_status_flag {}
```

If true, enhanced level of integrity assurance.

If false, indicates that the conveying signal is provided with the legacy level of integrity assurance. That is, the probability that the instantaneous URE of the conveying signal exceeds 4.42 times the upper bound value of the current broadcast URA index, for more than 5.2 seconds, without an accompanying alert, is less than 1E-5 per hour.

If true, indicates that the conveying signal is provided with an enhanced level of integrity assurance. That is, the probability that the instantaneous URE of the conveying signal exceeds 5.73 times the upper bound value of the current broadcast URA index, for more than 5.2 seconds, without an accompanying alert, is less than 1E-8 per hour.

Definition at line 74 of file [gps\\_ephemeris.h](#).

Referenced by [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), and [serialize\(\)](#).

### 10.199.4.7 IODC

```
int32_t Gps_Ephemeris::IODC {}
```

Issue of Data, Clock.

Definition at line 53 of file [gps\\_ephemeris.h](#).

Referenced by [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), and [serialize\(\)](#).

**10.199.4.8 IODE\_SF2**

```
int32_t Gps_Ephemeris::IODE_SF2 {}
```

Issue of Data, Ephemeris (IODE), subframe 2.

Definition at line 54 of file [gps\\_ephemeris.h](#).

Referenced by [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), and [serialize\(\)](#).

**10.199.4.9 IODE\_SF3**

```
int32_t Gps_Ephemeris::IODE_SF3 {}
```

Issue of Data, Ephemeris (IODE), subframe 3.

Definition at line 55 of file [gps\\_ephemeris.h](#).

Referenced by [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), and [serialize\(\)](#).

**10.199.4.10 L2\_P\_data\_flag**

```
bool Gps_Ephemeris::L2_P_data_flag {}
```

When true, indicates that the NAV data stream was commanded OFF on the P-code of the L2 channel.

Definition at line 49 of file [gps\\_ephemeris.h](#).

Referenced by [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), and [serialize\(\)](#).

**10.199.4.11 satelliteBlock**

```
std::map<int, std::string> Gps_Ephemeris::satelliteBlock
```

Map that stores to which block the PRN belongs <https://www.navcen.uscg.gov/?Do=constellation&Status>.

Definition at line 78 of file [gps\\_ephemeris.h](#).

**10.199.4.12 spare1**

```
double Gps_Ephemeris::spare1 {}
```

Definition at line 59 of file [gps\\_ephemeris.h](#).

**10.199.4.13 spare2**

```
double Gps_Ephemeris::spare2 {}
```

Definition at line 60 of file [gps\\_ephemeris.h](#).

**10.199.4.14 SV\_accuracy**

```
int32_t Gps_Ephemeris::SV_accuracy {}
```

User Range Accuracy (URA) index of the SV (reference paragraph 6.2.1) for the standard positioning service user (Ref 20.3.3.3.1.3 IS-GPS-200M)

Definition at line 50 of file [gps\\_ephemeris.h](#).

Referenced by [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), and [serialize\(\)](#).

**10.199.4.15 SV\_health**

```
int32_t Gps_Ephemeris::SV_health {}
```

Satellite health status.

Definition at line 51 of file [gps\\_ephemeris.h](#).

Referenced by [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), and [serialize\(\)](#).

**10.199.4.16 TGD**

```
double Gps_Ephemeris::TGD {}
```

Estimated Group Delay Differential: L1-L2 correction term only for the benefit of "L1 P(Y)" or "L2 P(Y)" s users [s].

Definition at line 52 of file [gps\\_ephemeris.h](#).

Referenced by [Serdes\\_Gps\\_Eph::readProtobuffer\(\)](#), and [serialize\(\)](#).

The documentation for this class was generated from the following file:

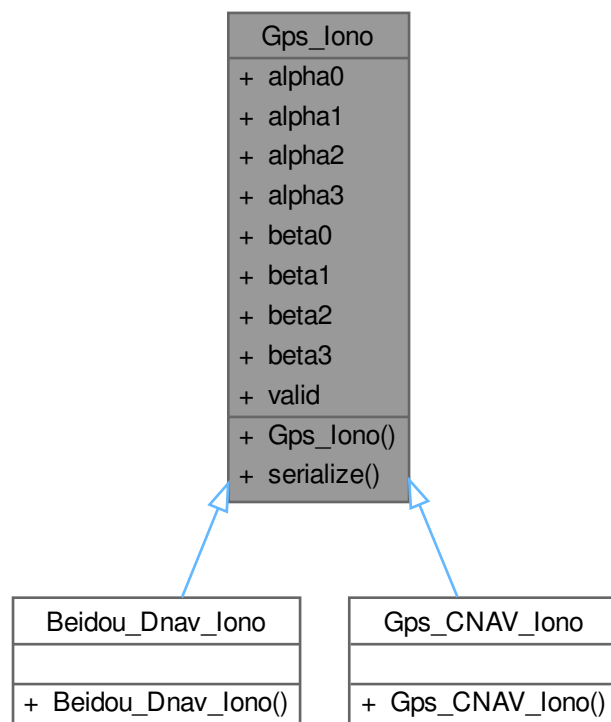
- [gps\\_ephemeris.h](#)

## 10.200 Gps\_Iono Class Reference

This class is a storage for the GPS IONOSPHERIC data as described in IS-GPS-200M.

```
#include <gps_iono.h>
```

Inheritance diagram for Gps\_Iono:



Collaboration diagram for Gps\_Iono:

Gps_Iono
+ alpha0
+ alpha1
+ alpha2
+ alpha3
+ beta0
+ beta1
+ beta2
+ beta3
+ valid
+ Gps_Iono()
+ serialize()

### Public Member Functions

- [Gps\\_Iono](#) ()=default  
*Default constructor.*
- `template<class Archive>`  
void [serialize](#) (Archive &archive, const unsigned int version)  
*Serialize is a boost standard method to be called by the boost XML serialization. Here is used to save the ephemeris data on disk file.*

### Public Attributes

- double [alpha0](#) {}  
*Coefficient 0 of a cubic equation representing the amplitude of the vertical delay [s].*
- double [alpha1](#) {}  
*Coefficient 1 of a cubic equation representing the amplitude of the vertical delay [s/semi-circle].*
- double [alpha2](#) {}  
*Coefficient 2 of a cubic equation representing the amplitude of the vertical delay [s(semi-circle)<sup>2</sup>].*
- double [alpha3](#) {}  
*Coefficient 3 of a cubic equation representing the amplitude of the vertical delay [s(semi-circle)<sup>3</sup>].*
- double [beta0](#) {}  
*Coefficient 0 of a cubic equation representing the period of the model [s].*
- double [beta1](#) {}  
*Coefficient 1 of a cubic equation representing the period of the model [s/semi-circle].*
- double [beta2](#) {}  
*Coefficient 2 of a cubic equation representing the period of the model [s(semi-circle)<sup>2</sup>].*
- double [beta3](#) {}  
*Coefficient 3 of a cubic equation representing the period of the model [s(semi-circle)<sup>3</sup>].*
- bool [valid](#) {}  
*Valid flag.*

### 10.200.1 Detailed Description

This class is a storage for the GPS IONOSPHERIC data as described in IS-GPS-200M.

See <https://www.gps.gov/technical/icwg/IS-GPS-200M.pdf> Appendix II

Definition at line 35 of file [gps\\_iono.h](#).

### 10.200.2 Constructor & Destructor Documentation

#### 10.200.2.1 Gps\_Iono()

```
Gps_Iono::Gps_Iono () [default]
```

Default constructor.

### 10.200.3 Member Function Documentation

#### 10.200.3.1 serialize()

```
template<class Archive>
void Gps_Iono::serialize (
    Archive & archive,
    const unsigned int version) [inline]
```

Serialize is a boost standard method to be called by the boost XML serialization. Here is used to save the ephemeris data on disk file.

Definition at line 58 of file [gps\\_iono.h](#).

References [alpha0](#), [alpha1](#), [alpha2](#), [alpha3](#), [beta0](#), [beta1](#), [beta2](#), and [beta3](#).

### 10.200.4 Member Data Documentation

#### 10.200.4.1 alpha0

```
double Gps_Iono::alpha0 {}
```

Coefficient 0 of a cubic equation representing the amplitude of the vertical delay [s].

Definition at line 41 of file [gps\\_iono.h](#).

Referenced by [serialize\(\)](#).

#### 10.200.4.2 alpha1

```
double Gps_Iono::alpha1 {}
```

Coefficient 1 of a cubic equation representing the amplitude of the vertical delay [s/semi-circle].

Definition at line 42 of file [gps\\_iono.h](#).

Referenced by [serialize\(\)](#).

#### 10.200.4.3 alpha2

```
double Gps_Iono::alpha2 {}
```

Coefficient 2 of a cubic equation representing the amplitude of the vertical delay [s(semi-circle)<sup>2</sup>].

Definition at line 43 of file [gps\\_iono.h](#).

Referenced by [serialize\(\)](#).

#### 10.200.4.4 alpha3

```
double Gps_Iono::alpha3 {}
```

Coefficient 3 of a cubic equation representing the amplitude of the vertical delay [s(semi-circle)<sup>3</sup>].

Definition at line 44 of file [gps\\_iono.h](#).

Referenced by [serialize\(\)](#).

#### 10.200.4.5 beta0

```
double Gps_Iono::beta0 {}
```

Coefficient 0 of a cubic equation representing the period of the model [s].



Definition at line 45 of file [gps\\_iono.h](#).  
Referenced by [serialize\(\)](#).

#### 10.200.4.6 beta1

```
double Gps_Iono::beta1 {}
```

Coefficient 1 of a cubic equation representing the period of the model [s/semi-circle].  
Definition at line 46 of file [gps\\_iono.h](#).  
Referenced by [serialize\(\)](#).

#### 10.200.4.7 beta2

```
double Gps_Iono::beta2 {}
```

Coefficient 2 of a cubic equation representing the period of the model [s(semi-circle)^2].  
Definition at line 47 of file [gps\\_iono.h](#).  
Referenced by [serialize\(\)](#).

#### 10.200.4.8 beta3

```
double Gps_Iono::beta3 {}
```

Coefficient 3 of a cubic equation representing the period of the model [s(semi-circle)^3].  
Definition at line 48 of file [gps\\_iono.h](#).  
Referenced by [serialize\(\)](#).

#### 10.200.4.9 valid

```
bool Gps_Iono::valid {}
```

Valid flag.  
Definition at line 50 of file [gps\\_iono.h](#).  
The documentation for this class was generated from the following file:

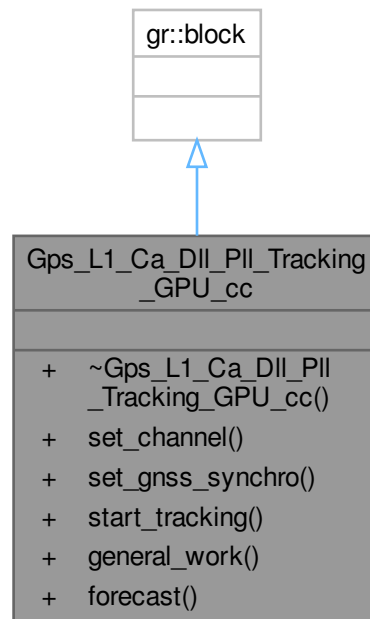
- [gps\\_iono.h](#)

## 10.201 Gps\_L1\_Ca\_Dll\_Pll\_Tracking\_GPU\_cc Class Reference

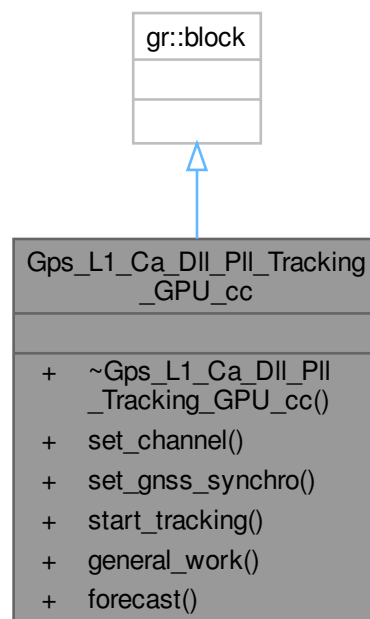
This class implements a DLL + PLL tracking loop block.  

```
#include <gps_l1_ca_dll_pll_tracking_gpu_cc.h>
```

Inheritance diagram for Gps\_L1\_Ca\_Dll\_Pll\_Tracking\_GPU\_cc:



Collaboration diagram for Gps\_L1\_Ca\_Dll\_Pll\_Tracking\_GPU\_cc:



**Public Member Functions**

- void **set\_channel** (uint32\_t channel)
- void **set\_gnss\_synchro** ([Gnss\\_Synchro](#) \*p\_gnss\_synchro)
- void **start\_tracking** ()
- int **general\_work** (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)
- void **forecast** (int noutput\_items, gr\_vector\_int &ninput\_items\_required)

**Friends**

- [gps\\_l1\\_ca\\_dll\\_pll\\_tracking\\_gpu\\_cc\\_sptr](#)    **gps\_l1\_ca\_dll\_pll\_make\_tracking\_gpu\_cc** (int64\_t fs\_in, uint32\_t vector\_length, bool dump, std::string dump\_filename, float pll\_bw\_hz, float dll\_bw\_hz, float early\_↔late\_space\_chips)

**10.201.1 Detailed Description**

This class implements a DLL + PLL tracking loop block.

Definition at line 60 of file [gps\\_l1\\_ca\\_dll\\_pll\\_tracking\\_gpu\\_cc.h](#).

The documentation for this class was generated from the following file:

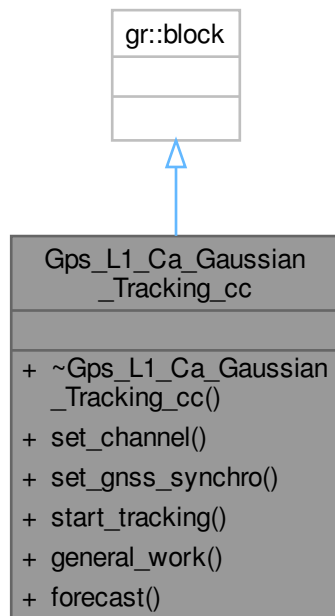
- [gps\\_l1\\_ca\\_dll\\_pll\\_tracking\\_gpu\\_cc.h](#)

**10.202 Gps\_L1\_Ca\_Gaussian\_Tracking\_cc Class Reference**

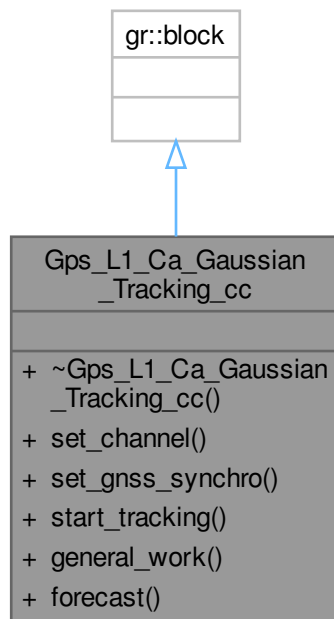
This class implements a DLL + PLL tracking loop block.

#include <gps\_l1\_ca\_gaussian\_tracking\_cc.h>

Inheritance diagram for Gps\_L1\_Ca\_Gaussian\_Tracking\_cc:



Collaboration diagram for Gps\_L1\_Ca\_Gaussian\_Tracking\_cc:



### Public Member Functions

- void **set\_channel** (uint32\_t channel)
- void **set\_gnss\_synchro** ([Gnss\\_Synchro](#) \*p\_gnss\_synchro)
- void **start\_tracking** ()
- int **general\_work** (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)
- void **forecast** (int noutput\_items, gr\_vector\_int &ninput\_items\_required)

### Friends

- `gps_l1_ca_gaussian_tracking_cc_sptr` **gps\_l1\_ca\_gaussian\_make\_tracking\_cc** (uint32\_t order, int64\_t fs\_in, uint32\_t vector\_length, bool dump, const std::string &dump\_filename, float dll\_bw\_hz, float early\_late\_space\_chips, bool bce\_run, uint32\_t bce\_ptrans, uint32\_t bce\_strans, int32\_t bce\_nu, int32\_t bce\_kappa)

### 10.202.1 Detailed Description

This class implements a DLL + PLL tracking loop block.

Definition at line 74 of file [gps\\_l1\\_ca\\_gaussian\\_tracking\\_cc.h](#).

The documentation for this class was generated from the following file:

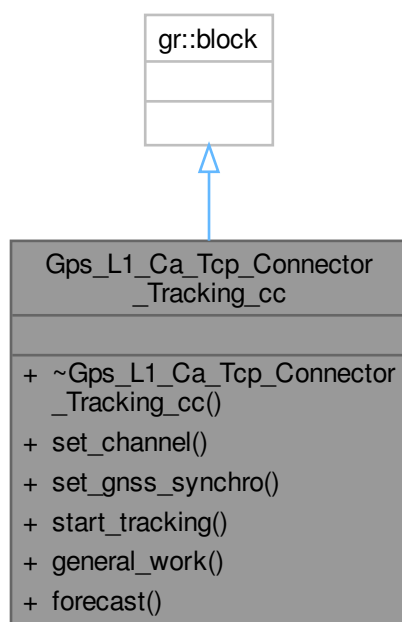
- [gps\\_l1\\_ca\\_gaussian\\_tracking\\_cc.h](#)

## 10.203 Gps\_L1\_Ca\_Tcp\_Connector\_Tracking\_cc Class Reference

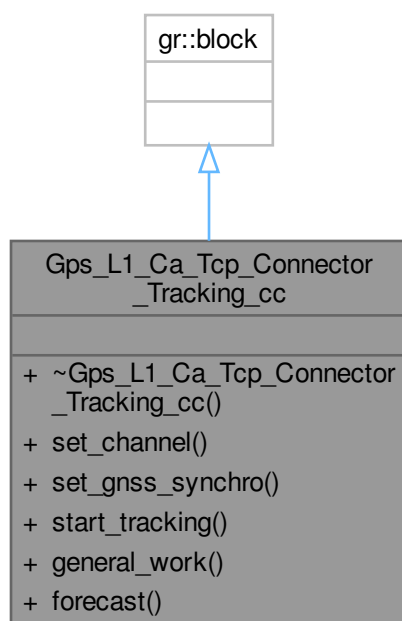
This class implements a DLL + PLL tracking loop block.

```
#include <gps_l1_ca_tcp_connector_tracking_cc.h>
```

Inheritance diagram for Gps\_L1\_Ca\_Tcp\_Connector\_Tracking\_cc:



Collaboration diagram for Gps\_L1\_Ca\_Tcp\_Connector\_Tracking\_cc:



### Public Member Functions

- void **set\_channel** (uint32\_t channel)
- void **set\_gnss\_synchro** ([Gnss\\_Synchro](#) \*p\_gnss\_synchro)
- void **start\_tracking** ()
- int **general\_work** (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)
- void **forecast** (int noutput\_items, gr\_vector\_int &ninput\_items\_required)

### Friends

- [gps\\_l1\\_ca\\_tcp\\_connector\\_tracking\\_cc\\_sptr](#) **gps\_l1\_ca\_tcp\_connector\_make\_tracking\_cc** (int64\_t fs\_in, uint32\_t vector\_length, bool dump, const std::string &dump\_filename, float early\_late\_space\_chips, size\_t port\_ch0)

### 10.203.1 Detailed Description

This class implements a DLL + PLL tracking loop block.

Definition at line 58 of file [gps\\_l1\\_ca\\_tcp\\_connector\\_tracking\\_cc.h](#).

The documentation for this class was generated from the following file:

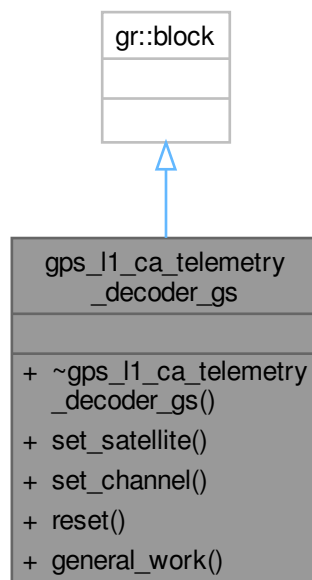
- [gps\\_l1\\_ca\\_tcp\\_connector\\_tracking\\_cc.h](#)

## 10.204 [gps\\_l1\\_ca\\_telemetry\\_decoder\\_gs](#) Class Reference

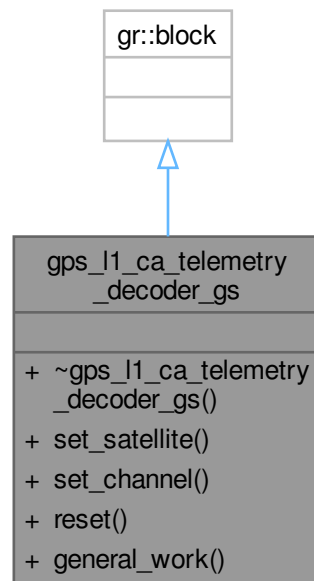
This class implements a block that decodes the NAV data defined in IS-GPS-200M.

```
#include <gps_l1_ca_telemetry_decoder_gs.h>
```

Inheritance diagram for [gps\\_l1\\_ca\\_telemetry\\_decoder\\_gs](#):



Collaboration diagram for `gps_l1_ca_telemetry_decoder_gs`:



### Public Member Functions

- void `set_satellite` (const [Gnss\\_Satellite](#) &satellite)  
*Set satellite PRN.*
- void `set_channel` (int channel)  
*Set receiver's channel.*
- void `reset` ()
- int `general_work` (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items) override  
*This is where all signal processing takes place.*

### Friends

- `gps_l1_ca_telemetry_decoder_gs_sptr` `gps_l1_ca_make_telemetry_decoder_gs` (const [Gnss\\_Satellite](#) &satellite, const [Tlm\\_Conf](#) &conf)

## 10.204.1 Detailed Description

This class implements a block that decodes the NAV data defined in IS-GPS-200M.  
Definition at line 55 of file [gps\\_l1\\_ca\\_telemetry\\_decoder\\_gs.h](#).

## 10.204.2 Member Function Documentation

### 10.204.2.1 `general_work()`

```

int gps_l1_ca_telemetry_decoder_gs::general_work (
    int noutput_items,
    gr_vector_int & ninput_items,
    gr_vector_const_void_star & input_items,
    gr_vector_void_star & output_items) [override]
  
```

This is where all signal processing takes place.

#### 10.204.2.2 `set_channel()`

```
void gps_l1_ca_telemetry_decoder_gs::set_channel (
    int channel)
```

Set receiver's channel.

#### 10.204.2.3 `set_satellite()`

```
void gps_l1_ca_telemetry_decoder_gs::set_satellite (
    const Gnss_Satellite & satellite)
```

Set satellite PRN.

The documentation for this class was generated from the following file:

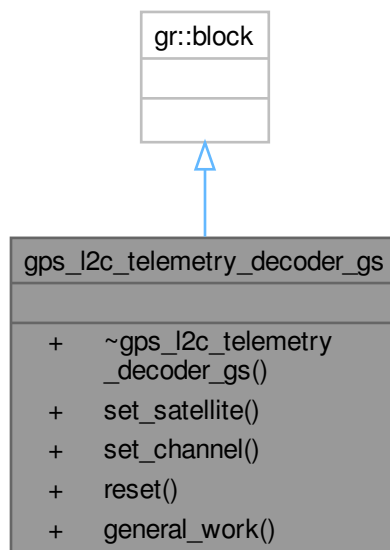
- [gps\\_l1\\_ca\\_telemetry\\_decoder\\_gs.h](#)

## 10.205 `gps_l2c_telemetry_decoder_gs` Class Reference

This class implements a block that decodes CNAV data defined in IS-GPS-200M.

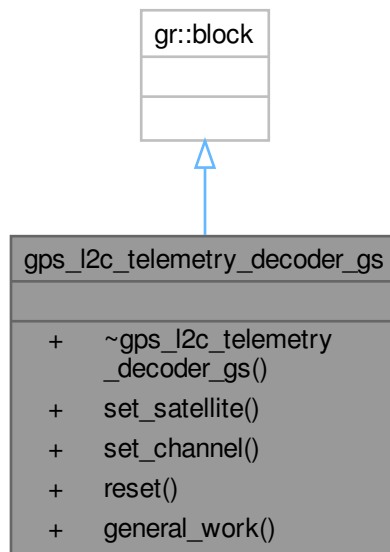
```
#include <gps_l2c_telemetry_decoder_gs.h>
```

Inheritance diagram for `gps_l2c_telemetry_decoder_gs`:





Collaboration diagram for `gps_l2c_telemetry_decoder_gs`:



### Public Member Functions

- void `set_satellite` (const [Gnss\\_Satellite](#) &satellite)  
*Set satellite PRN.*
- void `set_channel` (int32\_t channel)  
*Set receiver's channel.*
- void `reset` ()
- int `general_work` (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items) override  
*This is where all signal processing takes place.*

### Friends

- `gps_l2c_telemetry_decoder_gs_sptr` `gps_l2c_make_telemetry_decoder_gs` (const [Gnss\\_Satellite](#) &satellite, const [Tim\\_Conf](#) &conf)

## 10.205.1 Detailed Description

This class implements a block that decodes CNAV data defined in IS-GPS-200M.  
Definition at line 56 of file [gps\\_l2c\\_telemetry\\_decoder\\_gs.h](#).

## 10.205.2 Member Function Documentation

### 10.205.2.1 general\_work()

```

int gps_l2c_telemetry_decoder_gs::general_work (
    int noutput_items,
    gr_vector_int & ninput_items,
    gr_vector_const_void_star & input_items,
    gr_vector_void_star & output_items) [override]
  
```

This is where all signal processing takes place.

### 10.205.2.2 set\_channel()

```
void gps_l2c_telemetry_decoder_gs::set_channel (
    int32_t channel)
```

Set receiver's channel.

### 10.205.2.3 set\_satellite()

```
void gps_l2c_telemetry_decoder_gs::set_satellite (
    const Gnss_Satellite & satellite)
```

Set satellite PRN.

The documentation for this class was generated from the following file:

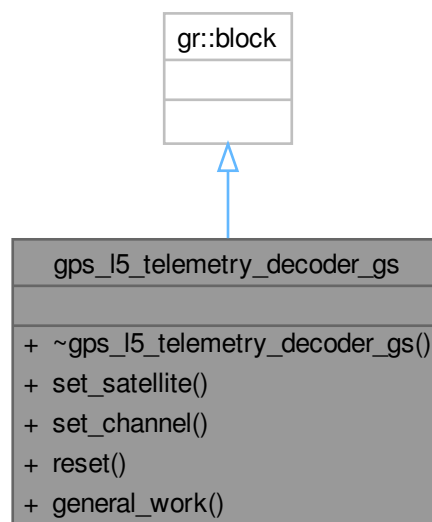
- [gps\\_l2c\\_telemetry\\_decoder\\_gs.h](#)

## 10.206 gps\_l5\_telemetry\_decoder\_gs Class Reference

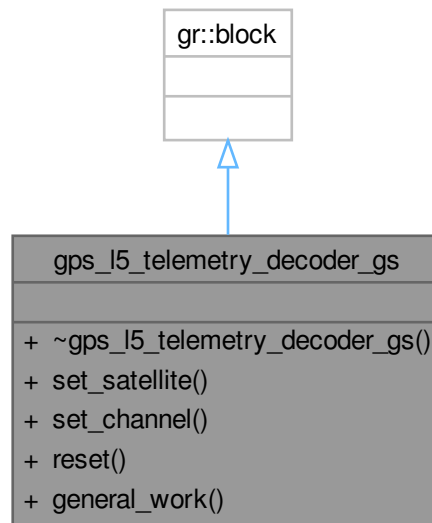
This class implements a GPS L5 Telemetry decoder.

```
#include <gps_l5_telemetry_decoder_gs.h>
```

Inheritance diagram for gps\_l5\_telemetry\_decoder\_gs:



Collaboration diagram for `gps_l5_telemetry_decoder_gs`:



### Public Member Functions

- void `set_satellite` (const [Gnss\\_Satellite](#) &satellite)  
*Set satellite PRN.*
- void `set_channel` (int32\_t channel)  
*Set receiver's channel.*
- void `reset` ()
- int `general_work` (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items) override

### Friends

- `gps_l5_telemetry_decoder_gs_sptr` `gps_l5_make_telemetry_decoder_gs` (const [Gnss\\_Satellite](#) &satellite, const [Tlm\\_Conf](#) &conf)

## 10.206.1 Detailed Description

This class implements a GPS L5 Telemetry decoder.  
Definition at line 58 of file [gps\\_l5\\_telemetry\\_decoder\\_gs.h](#).

## 10.206.2 Member Function Documentation

### 10.206.2.1 set\_channel()

```
void gps_l5_telemetry_decoder_gs::set_channel (
    int32_t channel)
```

Set receiver's channel.

### 10.206.2.2 set\_satellite()

```
void gps_l5_telemetry_decoder_gs::set_satellite (
    const Gnss\_Satellite & satellite)
```

Set satellite PRN.

The documentation for this class was generated from the following file:

- [gps\\_l5\\_telemetry\\_decoder\\_gs.h](#)

## 10.207 Gps\_Navigation\_Message Class Reference

This class decodes a GPS NAV Data message as described in IS-GPS-200M.

```
#include <gps_navigation_message.h>
```

Collaboration diagram for Gps\_Navigation\_Message:

Gps_Navigation_Message
<ul style="list-style-type: none"> <li>+ Gps_Navigation_Message()</li> <li>+ get_ephemeris()</li> <li>+ get_iono()</li> <li>+ get_almanac()</li> <li>+ get_utc_model()</li> <li>+ subframe_decoder()</li> <li>+ utc_time()</li> <li>+ get_TOW()</li> <li>+ get_GPS_week()</li> <li>+ set_satellite_PRN()</li> <li>and 6 more...</li> </ul>

### Public Member Functions

- [Gps\\_Navigation\\_Message](#) ()
- [Gps\\_Ephemeris get\\_ephemeris](#) () const  
*Obtain a GPS SV Ephemeris class filled with current SV data.*
- [Gps\\_Iono get\\_iono](#) ()  
*Obtain a GPS ionospheric correction parameters class filled with current SV data.*
- [Gps\\_Almanac get\\_almanac](#) ()  
*Obtain a GPS almanac class filled with current SV data.*
- [Gps\\_Utc\\_Model get\\_utc\\_model](#) ()  
*Obtain a GPS Almanac model parameters class filled with current SV data.*
- [int32\\_t subframe\\_decoder](#) (const char \*subframe)  
*Decodes the GPS NAV message.*
- [double utc\\_time](#) (double gpstime\_corrected) const  
*Computes the Coordinated Universal Time (UTC) and returns it in [s] (IS-GPS-200M, 20.3.3.5.2.4)*
- [int32\\_t get\\_TOW](#) () const  
*Gets Time of Week, in seconds.*
- [int32\\_t get\\_GPS\\_week](#) () const  
*Sets Time of Week, in seconds.*

- void [set\\_satellite\\_PRN](#) (uint32\_t prn)  
*Sets satellite PRN number.*
- uint32\_t [get\\_satellite\\_PRN](#) () const  
*Gets satellite PRN number.*
- void [set\\_channel](#) (int32\_t channel\_id)  
*Sets channel ID.*
- bool [get\\_flag\\_iono\\_valid](#) () const  
*Gets flag\_iono\_valid.*
- bool [get\\_flag\\_utc\\_model\\_valid](#) () const  
*Gets flag\_utc\_model\_valid.*
- bool [satellite\\_validation](#) ()
- bool [almanac\\_validation](#) () const

### 10.207.1 Detailed Description

This class decodes a GPS NAV Data message as described in IS-GPS-200M.

See <https://www.gps.gov/technical/icwg/IS-GPS-200M.pdf> Appendix II  
Definition at line 46 of file [gps\\_navigation\\_message.h](#).

### 10.207.2 Constructor & Destructor Documentation

#### 10.207.2.1 Gps\_Navigation\_Message()

`Gps_Navigation_Message::Gps_Navigation_Message ()`  
Default constructor

### 10.207.3 Member Function Documentation

#### 10.207.3.1 get\_almanac()

`Gps_Almanac Gps_Navigation_Message::get_almanac ()`  
Obtain a GPS almanac class filled with current SV data.

#### 10.207.3.2 get\_ephemeris()

`Gps_Ephemeris Gps_Navigation_Message::get_ephemeris () const`  
Obtain a GPS SV Ephemeris class filled with current SV data.

#### 10.207.3.3 get\_flag\_iono\_valid()

`bool Gps_Navigation_Message::get_flag_iono_valid () const [inline]`  
Gets flag\_iono\_valid.  
Definition at line 128 of file [gps\\_navigation\\_message.h](#).

#### 10.207.3.4 get\_flag\_utc\_model\_valid()

`bool Gps_Navigation_Message::get_flag_utc_model_valid () const [inline]`  
Gets flag\_utc\_model\_valid.  
Definition at line 136 of file [gps\\_navigation\\_message.h](#).

#### 10.207.3.5 get\_GPS\_week()

`int32_t Gps_Navigation_Message::get_GPS_week () const [inline]`  
Sets Time of Week, in seconds.  
Definition at line 96 of file [gps\\_navigation\\_message.h](#).

**10.207.3.6 get\_iono()**

[Gps\\_Iono](#) `Gps_Navigation_Message::get_iono ()`

Obtain a GPS ionospheric correction parameters class filled with current SV data.

**10.207.3.7 get\_satellite\_PRN()**

`uint32_t Gps_Navigation_Message::get_satellite_PRN () const [inline]`

Gets satellite PRN number.

Definition at line 112 of file [gps\\_navigation\\_message.h](#).

**10.207.3.8 get\_TOW()**

`int32_t Gps_Navigation_Message::get_TOW () const [inline]`

Gets Time of Week, in seconds.

Definition at line 88 of file [gps\\_navigation\\_message.h](#).

**10.207.3.9 get\_utc\_model()**

[Gps\\_Utc\\_Model](#) `Gps_Navigation_Message::get_utc_model ()`

Obtain a GPS Almanac model parameters class filled with current SV data.

**10.207.3.10 set\_channel()**

`void Gps_Navigation_Message::set_channel (`  
`int32_t channel_id) [inline]`

Sets channel ID.

Definition at line 120 of file [gps\\_navigation\\_message.h](#).

**10.207.3.11 set\_satellite\_PRN()**

`void Gps_Navigation_Message::set_satellite_PRN (`  
`uint32_t prn) [inline]`

Sets satellite PRN number.

Definition at line 104 of file [gps\\_navigation\\_message.h](#).

**10.207.3.12 subframe\_decoder()**

`int32_t Gps_Navigation_Message::subframe_decoder (`  
`const char * subframe)`

Decodes the GPS NAV message.

**10.207.3.13 utc\_time()**

`double Gps_Navigation_Message::utc_time (`  
`double gpstime_corrected) const`

Computes the Coordinated Universal Time (UTC) and returns it in [s] (IS-GPS-200M, 20.3.3.5.2.4)

The documentation for this class was generated from the following file:

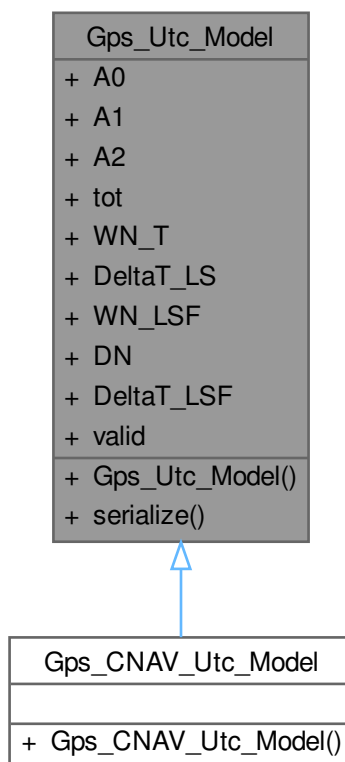
- [gps\\_navigation\\_message.h](#)

**10.208 Gps\_Utc\_Model Class Reference**

This class is a storage for the GPS UTC MODEL data as described in IS-GPS-200M.

`#include <gps_utc_model.h>`

Inheritance diagram for Gps\_Utc\_Model:



Collaboration diagram for Gps\_Utc\_Model:

Gps_Utc_Model
+ A0
+ A1
+ A2
+ tot
+ WN_T
+ DeltaT_LS
+ WN_LSF
+ DN
+ DeltaT_LSF
+ valid
+ Gps_Utc_Model()
+ serialize()

### Public Member Functions

- [Gps\\_Utc\\_Model](#) ()=default
- template<class Archive>  
void [serialize](#) (Archive &archive, const uint32\_t version)

### Public Attributes

- double [A0](#) {}  
*Constant of a model that relates GPS and UTC time (ref. 20.3.3.5.2.4 IS-GPS-200M) [s].*
- double [A1](#) {}  
*1st order term of a model that relates GPS and UTC time (ref. 20.3.3.5.2.4 IS-GPS-200M) [s/s]*
- double [A2](#) {}  
*2nd order term of a model that relates GPS and UTC time (ref. 20.3.3.5.2.4 IS-GPS-200M) [s/s]*
- int32\_t [tot](#) {}  
*Reference time for UTC data (reference 20.3.4.5 and 20.3.3.5.2.4 IS-GPS-200M) [s].*
- int32\_t [WN\\_T](#) {}  
*UTC reference week number [weeks].*
- int32\_t [DeltaT\\_LS](#) {}  
*Delta time due to leap seconds [s]. Number of leap seconds since 6-Jan-1980 as transmitted by the GPS almanac.*
- int32\_t [WN\\_LSF](#) {}  
*Week number at the end of which the leap second becomes effective [weeks].*
- int32\_t [DN](#) {}  
*Day number (DN) at the end of which the leap second becomes effective [days].*
- int32\_t [DeltaT\\_LSF](#) {}  
*Scheduled future or recent past (relative to NAV message upload) value of the delta time due to leap seconds [s].*
- bool [valid](#) {}



### 10.208.1 Detailed Description

This class is a storage for the GPS UTC MODEL data as described in IS-GPS-200M.

See <https://www.gps.gov/technical/icwg/IS-GPS-200M.pdf> Appendix II

Definition at line 35 of file [gps\\_utc\\_model.h](#).

### 10.208.2 Constructor & Destructor Documentation

#### 10.208.2.1 Gps\_Utc\_Model()

```
Gps_Utc_Model::Gps_Utc_Model () [default]
```

Default constructor

### 10.208.3 Member Function Documentation

#### 10.208.3.1 serialize()

```
template<class Archive>
void Gps_Utc_Model::serialize (
    Archive & archive,
    const uint32_t version) [inline]
```

Definition at line 61 of file [gps\\_utc\\_model.h](#).

### 10.208.4 Member Data Documentation

#### 10.208.4.1 A0

```
double Gps_Utc_Model::A0 {}
```

Constant of a model that relates GPS and UTC time (ref. 20.3.3.5.2.4 IS-GPS-200M) [s].

Definition at line 44 of file [gps\\_utc\\_model.h](#).

#### 10.208.4.2 A1

```
double Gps_Utc_Model::A1 {}
```

1st order term of a model that relates GPS and UTC time (ref. 20.3.3.5.2.4 IS-GPS-200M) [s/s]

Definition at line 45 of file [gps\\_utc\\_model.h](#).

#### 10.208.4.3 A2

```
double Gps_Utc_Model::A2 {}
```

2nd order term of a model that relates GPS and UTC time (ref. 20.3.3.5.2.4 IS-GPS-200M) [s/s]

Definition at line 46 of file [gps\\_utc\\_model.h](#).

#### 10.208.4.4 DeltaT\_LS

```
int32_t Gps_Utc_Model::DeltaT_LS {}
```

Delta time due to leap seconds [s]. Number of leap seconds since 6-Jan-1980 as transmitted by the GPS almanac.

Definition at line 49 of file [gps\\_utc\\_model.h](#).

#### 10.208.4.5 DeltaT\_LSF

```
int32_t Gps_Utc_Model::DeltaT_LSF {}
```

Scheduled future or recent past (relative to NAV message upload) value of the delta time due to leap seconds [s].

Definition at line 52 of file [gps\\_utc\\_model.h](#).

#### 10.208.4.6 DN

```
int32_t Gps_Utc_Model::DN {}
```

Day number (DN) at the end of which the leap second becomes effective [days].

Definition at line 51 of file [gps\\_utc\\_model.h](#).

#### 10.208.4.7 tot

```
int32_t Gps_Utc_Model::tot {}
```

Reference time for UTC data (reference 20.3.4.5 and 20.3.3.5.2.4 IS-GPS-200M) [s].

Definition at line 47 of file [gps\\_utc\\_model.h](#).

#### 10.208.4.8 valid

```
bool Gps_Utc_Model::valid {}
```

Definition at line 54 of file [gps\\_utc\\_model.h](#).

#### 10.208.4.9 WN\_LSF

```
int32_t Gps_Utc_Model::WN_LSF {}
```

Week number at the end of which the leap second becomes effective [weeks].

Definition at line 50 of file [gps\\_utc\\_model.h](#).

#### 10.208.4.10 WN\_T

```
int32_t Gps_Utc_Model::WN_T {}
```

UTC reference week number [weeks].

Definition at line 48 of file [gps\\_utc\\_model.h](#).

The documentation for this class was generated from the following file:

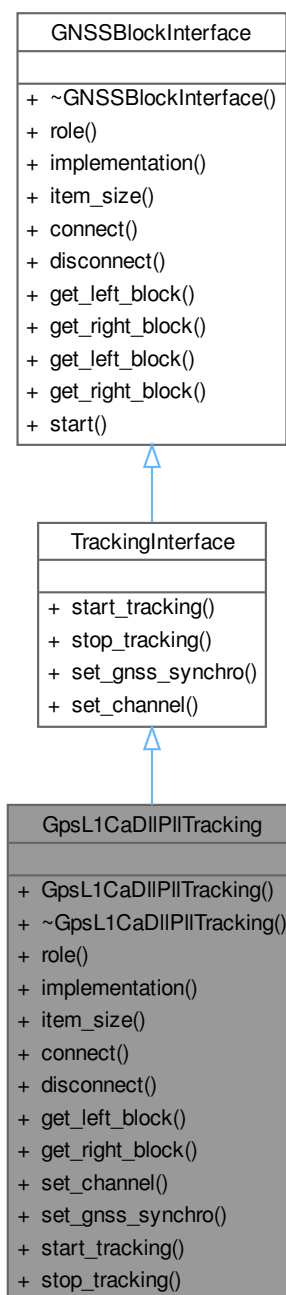
- [gps\\_utc\\_model.h](#)

## 10.209 GpsL1CaDIIPITracking Class Reference

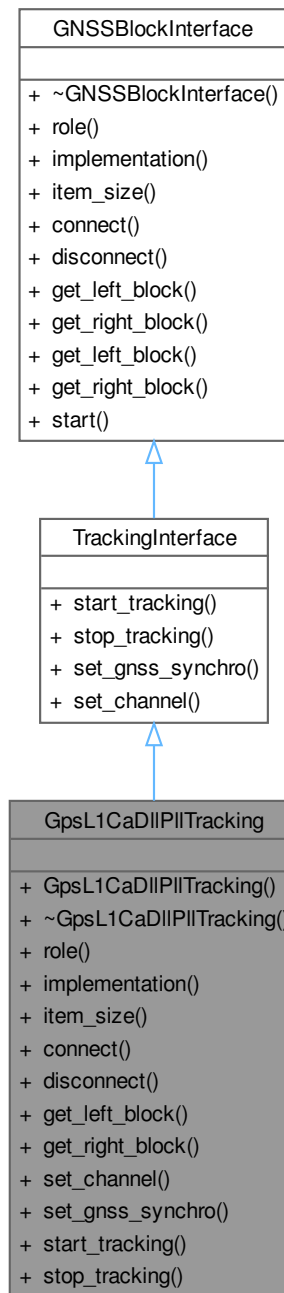
This class implements a code DLL + carrier PLL tracking loop.

```
#include <gps_l1_ca_dll_pll_tracking.h>
```

Inheritance diagram for GpsL1CaDIIPIITracking:



Collaboration diagram for GpsL1CaDIIPIITracking:



## Public Member Functions

- **GpsL1CaDIIPIITracking** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override
  - Returns "GPS\_L1\_CA\_DLL\_PLL\_Tracking".
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_channel](#) (unsigned int channel) override  
*Set tracking channel unique ID.*
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [start\\_tracking](#) () override
- void [stop\\_tracking](#) () override  
*Stop running tracking.*

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()  
*Start the flow of samples if needed.*

### 10.209.1 Detailed Description

This class implements a code DLL + carrier PLL tracking loop.  
Definition at line 44 of file [gps\\_l1\\_ca\\_dll\\_pll\\_tracking.h](#).

### 10.209.2 Member Function Documentation

#### 10.209.2.1 connect()

```
void GpsL1CaDllPllTracking::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.209.2.2 disconnect()

```
void GpsL1CaDllPllTracking::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.209.2.3 get\_left\_block()

```
gr::basic_block_sptr GpsL1CaDllPllTracking::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.209.2.4 get\_right\_block()

```
gr::basic_block_sptr GpsL1CaDllPllTracking::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.209.2.5 implementation()

```
std::string GpsL1CaDllPllTracking::implementation () [inline], [override], [virtual]
```

Returns "GPS\_L1\_CA\_DLL\_PLL\_Tracking".  
Implements [GNSSBlockInterface](#).  
Definition at line 61 of file [gps\\_l1\\_ca\\_dll\\_pll\\_tracking.h](#).

**10.209.2.6 item\_size()**

```
size_t GpsL1CaDllPllTracking::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 66 of file [gps\\_l1\\_ca\\_dll\\_pll\\_tracking.h](#).

**10.209.2.7 role()**

```
std::string GpsL1CaDllPllTracking::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 55 of file [gps\\_l1\\_ca\\_dll\\_pll\\_tracking.h](#).

**10.209.2.8 set\_channel()**

```
void GpsL1CaDllPllTracking::set_channel (
    unsigned int channel) [override], [virtual]
```

Set tracking channel unique ID.

Implements [TrackingInterface](#).

**10.209.2.9 set\_gnss\_synchro()**

```
void GpsL1CaDllPllTracking::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [TrackingInterface](#).

**10.209.2.10 start\_tracking()**

```
void GpsL1CaDllPllTracking::start_tracking () [override], [virtual]
```

Implements [TrackingInterface](#).

**10.209.2.11 stop\_tracking()**

```
void GpsL1CaDllPllTracking::stop_tracking () [override], [virtual]
```

Stop running tracking.

Implements [TrackingInterface](#).

The documentation for this class was generated from the following file:

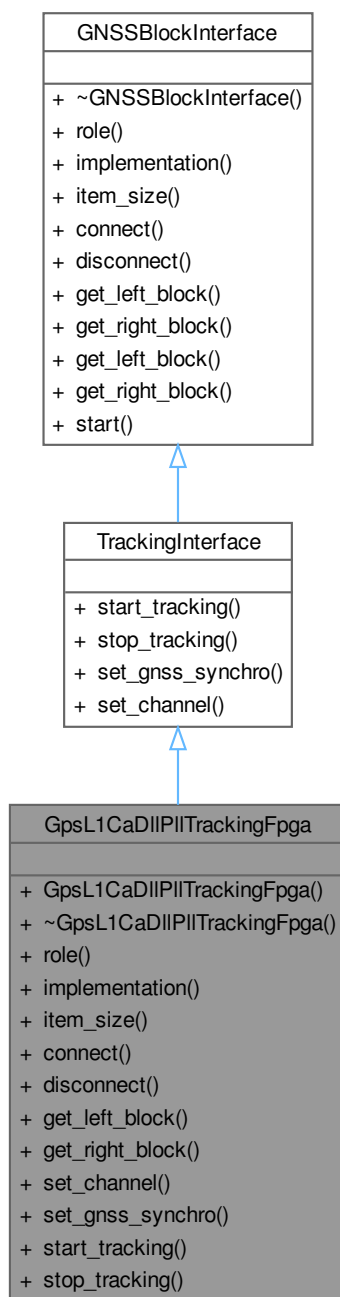
- [gps\\_l1\\_ca\\_dll\\_pll\\_tracking.h](#)

**10.210 GpsL1CaDllPllTrackingFpga Class Reference**

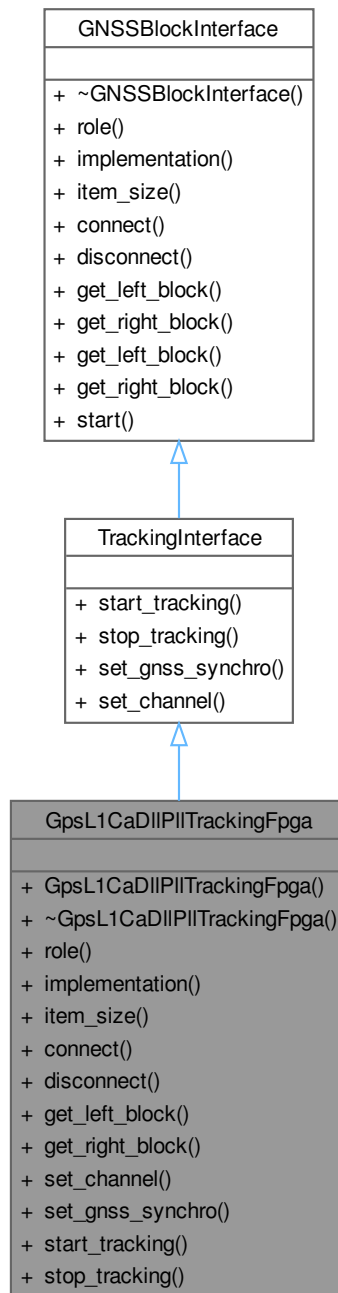
This class implements a code DLL + carrier PLL tracking loop.

```
#include <gps_l1_ca_dll_pll_tracking_fpga.h>
```

Inheritance diagram for GpsL1CaDIIPIITrackingFpga:



Collaboration diagram for GpsL1CaDIIPIITrackingFpga:



## Public Member Functions

- [GpsL1CaDIIPIITrackingFpga](#) (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)  
*Constructor.*
- virtual [~GpsL1CaDIIPIITrackingFpga](#) ()  
*Destructor.*
- std::string [role](#) () override



- Role.*
- `std::string implementation ()` override  
*Returns "GPS\_L1\_CA\_DLL\_PLL\_Tracking\_FPGA".*
- `size_t item_size ()` override  
*Returns size of `lv_16sc_t`.*
- `void connect (gr::top_block_sptr top_block)` override  
*Connect.*
- `void disconnect (gr::top_block_sptr top_block)` override  
*Disconnect.*
- `gr::basic_block_sptr get_left_block ()` override  
*Get left block.*
- `gr::basic_block_sptr get_right_block ()` override  
*Get right block.*
- `void set_channel (unsigned int channel)` override  
*Set tracking channel unique ID.*
- `void set_gnss_synchro (Gnss_Synchro *p_gnss_synchro)` override  
*Set acquisition/tracking common `Gnss_Synchro` object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- `void start_tracking ()` override  
*Start the tracking process in the FPGA.*
- `void stop_tracking ()` override  
*Stop the tracking process in the FPGA.*

### Public Member Functions inherited from GNSSBlockInterface

- `virtual gr::basic_block_sptr get_left_block (int RF_channel)`
- `virtual gr::basic_block_sptr get_right_block (int RF_channel)`
- `virtual void start ()`  
*Start the flow of samples if needed.*

## 10.210.1 Detailed Description

This class implements a code DLL + carrier PLL tracking loop.

Definition at line 41 of file `gps_l1_ca_dll_pll_tracking_fpga.h`.

## 10.210.2 Constructor & Destructor Documentation

### 10.210.2.1 GpsL1CaDllPllTrackingFpga()

```
GpsL1CaDllPllTrackingFpga::GpsL1CaDllPllTrackingFpga (
    const ConfigurationInterface * configuration,
    const std::string & role,
    unsigned int in_streams,
    unsigned int out_streams)
```

Constructor.

References `role()`.

Here is the call graph for this function:



### 10.210.2.2 ~GpsL1CaDllPllTrackingFpga()

virtual GpsL1CaDllPllTrackingFpga::~~GpsL1CaDllPllTrackingFpga () [virtual]  
Destructor.

## 10.210.3 Member Function Documentation

### 10.210.3.1 connect()

void GpsL1CaDllPllTrackingFpga::connect (   
 gr::top\_block\_sptr top\_block) [override], [virtual]

Connect.

Implements [GNSSBlockInterface](#).

### 10.210.3.2 disconnect()

void GpsL1CaDllPllTrackingFpga::disconnect (   
 gr::top\_block\_sptr top\_block) [override], [virtual]

Disconnect.

Implements [GNSSBlockInterface](#).

### 10.210.3.3 get\_left\_block()

gr::basic\_block\_sptr GpsL1CaDllPllTrackingFpga::get\_left\_block () [override], [virtual]

Get left block.

Implements [GNSSBlockInterface](#).

### 10.210.3.4 get\_right\_block()

gr::basic\_block\_sptr GpsL1CaDllPllTrackingFpga::get\_right\_block () [override], [virtual]

Get right block.

Implements [GNSSBlockInterface](#).

### 10.210.3.5 implementation()

std::string GpsL1CaDllPllTrackingFpga::implementation () [inline], [override], [virtual]

Returns "GPS\_L1\_CA\_DLL\_PLL\_Tracking\_FPGA".

Implements [GNSSBlockInterface](#).

Definition at line 69 of file [gps\\_l1\\_ca\\_dll\\_pll\\_tracking\\_fpga.h](#).

### 10.210.3.6 item\_size()

size\_t GpsL1CaDllPllTrackingFpga::item\_size () [inline], [override], [virtual]

Returns size of lv\_16sc\_t.

Implements [GNSSBlockInterface](#).

Definition at line 77 of file [gps\\_l1\\_ca\\_dll\\_pll\\_tracking\\_fpga.h](#).

### 10.210.3.7 role()

std::string GpsL1CaDllPllTrackingFpga::role () [inline], [override], [virtual]

Role.

Implements [GNSSBlockInterface](#).

Definition at line 61 of file [gps\\_l1\\_ca\\_dll\\_pll\\_tracking\\_fpga.h](#).

Referenced by [GpsL1CaDllPllTrackingFpga\(\)](#).

### 10.210.3.8 set\_channel()

void GpsL1CaDllPllTrackingFpga::set\_channel (   
 unsigned int channel) [override], [virtual]

Set tracking channel unique ID.

Implements [TrackingInterface](#).

### 10.210.3.9 set\_gnss\_synchro()

```
void GpsL1CaDllPllTrackingFpga::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [TrackingInterface](#).

### 10.210.3.10 start\_tracking()

```
void GpsL1CaDllPllTrackingFpga::start_tracking () [override], [virtual]
```

Start the tracking process in the FPGA.

Implements [TrackingInterface](#).

### 10.210.3.11 stop\_tracking()

```
void GpsL1CaDllPllTrackingFpga::stop_tracking () [override], [virtual]
```

Stop the tracking process in the FPGA.

Implements [TrackingInterface](#).

The documentation for this class was generated from the following file:

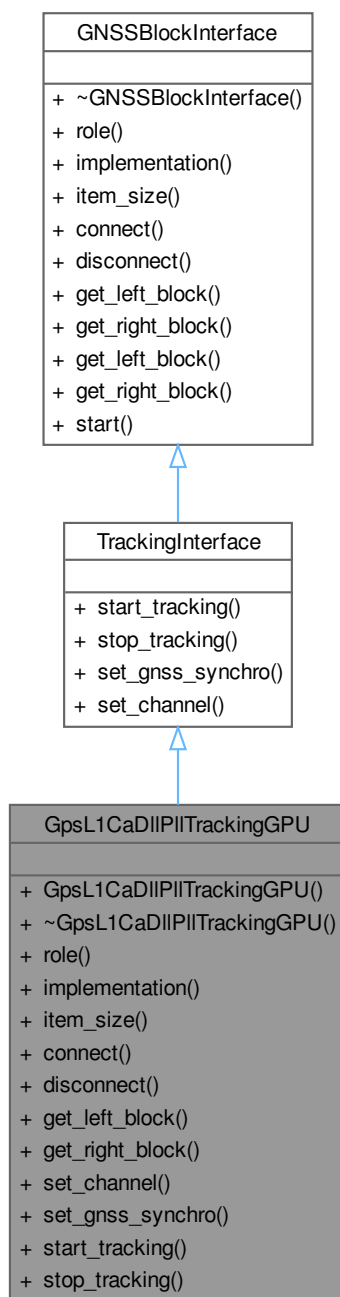
- [gps\\_l1\\_ca\\_dll\\_pll\\_tracking\\_fpga.h](#)

## 10.211 GpsL1CaDllPllTrackingGPU Class Reference

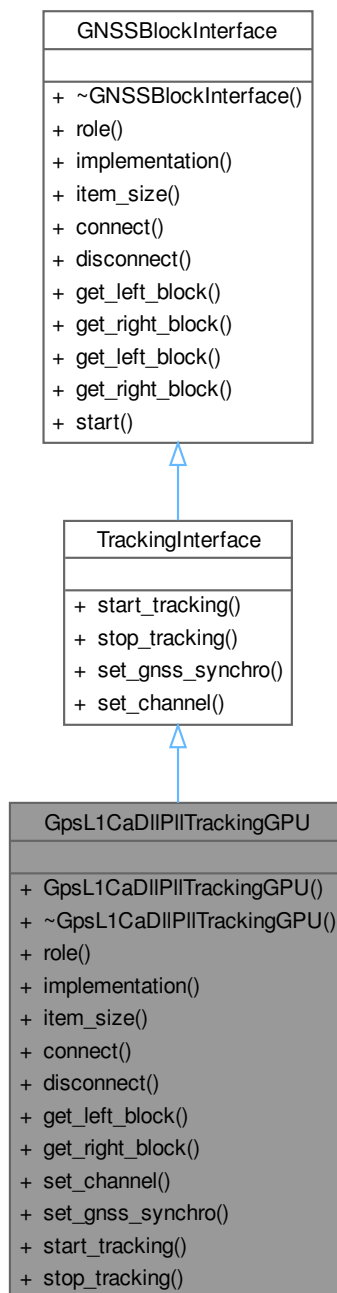
This class implements a code DLL + carrier PLL tracking loop using GPU accelerated functions.

```
#include <gps_l1_ca_dll_pll_tracking_gpu.h>
```

Inheritance diagram for GpsL1CaDIIPIITrackingGPU:



Collaboration diagram for GpsL1CaDIIPIITrackingGPU:



## Public Member Functions

- **GpsL1CaDIIPIITrackingGPU** (const [ConfigurationInterface](#) \*configuration, std::string role, unsigned int in↔\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override
  - Returns "GPS\_L1\_CA\_DLL\_PLL\_Tracking\_GPU".
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_channel](#) (unsigned int channel) override  
*Set tracking channel unique ID.*
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [start\\_tracking](#) () override
- void [stop\\_tracking](#) () override  
*Stop running tracking.*

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()  
*Start the flow of samples if needed.*

### 10.211.1 Detailed Description

This class implements a code DLL + carrier PLL tracking loop using GPU accelerated functions.  
Definition at line 41 of file [gps\\_l1\\_ca\\_dll\\_pll\\_tracking\\_gpu.h](#).

### 10.211.2 Member Function Documentation

#### 10.211.2.1 connect()

```
void GpsL1CaDllPllTrackingGPU::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.211.2.2 disconnect()

```
void GpsL1CaDllPllTrackingGPU::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.211.2.3 get\_left\_block()

```
gr::basic_block_sptr GpsL1CaDllPllTrackingGPU::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.211.2.4 get\_right\_block()

```
gr::basic_block_sptr GpsL1CaDllPllTrackingGPU::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.211.2.5 implementation()

```
std::string GpsL1CaDllPllTrackingGPU::implementation () [inline], [override], [virtual]
```

Returns "GPS\_L1\_CA\_DLL\_PLL\_Tracking\_GPU".  
Implements [GNSSBlockInterface](#).  
Definition at line 58 of file [gps\\_l1\\_ca\\_dll\\_pll\\_tracking\\_gpu.h](#).

**10.211.2.6 item\_size()**

```
size_t GpsL1CaDllPllTrackingGPU::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 63 of file [gps\\_l1\\_ca\\_dll\\_pll\\_tracking\\_gpu.h](#).

**10.211.2.7 role()**

```
std::string GpsL1CaDllPllTrackingGPU::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 52 of file [gps\\_l1\\_ca\\_dll\\_pll\\_tracking\\_gpu.h](#).

**10.211.2.8 set\_channel()**

```
void GpsL1CaDllPllTrackingGPU::set_channel (
    unsigned int channel) [override], [virtual]
```

Set tracking channel unique ID.

Implements [TrackingInterface](#).

**10.211.2.9 set\_gnss\_synchro()**

```
void GpsL1CaDllPllTrackingGPU::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [TrackingInterface](#).

**10.211.2.10 start\_tracking()**

```
void GpsL1CaDllPllTrackingGPU::start_tracking () [override], [virtual]
```

Implements [TrackingInterface](#).

**10.211.2.11 stop\_tracking()**

```
void GpsL1CaDllPllTrackingGPU::stop_tracking () [override], [virtual]
```

Stop running tracking.

Implements [TrackingInterface](#).

The documentation for this class was generated from the following file:

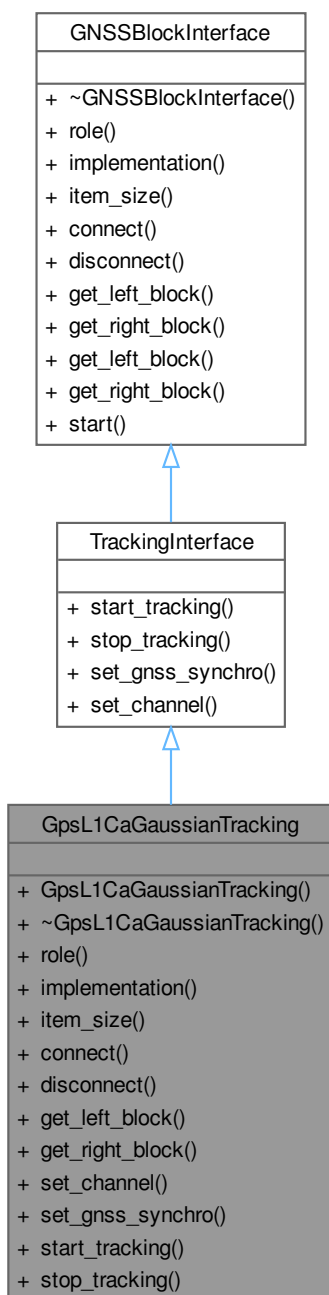
- [gps\\_l1\\_ca\\_dll\\_pll\\_tracking\\_gpu.h](#)

**10.212 GpsL1CaGaussianTracking Class Reference**

This class implements a code DLL + carrier PLL tracking loop.

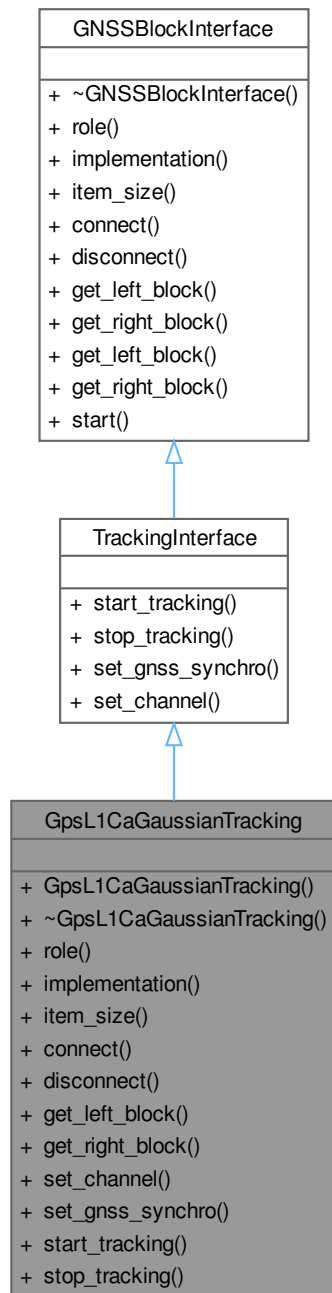
```
#include <gps_l1_ca_gaussian_tracking.h>
```

Inheritance diagram for GpsL1CaGaussianTracking:





Collaboration diagram for GpsL1CaGaussianTracking:



### Public Member Functions

- **GpsL1CaGaussianTracking** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override
  - Returns "GPS\_L1\_CA\_Gaussian\_Tracking".*
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_channel](#) (unsigned int channel) override
 

*Set tracking channel unique ID.*
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override
 

*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [start\\_tracking](#) () override
- void [stop\\_tracking](#) () override
 

*Stop running tracking.*

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()
 

*Start the flow of samples if needed.*

### 10.212.1 Detailed Description

This class implements a code DLL + carrier PLL tracking loop.  
 Definition at line 44 of file [gps\\_l1\\_ca\\_gaussian\\_tracking.h](#).

### 10.212.2 Member Function Documentation

#### 10.212.2.1 connect()

```
void GpsL1CaGaussianTracking::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.212.2.2 disconnect()

```
void GpsL1CaGaussianTracking::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.212.2.3 get\_left\_block()

```
gr::basic_block_sptr GpsL1CaGaussianTracking::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.212.2.4 get\_right\_block()

```
gr::basic_block_sptr GpsL1CaGaussianTracking::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.212.2.5 implementation()

```
std::string GpsL1CaGaussianTracking::implementation () [inline], [override], [virtual]
```

Returns "GPS\_L1\_CA\_Gaussian\_Tracking".  
 Implements [GNSSBlockInterface](#).  
 Definition at line 61 of file [gps\\_l1\\_ca\\_gaussian\\_tracking.h](#).

#### 10.212.2.6 item\_size()

```
size_t GpsL1CaGaussianTracking::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 66 of file [gps\\_l1\\_ca\\_gaussian\\_tracking.h](#).

#### 10.212.2.7 role()

```
std::string GpsL1CaGaussianTracking::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 55 of file [gps\\_l1\\_ca\\_gaussian\\_tracking.h](#).

#### 10.212.2.8 set\_channel()

```
void GpsL1CaGaussianTracking::set_channel (
    unsigned int channel) [override], [virtual]
```

Set tracking channel unique ID.

Implements [TrackingInterface](#).

#### 10.212.2.9 set\_gnss\_synchro()

```
void GpsL1CaGaussianTracking::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [TrackingInterface](#).

#### 10.212.2.10 start\_tracking()

```
void GpsL1CaGaussianTracking::start_tracking () [override], [virtual]
```

Implements [TrackingInterface](#).

#### 10.212.2.11 stop\_tracking()

```
void GpsL1CaGaussianTracking::stop_tracking () [override], [virtual]
```

Stop running tracking.

Implements [TrackingInterface](#).

The documentation for this class was generated from the following file:

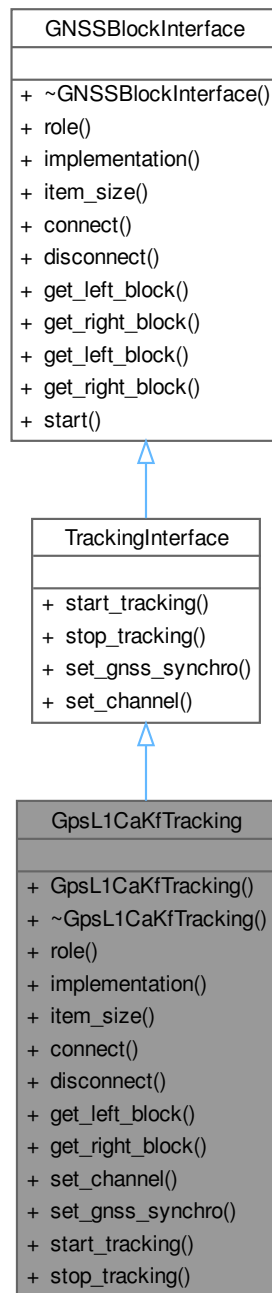
- [gps\\_l1\\_ca\\_gaussian\\_tracking.h](#)

## 10.213 GpsL1CaKfTracking Class Reference

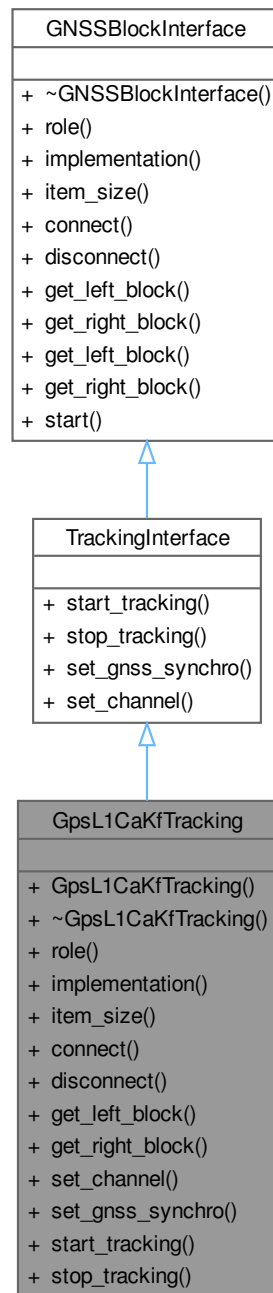
This class implements a code + carrier Kalman Filter tracking loop with VTL capabilities.

```
#include <gps_l1_ca_kf_tracking.h>
```

Inheritance diagram for GpsL1CaKfTracking:



Collaboration diagram for GpsL1CaKfTracking:



### Public Member Functions

- **GpsL1CaKfTracking** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "GPS\_L1\_CA\_KF\_Tracking".
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_channel](#) (unsigned int channel) override
 

*Set tracking channel unique ID.*
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override
 

*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [start\\_tracking](#) () override
- void [stop\\_tracking](#) () override
 

*Stop running tracking.*

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()
 

*Start the flow of samples if needed.*

### 10.213.1 Detailed Description

This class implements a code + carrier Kalman Filter tracking loop with VTL capabilities.  
Definition at line 36 of file [gps\\_l1\\_ca\\_kf\\_tracking.h](#).

### 10.213.2 Member Function Documentation

#### 10.213.2.1 connect()

```
void GpsL1CaKfTracking::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.213.2.2 disconnect()

```
void GpsL1CaKfTracking::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.213.2.3 get\_left\_block()

```
gr::basic_block_sptr GpsL1CaKfTracking::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.213.2.4 get\_right\_block()

```
gr::basic_block_sptr GpsL1CaKfTracking::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.213.2.5 implementation()

```
std::string GpsL1CaKfTracking::implementation () [inline], [override], [virtual]
```

Returns "GPS\_L1\_CA\_KF\_Tracking".  
Implements [GNSSBlockInterface](#).  
Definition at line 53 of file [gps\\_l1\\_ca\\_kf\\_tracking.h](#).

**10.213.2.6 item\_size()**

```
size_t GpsL1CaKfTracking::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 58 of file [gps\\_l1\\_ca\\_kf\\_tracking.h](#).

**10.213.2.7 role()**

```
std::string GpsL1CaKfTracking::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 47 of file [gps\\_l1\\_ca\\_kf\\_tracking.h](#).

**10.213.2.8 set\_channel()**

```
void GpsL1CaKfTracking::set_channel (
    unsigned int channel) [override], [virtual]
```

Set tracking channel unique ID.

Implements [TrackingInterface](#).

**10.213.2.9 set\_gnss\_synchro()**

```
void GpsL1CaKfTracking::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [TrackingInterface](#).

**10.213.2.10 start\_tracking()**

```
void GpsL1CaKfTracking::start_tracking () [override], [virtual]
```

Implements [TrackingInterface](#).

**10.213.2.11 stop\_tracking()**

```
void GpsL1CaKfTracking::stop_tracking () [override], [virtual]
```

Stop running tracking.

Implements [TrackingInterface](#).

The documentation for this class was generated from the following file:

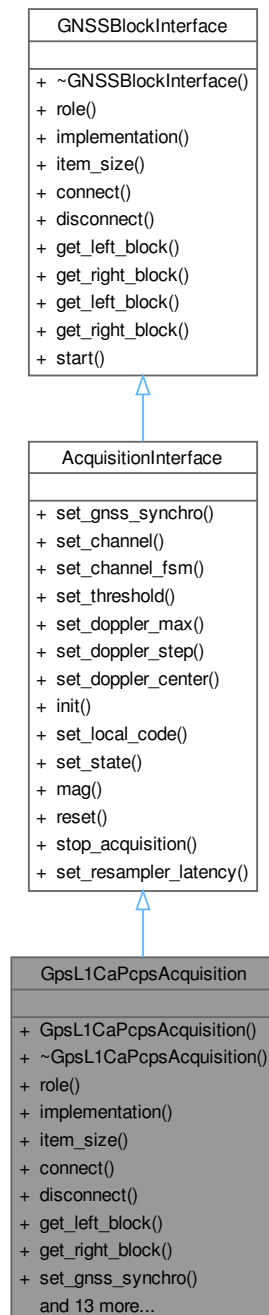
- [gps\\_l1\\_ca\\_kf\\_tracking.h](#)

**10.214 GpsL1CaPcpsAcquisition Class Reference**

This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.

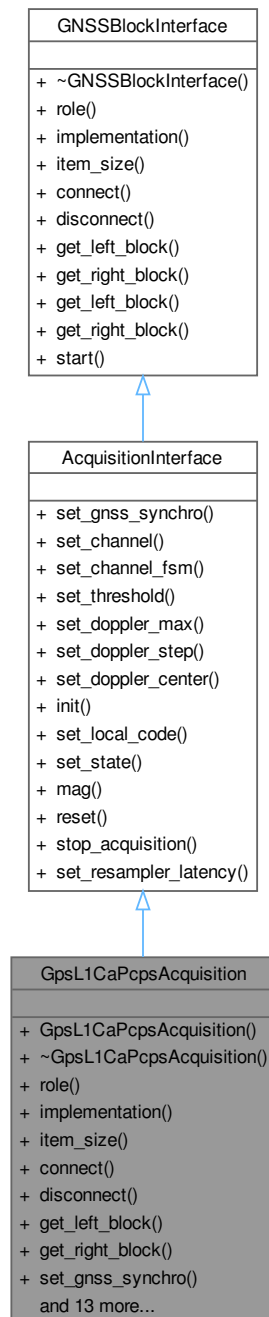
```
#include <gps_l1_ca_pcps_acquisition.h>
```

Inheritance diagram for GpsL1CaPcpsAcquisition:





Collaboration diagram for GpsL1CaPcpsAcquisition:



## Public Member Functions

- **GpsL1CaPcpsAcquisition** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override
  - Returns "GPS\_L1\_CA\_PCPS\_Acquisition".
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [set\\_channel](#) (unsigned int channel) override  
*Set acquisition channel unique ID.*
- void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm) override  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold) override  
*Set statistics threshold of PCPS algorithm.*
- void [set\\_doppler\\_max](#) (unsigned int doppler\_max) override  
*Set maximum Doppler off grid search.*
- void [set\\_doppler\\_step](#) (unsigned int doppler\_step) override  
*Set Doppler steps for the grid search.*
- void [set\\_doppler\\_center](#) (int doppler\_center) override  
*Set Doppler center for the grid search.*
- void [init](#) () override  
*Initializes acquisition algorithm.*
- void [set\\_local\\_code](#) () override  
*Sets local code for GPS L1/CA PCPS acquisition algorithm.*
- signed int [mag](#) () override  
*Returns the maximum peak of grid search.*
- void [reset](#) () override  
*Restart acquisition algorithm.*
- void [set\\_state](#) (int state) override  
*If state = 1, it forces the block to start acquiring from the first sample.*
- void [stop\\_acquisition](#) () override  
*Stop running acquisition.*
- void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples) override  
*Sets the resampler latency to account it in the acquisition code delay estimation.*

### Public Member Functions inherited from [AcquisitionInterface](#)

- virtual void [set\\_doppler\\_center](#) (int doppler\_center \_\_attribute\_\_((unused)))

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

## 10.214.1 Detailed Description

This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.  
Definition at line 50 of file [gps\\_l1\\_ca\\_pcps\\_acquisition.h](#).

## 10.214.2 Member Function Documentation

### 10.214.2.1 connect()

```
void GpsL1CaPcpsAcquisition::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.214.2.2 disconnect()

```
void GpsL1CaPcpsAcquisition::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.214.2.3 get\_left\_block()

```
gr::basic_block_sptr GpsL1CaPcpsAcquisition::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.214.2.4 get\_right\_block()

```
gr::basic_block_sptr GpsL1CaPcpsAcquisition::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.214.2.5 implementation()

```
std::string GpsL1CaPcpsAcquisition::implementation () [inline], [override], [virtual]
```

Returns "GPS\_L1\_CA\_PCPS\_Acquisition".

Implements [GNSSBlockInterface](#).

Definition at line 69 of file [gps\\_l1\\_ca\\_pcps\\_acquisition.h](#).

### 10.214.2.6 init()

```
void GpsL1CaPcpsAcquisition::init () [override], [virtual]
```

Initializes acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.214.2.7 item\_size()

```
size_t GpsL1CaPcpsAcquisition::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 74 of file [gps\\_l1\\_ca\\_pcps\\_acquisition.h](#).

### 10.214.2.8 mag()

```
signed int GpsL1CaPcpsAcquisition::mag () [override], [virtual]
```

Returns the maximum peak of grid search.

Implements [AcquisitionInterface](#).

### 10.214.2.9 reset()

```
void GpsL1CaPcpsAcquisition::reset () [override], [virtual]
```

Restart acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.214.2.10 role()

```
std::string GpsL1CaPcpsAcquisition::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 61 of file [gps\\_l1\\_ca\\_pcps\\_acquisition.h](#).

**10.214.2.11 set\_channel()**

```
void GpsL1CaPcpsAcquisition::set_channel (
    unsigned int channel) [inline], [override], [virtual]
```

Set acquisition channel unique ID.

Implements [AcquisitionInterface](#).

Definition at line 94 of file [gps\\_l1\\_ca\\_pcps\\_acquisition.h](#).

**10.214.2.12 set\_channel\_fsm()**

```
void GpsL1CaPcpsAcquisition::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline], [override], [virtual]
```

Set channel fsm associated to this acquisition instance.

Implements [AcquisitionInterface](#).

Definition at line 103 of file [gps\\_l1\\_ca\\_pcps\\_acquisition.h](#).

**10.214.2.13 set\_doppler\_center()**

```
void GpsL1CaPcpsAcquisition::set_doppler_center (
    int doppler_center) [override]
```

Set Doppler center for the grid search.

**10.214.2.14 set\_doppler\_max()**

```
void GpsL1CaPcpsAcquisition::set_doppler_max (
    unsigned int doppler_max) [override], [virtual]
```

Set maximum Doppler off grid search.

Implements [AcquisitionInterface](#).

**10.214.2.15 set\_doppler\_step()**

```
void GpsL1CaPcpsAcquisition::set_doppler_step (
    unsigned int doppler_step) [override], [virtual]
```

Set Doppler steps for the grid search.

Implements [AcquisitionInterface](#).

**10.214.2.16 set\_gnss\_synchro()**

```
void GpsL1CaPcpsAcquisition::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [AcquisitionInterface](#).

**10.214.2.17 set\_local\_code()**

```
void GpsL1CaPcpsAcquisition::set_local_code () [override], [virtual]
```

Sets local code for GPS L1/CA PCPS acquisition algorithm.

Implements [AcquisitionInterface](#).

**10.214.2.18 set\_resampler\_latency()**

```
void GpsL1CaPcpsAcquisition::set_resampler_latency (
    uint32_t latency_samples) [override], [virtual]
```

Sets the resampler latency to account it in the acquisition code delay estimation.

Implements [AcquisitionInterface](#).

**10.214.2.19 set\_state()**

```
void GpsL1CaPcpsAcquisition::set_state (
    int state) [override], [virtual]
```

If state = 1, it forces the block to start acquiring from the first sample.

Implements [AcquisitionInterface](#).

**10.214.2.20 set\_threshold()**

```
void GpsL1CaPcpsAcquisition::set_threshold (
    float threshold) [override], [virtual]
```

Set statistics threshold of PCPS algorithm.

Implements [AcquisitionInterface](#).

**10.214.2.21 stop\_acquisition()**

```
void GpsL1CaPcpsAcquisition::stop_acquisition () [override], [virtual]
```

Stop running acquisition.

Implements [AcquisitionInterface](#).

The documentation for this class was generated from the following file:

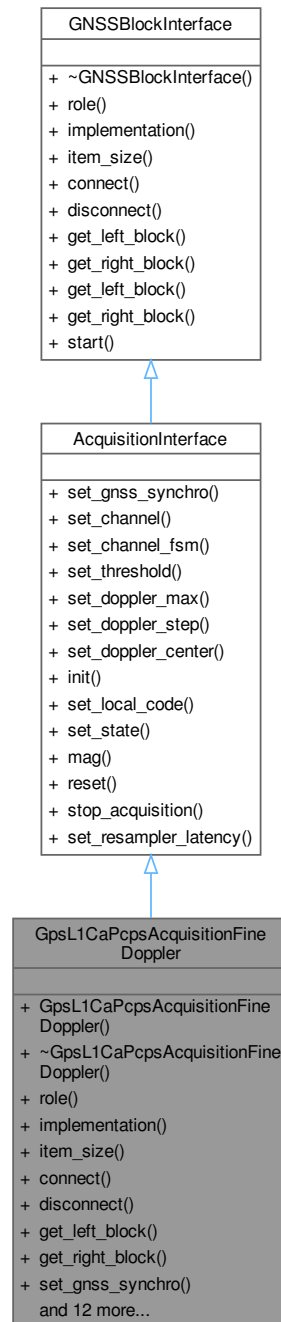
- [gps\\_l1\\_ca\\_pcps\\_acquisition.h](#)

**10.215 GpsL1CaPcpsAcquisitionFineDoppler Class Reference**

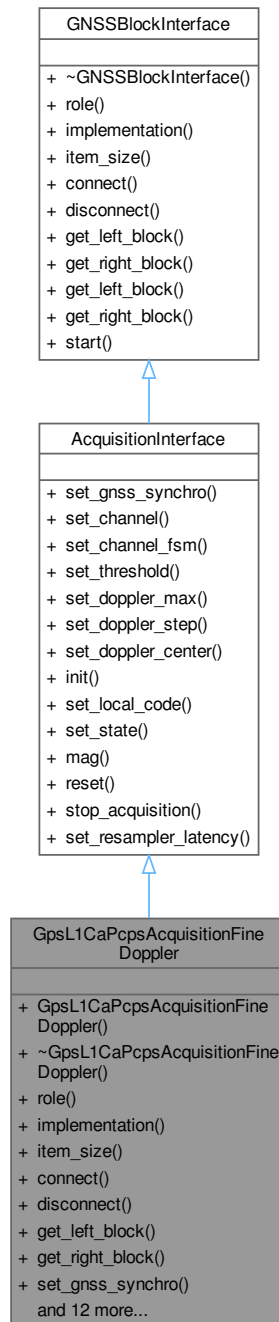
This class Adapts a PCPS acquisition block with fine Doppler estimation to an [AcquisitionInterface](#) for GPS L1 C/A signals.

```
#include <gps_l1_ca_pcps_acquisition_fine_doppler.h>
```

Inheritance diagram for GpsL1CaPcpsAcquisitionFineDoppler:



Collaboration diagram for GpsL1CaPcpsAcquisitionFineDoppler:



## Public Member Functions

- **GpsL1CaPcpsAcquisitionFineDoppler** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override
 

*Returns "GPS\_L1\_CA\_PCPS\_Acquisition\_Fine\_Doppler".*
- size\_t [item\\_size](#) () override

- void **connect** (gnss\_shared\_ptr< gr::top\_block > top\_block) override
- void **disconnect** (gnss\_shared\_ptr< gr::top\_block > top\_block) override
- gnss\_shared\_ptr< gr::basic\_block > **get\_left\_block** () override
- gnss\_shared\_ptr< gr::basic\_block > **get\_right\_block** () override
- void **set\_gnss\_synchro** (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common Gnss\_Synchro object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void **set\_channel** (unsigned int channel) override  
*Set acquisition channel unique ID.*
- void **set\_channel\_fsm** (std::weak\_ptr< ChannelFsm > channel\_fsm) override  
*Set channel fsm associated to this acquisition instance.*
- void **set\_threshold** (float threshold) override  
*Set statistics threshold of PCPS algorithm.*
- void **set\_doppler\_max** (unsigned int doppler\_max) override  
*Set maximum Doppler off grid search.*
- void **set\_doppler\_step** (unsigned int doppler\_step) override  
*Set Doppler steps for the grid search.*
- void **init** () override  
*Initializes acquisition algorithm.*
- void **set\_local\_code** () override
- signed int **mag** () override  
*Returns the maximum peak of grid search.*
- void **reset** () override  
*Restart acquisition algorithm.*
- void **set\_state** (int state) override  
*If state = 1, it forces the block to start acquiring from the first sample.*
- void **stop\_acquisition** () override  
*Stop running acquisition.*
- void **set\_resampler\_latency** (uint32\_t latency\_samples \_\_attribute\_\_((unused))) override

### Public Member Functions inherited from AcquisitionInterface

- virtual void **set\_doppler\_center** (int doppler\_center \_\_attribute\_\_((unused)))
- virtual void **set\_resampler\_latency** (uint32\_t latency\_samples)=0

### Public Member Functions inherited from GNSSBlockInterface

- virtual void **connect** (gr::top\_block\_sptr top\_block)=0
- virtual void **disconnect** (gr::top\_block\_sptr top\_block)=0
- virtual gr::basic\_block\_sptr **get\_left\_block** (int RF\_channel)
- virtual gr::basic\_block\_sptr **get\_right\_block** (int RF\_channel)
- virtual void **start** ()

*Start the flow of samples if needed.*

#### 10.215.1 Detailed Description

This class Adapts a PCPS acquisition block with fine Doppler estimation to an [AcquisitionInterface](#) for GPS L1 C/A signals.

Definition at line 45 of file [gps\\_l1\\_ca\\_pcps\\_acquisition\\_fine\\_doppler.h](#).



## 10.215.2 Member Function Documentation

### 10.215.2.1 `get_left_block()`

```
gnss_shared_ptr< gr::basic_block > GpsL1CaPcpsAcquisitionFineDoppler::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.215.2.2 `get_right_block()`

```
gnss_shared_ptr< gr::basic_block > GpsL1CaPcpsAcquisitionFineDoppler::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.215.2.3 `implementation()`

```
std::string GpsL1CaPcpsAcquisitionFineDoppler::implementation () [inline], [override], [virtual]
```

Returns "GPS\_L1\_CA\_PCPS\_Acquisition\_Fine\_Doppler".

Implements [GNSSBlockInterface](#).

Definition at line 63 of file [gps\\_l1\\_ca\\_pcps\\_acquisition\\_fine\\_doppler.h](#).

### 10.215.2.4 `init()`

```
void GpsL1CaPcpsAcquisitionFineDoppler::init () [override], [virtual]
```

Initializes acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.215.2.5 `item_size()`

```
size_t GpsL1CaPcpsAcquisitionFineDoppler::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 68 of file [gps\\_l1\\_ca\\_pcps\\_acquisition\\_fine\\_doppler.h](#).

### 10.215.2.6 `mag()`

```
signed int GpsL1CaPcpsAcquisitionFineDoppler::mag () [override], [virtual]
```

Returns the maximum peak of grid search.

Implements [AcquisitionInterface](#).

### 10.215.2.7 `reset()`

```
void GpsL1CaPcpsAcquisitionFineDoppler::reset () [override], [virtual]
```

Restart acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.215.2.8 `role()`

```
std::string GpsL1CaPcpsAcquisitionFineDoppler::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 55 of file [gps\\_l1\\_ca\\_pcps\\_acquisition\\_fine\\_doppler.h](#).

### 10.215.2.9 `set_channel()`

```
void GpsL1CaPcpsAcquisitionFineDoppler::set_channel ( unsigned int channel) [inline], [override], [virtual]
```

Set acquisition channel unique ID.

Implements [AcquisitionInterface](#).

Definition at line 88 of file [gps\\_l1\\_ca\\_pcps\\_acquisition\\_fine\\_doppler.h](#).

**10.215.2.10 set\_channel\_fsm()**

```
void GpsL1CaPcpsAcquisitionFineDoppler::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline], [override], [virtual]
```

Set channel fsm associated to this acquisition instance.

Implements [AcquisitionInterface](#).

Definition at line 97 of file [gps\\_l1\\_ca\\_pcps\\_acquisition\\_fine\\_doppler.h](#).

**10.215.2.11 set\_doppler\_max()**

```
void GpsL1CaPcpsAcquisitionFineDoppler::set_doppler_max (
    unsigned int doppler_max) [override], [virtual]
```

Set maximum Doppler off grid search.

Implements [AcquisitionInterface](#).

**10.215.2.12 set\_doppler\_step()**

```
void GpsL1CaPcpsAcquisitionFineDoppler::set_doppler_step (
    unsigned int doppler_step) [override], [virtual]
```

Set Doppler steps for the grid search.

Implements [AcquisitionInterface](#).

**10.215.2.13 set\_gnss\_synchro()**

```
void GpsL1CaPcpsAcquisitionFineDoppler::set_gnss_synchro (
    Gnss\_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [AcquisitionInterface](#).

**10.215.2.14 set\_local\_code()**

```
void GpsL1CaPcpsAcquisitionFineDoppler::set_local_code () [override], [virtual]
```

Implements [AcquisitionInterface](#).

**10.215.2.15 set\_resampler\_latency()**

```
void GpsL1CaPcpsAcquisitionFineDoppler::set_resampler_latency (
    uint32_t latency_samples __attribute__((unused))) [inline], [override]
```

Definition at line 145 of file [gps\\_l1\\_ca\\_pcps\\_acquisition\\_fine\\_doppler.h](#).

**10.215.2.16 set\_state()**

```
void GpsL1CaPcpsAcquisitionFineDoppler::set_state (
    int state) [override], [virtual]
```

If state = 1, it forces the block to start acquiring from the first sample.

Implements [AcquisitionInterface](#).

**10.215.2.17 set\_threshold()**

```
void GpsL1CaPcpsAcquisitionFineDoppler::set_threshold (
    float threshold) [override], [virtual]
```

Set statistics threshold of PCPS algorithm.

Implements [AcquisitionInterface](#).

**10.215.2.18 stop\_acquisition()**

```
void GpsL1CaPcpsAcquisitionFineDoppler::stop_acquisition () [override], [virtual]
```

Stop running acquisition.

Implements [AcquisitionInterface](#).

The documentation for this class was generated from the following file:

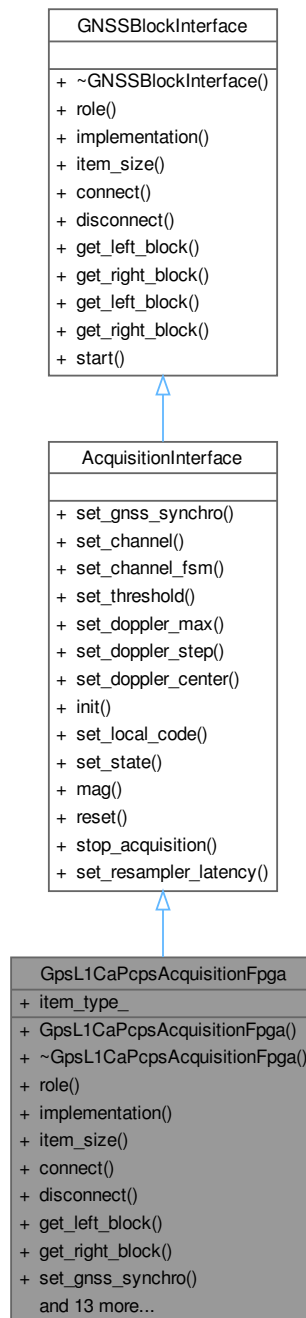
- [gps\\_l1\\_ca\\_pcps\\_acquisition\\_fine\\_doppler.h](#)

## 10.216 GpsL1CaPcpsAcquisitionFpga Class Reference

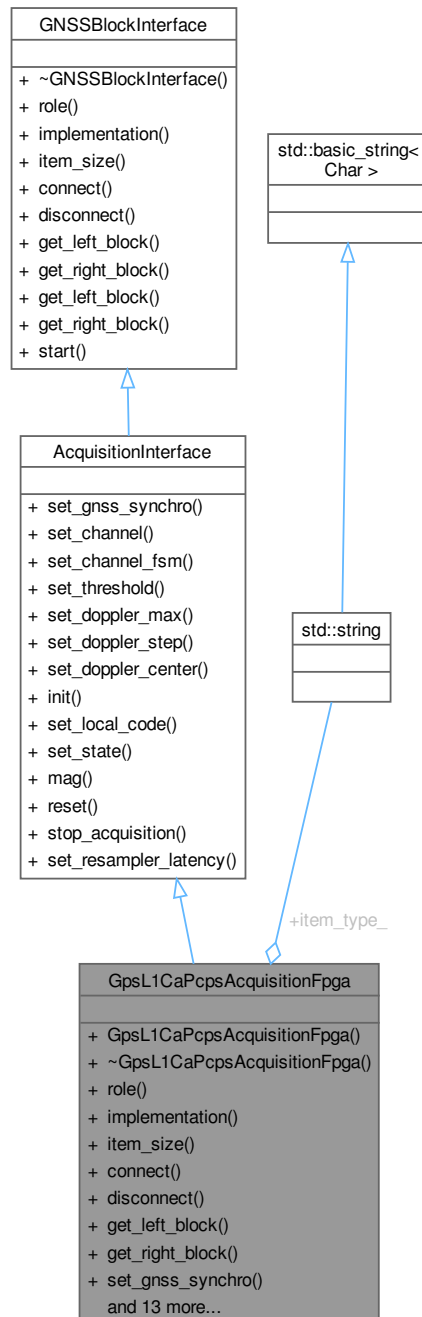
This class adapts a PCPS acquisition block off-loaded on an FPGA to an [AcquisitionInterface](#) for GPS L1 C/A signals.

```
#include <gps_l1_ca_pcps_acquisition_fpga.h>
```

Inheritance diagram for GpsL1CaPcpsAcquisitionFpga:



Collaboration diagram for GpsL1CaPcpsAcquisitionFpga:



## Public Member Functions

- `GpsL1CaPcpsAcquisitionFpga` (const `ConfigurationInterface` \*configuration, const `std::string` &role, unsigned int in\_streams, unsigned int out\_streams)  
*Constructor.*
- `~GpsL1CaPcpsAcquisitionFpga` ()=default  
*Destructor.*
- `std::string` role () override

- Role.*
- std::string [implementation](#) () override  
*Returns "GPS\_L1\_CA\_PCPS\_Acquisition\_FPGA".*
- size\_t [item\\_size](#) () override  
*Returns size of lv\_16sc\_t.*
- void [connect](#) (gr::top\_block\_sptr top\_block) override  
*Connect.*
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override  
*Disconnect.*
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override  
*Get left block.*
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override  
*Get right block.*
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [set\\_channel](#) (unsigned int channel) override  
*Set acquisition channel unique ID.*
- void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm) override  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold) override
- void [set\\_doppler\\_max](#) (unsigned int doppler\_max) override  
*Set maximum Doppler off grid search.*
- void [set\\_doppler\\_step](#) (unsigned int doppler\_step) override  
*Set Doppler steps for the grid search.*
- void [set\\_doppler\\_center](#) (int doppler\_center) override  
*Set Doppler center for the grid search.*
- void [init](#) () override  
*Initializes acquisition algorithm.*
- void [set\\_local\\_code](#) () override  
*Sets local code for GPS L1/CA PCPS acquisition algorithm.*
- signed int [mag](#) () override  
*Returns the maximum peak of grid search.*
- void [reset](#) () override  
*Restart acquisition algorithm.*
- void [set\\_state](#) (int state) override  
*If state = 1, it forces the block to start acquiring from the first sample.*
- void [stop\\_acquisition](#) () override  
*Stop running acquisition.*
- void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples \_\_attribute\_\_((unused))) override  
*Set Resampler Latency.*

### Public Member Functions inherited from [AcquisitionInterface](#)

- virtual void [set\\_doppler\\_center](#) (int doppler\_center \_\_attribute\_\_((unused)))
- virtual void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples)=0

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()  
*Start the flow of samples if needed.*

## Public Attributes

- `std::string item_type_`  
*Set statistics threshold of PCPS algorithm.*

### 10.216.1 Detailed Description

This class adapts a PCPS acquisition block off-loaded on an FPGA to an [AcquisitionInterface](#) for GPS L1 C/A signals.

Definition at line 45 of file [gps\\_l1\\_ca\\_pcps\\_acquisition\\_fpga.h](#).

### 10.216.2 Constructor & Destructor Documentation

#### 10.216.2.1 GpsL1CaPcpsAcquisitionFpga()

```
GpsL1CaPcpsAcquisitionFpga::GpsL1CaPcpsAcquisitionFpga (
    const ConfigurationInterface * configuration,
    const std::string & role,
    unsigned int in_streams,
    unsigned int out_streams)
```

Constructor.

References [role\(\)](#).

Here is the call graph for this function:



#### 10.216.2.2 ~GpsL1CaPcpsAcquisitionFpga()

```
GpsL1CaPcpsAcquisitionFpga::~GpsL1CaPcpsAcquisitionFpga () [default]
Destructor.
```

### 10.216.3 Member Function Documentation

#### 10.216.3.1 connect()

```
void GpsL1CaPcpsAcquisitionFpga::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Connect.

Implements [GNSSBlockInterface](#).

#### 10.216.3.2 disconnect()

```
void GpsL1CaPcpsAcquisitionFpga::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Disconnect.

Implements [GNSSBlockInterface](#).

#### 10.216.3.3 get\_left\_block()

```
gr::basic_block_sptr GpsL1CaPcpsAcquisitionFpga::get_left_block () [override], [virtual]
```

Get left block.

Implements [GNSSBlockInterface](#).

**10.216.3.4 get\_right\_block()**

```
gr::basic_block_sptr GpsL1CaPcpsAcquisitionFpga::get_right_block () [override], [virtual]
```

Get right block.

Implements [GNSSBlockInterface](#).

**10.216.3.5 implementation()**

```
std::string GpsL1CaPcpsAcquisitionFpga::implementation () [inline], [override], [virtual]
```

Returns "GPS\_L1\_CA\_PCPS\_Acquisition\_FPGA".

Implements [GNSSBlockInterface](#).

Definition at line 72 of file [gps\\_l1\\_ca\\_pcps\\_acquisition\\_fpga.h](#).

**10.216.3.6 init()**

```
void GpsL1CaPcpsAcquisitionFpga::init () [override], [virtual]
```

Initializes acquisition algorithm.

Implements [AcquisitionInterface](#).

**10.216.3.7 item\_size()**

```
size_t GpsL1CaPcpsAcquisitionFpga::item_size () [inline], [override], [virtual]
```

Returns size of `lv_16sc_t`.

Implements [GNSSBlockInterface](#).

Definition at line 80 of file [gps\\_l1\\_ca\\_pcps\\_acquisition\\_fpga.h](#).

**10.216.3.8 mag()**

```
signed int GpsL1CaPcpsAcquisitionFpga::mag () [override], [virtual]
```

Returns the maximum peak of grid search.

Implements [AcquisitionInterface](#).

**10.216.3.9 reset()**

```
void GpsL1CaPcpsAcquisitionFpga::reset () [override], [virtual]
```

Restart acquisition algorithm.

Implements [AcquisitionInterface](#).

**10.216.3.10 role()**

```
std::string GpsL1CaPcpsAcquisitionFpga::role () [inline], [override], [virtual]
```

Role.

Implements [GNSSBlockInterface](#).

Definition at line 64 of file [gps\\_l1\\_ca\\_pcps\\_acquisition\\_fpga.h](#).

Referenced by [GpsL1CaPcpsAcquisitionFpga\(\)](#).

**10.216.3.11 set\_channel()**

```
void GpsL1CaPcpsAcquisitionFpga::set_channel (
    unsigned int channel) [inline], [override], [virtual]
```

Set acquisition channel unique ID.

Implements [AcquisitionInterface](#).

Definition at line 115 of file [gps\\_l1\\_ca\\_pcps\\_acquisition\\_fpga.h](#).

**10.216.3.12 set\_channel\_fsm()**

```
void GpsL1CaPcpsAcquisitionFpga::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline], [override], [virtual]
```

Set channel fsm associated to this acquisition instance.

Implements [AcquisitionInterface](#).

Definition at line 124 of file [gps\\_l1\\_ca\\_pcps\\_acquisition\\_fpga.h](#).

#### 10.216.3.13 `set_doppler_center()`

```
void GpsL1CaPcpsAcquisitionFpga::set_doppler_center (
    int doppler_center) [override]
```

Set Doppler center for the grid search.

#### 10.216.3.14 `set_doppler_max()`

```
void GpsL1CaPcpsAcquisitionFpga::set_doppler_max (
    unsigned int doppler_max) [override], [virtual]
```

Set maximum Doppler off grid search.

Implements [AcquisitionInterface](#).

#### 10.216.3.15 `set_doppler_step()`

```
void GpsL1CaPcpsAcquisitionFpga::set_doppler_step (
    unsigned int doppler_step) [override], [virtual]
```

Set Doppler steps for the grid search.

Implements [AcquisitionInterface](#).

#### 10.216.3.16 `set_gnss_synchro()`

```
void GpsL1CaPcpsAcquisitionFpga::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [AcquisitionInterface](#).

#### 10.216.3.17 `set_local_code()`

```
void GpsL1CaPcpsAcquisitionFpga::set_local_code () [override], [virtual]
```

Sets local code for GPS L1/CA PCPS acquisition algorithm.

Implements [AcquisitionInterface](#).

#### 10.216.3.18 `set_resampler_latency()`

```
void GpsL1CaPcpsAcquisitionFpga::set_resampler_latency (
    uint32_t latency_samples __attribute__((unused))) [inline], [override]
```

Set Resampler Latency.

Definition at line 185 of file [gps\\_l1\\_ca\\_pcps\\_acquisition\\_fpga.h](#).

#### 10.216.3.19 `set_state()`

```
void GpsL1CaPcpsAcquisitionFpga::set_state (
    int state) [override], [virtual]
```

If state = 1, it forces the block to start acquiring from the first sample.

Implements [AcquisitionInterface](#).

#### 10.216.3.20 `set_threshold()`

```
void GpsL1CaPcpsAcquisitionFpga::set_threshold (
    float threshold) [override], [virtual]
```

Implements [AcquisitionInterface](#).



### 10.216.3.21 stop\_acquisition()

```
void GpsL1CaPcpsAcquisitionFpga::stop_acquisition () [override], [virtual]
```

Stop running acquisition.

Implements [AcquisitionInterface](#).

## 10.216.4 Member Data Documentation

### 10.216.4.1 item\_type\_

```
std::string GpsL1CaPcpsAcquisitionFpga::item_type_
```

Set statistics threshold of PCPS algorithm.

Definition at line 133 of file [gps\\_l1\\_ca\\_pcps\\_acquisition\\_fpga.h](#).

The documentation for this class was generated from the following file:

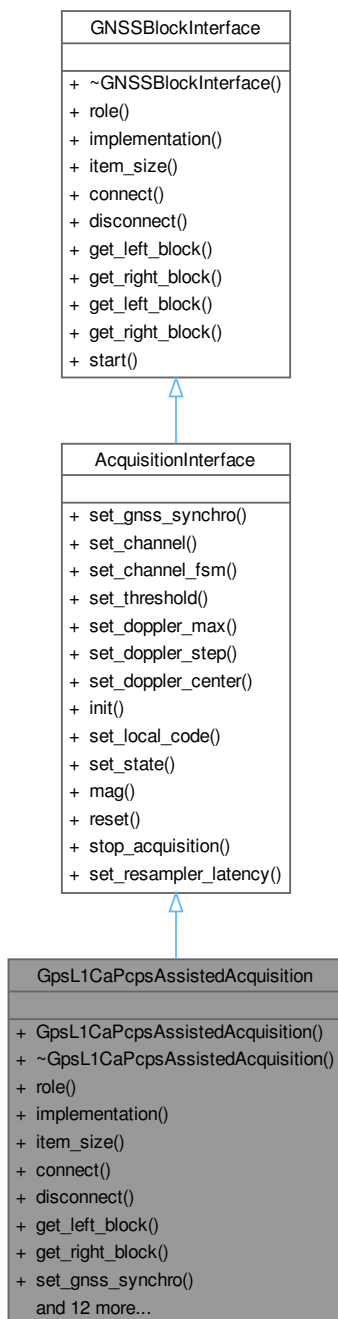
- [gps\\_l1\\_ca\\_pcps\\_acquisition\\_fpga.h](#)

## 10.217 GpsL1CaPcpsAssistedAcquisition Class Reference

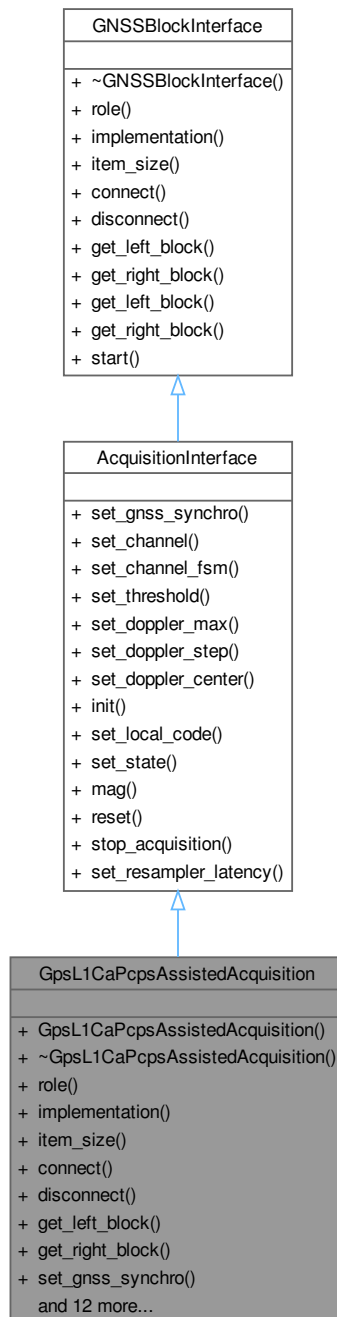
This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.

```
#include <gps_l1_ca_pcps_assisted_acquisition.h>
```

Inheritance diagram for GpsL1CaPcpsAssistedAcquisition:



Collaboration diagram for GpsL1CaPcpsAssistedAcquisition:



## Public Member Functions

- **GpsL1CaPcpsAssistedAcquisition** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override
  - Returns "GPS\_L1\_CA\_PCPS\_Assisted\_Acquisition".
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [set\\_channel](#) (unsigned int channel) override  
*Set acquisition channel unique ID.*
- void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm) override  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold) override  
*Set statistics threshold of PCPS algorithm.*
- void [set\\_doppler\\_max](#) (unsigned int doppler\_max) override  
*Set maximum Doppler off grid search.*
- void [set\\_doppler\\_step](#) (unsigned int doppler\_step) override  
*Set Doppler steps for the grid search.*
- void [init](#) () override  
*Initializes acquisition algorithm.*
- void [set\\_local\\_code](#) () override
- signed int [mag](#) () override  
*Returns the maximum peak of grid search.*
- void [reset](#) () override  
*Restart acquisition algorithm.*
- void [set\\_state](#) (int state \_\_attribute\_\_((unused))) override
- void [stop\\_acquisition](#) () override  
*Stop running acquisition.*
- void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples \_\_attribute\_\_((unused))) override

### Public Member Functions inherited from [AcquisitionInterface](#)

- virtual void [set\\_doppler\\_center](#) (int doppler\_center \_\_attribute\_\_((unused)))
- virtual void [set\\_state](#) (int state)=0
- virtual void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples)=0

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()  
*Start the flow of samples if needed.*

#### 10.217.1 Detailed Description

This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.

Definition at line 43 of file [gps\\_l1\\_ca\\_pcps\\_assisted\\_acquisition.h](#).

#### 10.217.2 Member Function Documentation

##### 10.217.2.1 [connect](#)()

```
void GpsL1CaPcpsAssistedAcquisition::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.217.2.2 disconnect()

```
void GpsL1CaPcpsAssistedAcquisition::disconnect (  
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.217.2.3 get\_left\_block()

```
gr::basic_block_sptr GpsL1CaPcpsAssistedAcquisition::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.217.2.4 get\_right\_block()

```
gr::basic_block_sptr GpsL1CaPcpsAssistedAcquisition::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.217.2.5 implementation()

```
std::string GpsL1CaPcpsAssistedAcquisition::implementation () [inline], [override], [virtual]
```

Returns "GPS\_L1\_CA\_PCPS\_Assisted\_Acquisition".

Implements [GNSSBlockInterface](#).

Definition at line 62 of file [gps\\_l1\\_ca\\_pcps\\_assisted\\_acquisition.h](#).

### 10.217.2.6 init()

```
void GpsL1CaPcpsAssistedAcquisition::init () [override], [virtual]
```

Initializes acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.217.2.7 item\_size()

```
size_t GpsL1CaPcpsAssistedAcquisition::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 67 of file [gps\\_l1\\_ca\\_pcps\\_assisted\\_acquisition.h](#).

### 10.217.2.8 mag()

```
signed int GpsL1CaPcpsAssistedAcquisition::mag () [override], [virtual]
```

Returns the maximum peak of grid search.

Implements [AcquisitionInterface](#).

### 10.217.2.9 reset()

```
void GpsL1CaPcpsAssistedAcquisition::reset () [override], [virtual]
```

Restart acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.217.2.10 role()

```
std::string GpsL1CaPcpsAssistedAcquisition::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 54 of file [gps\\_l1\\_ca\\_pcps\\_assisted\\_acquisition.h](#).

### 10.217.2.11 set\_channel()

```
void GpsL1CaPcpsAssistedAcquisition::set_channel (  
    unsigned int channel) [inline], [override], [virtual]
```

Set acquisition channel unique ID.

Implements [AcquisitionInterface](#).

Definition at line 87 of file [gps\\_l1\\_ca\\_pcps\\_assisted\\_acquisition.h](#).

**10.217.2.12 set\_channel\_fsm()**

```
void GpsL1CaPcpsAssistedAcquisition::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline], [override], [virtual]
```

Set channel fsm associated to this acquisition instance.

Implements [AcquisitionInterface](#).

Definition at line 96 of file [gps\\_l1\\_ca\\_pcps\\_assisted\\_acquisition.h](#).

**10.217.2.13 set\_doppler\_max()**

```
void GpsL1CaPcpsAssistedAcquisition::set_doppler_max (
    unsigned int doppler_max) [override], [virtual]
```

Set maximum Doppler off grid search.

Implements [AcquisitionInterface](#).

**10.217.2.14 set\_doppler\_step()**

```
void GpsL1CaPcpsAssistedAcquisition::set_doppler_step (
    unsigned int doppler_step) [override], [virtual]
```

Set Doppler steps for the grid search.

Implements [AcquisitionInterface](#).

**10.217.2.15 set\_gnss\_synchro()**

```
void GpsL1CaPcpsAssistedAcquisition::set_gnss_synchro (
    Gnss\_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [AcquisitionInterface](#).

**10.217.2.16 set\_local\_code()**

```
void GpsL1CaPcpsAssistedAcquisition::set_local_code () [override], [virtual]
```

Implements [AcquisitionInterface](#).

**10.217.2.17 set\_resampler\_latency()**

```
void GpsL1CaPcpsAssistedAcquisition::set_resampler_latency (
    uint32_t latency_samples __attribute__((unused))) [inline], [override]
```

Definition at line 140 of file [gps\\_l1\\_ca\\_pcps\\_assisted\\_acquisition.h](#).

**10.217.2.18 set\_state()**

```
void GpsL1CaPcpsAssistedAcquisition::set_state (
    int state __attribute__((unused))) [inline], [override]
```

Definition at line 133 of file [gps\\_l1\\_ca\\_pcps\\_assisted\\_acquisition.h](#).

**10.217.2.19 set\_threshold()**

```
void GpsL1CaPcpsAssistedAcquisition::set_threshold (
    float threshold) [override], [virtual]
```

Set statistics threshold of PCPS algorithm.

Implements [AcquisitionInterface](#).

**10.217.2.20 stop\_acquisition()**

```
void GpsL1CaPcpsAssistedAcquisition::stop_acquisition () [override], [virtual]
```

Stop running acquisition.

Implements [AcquisitionInterface](#).

The documentation for this class was generated from the following file:

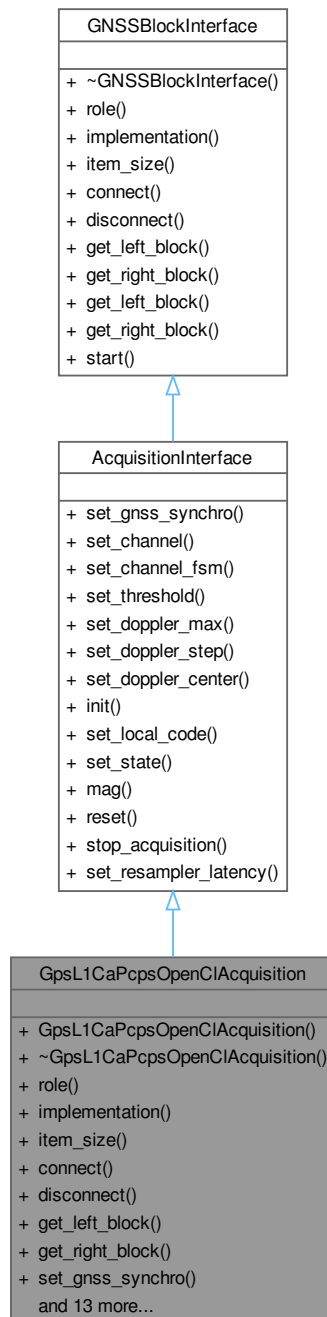
- [gps\\_l1\\_ca\\_pcps\\_assisted\\_acquisition.h](#)

## 10.218 GpsL1CaPcpsOpenCIAcquisition Class Reference

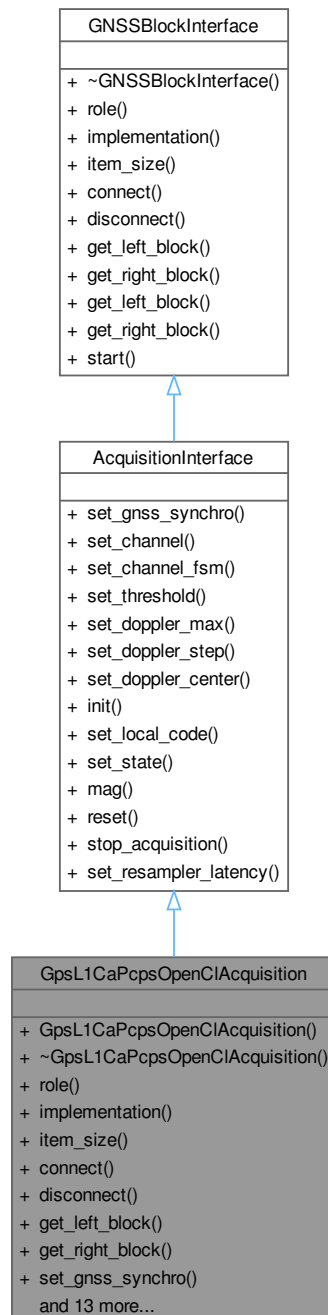
This class adapts an OpenCL PCPS acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.

```
#include <gps_l1_ca_pcps_openc1_acquisition.h>
```

Inheritance diagram for GpsL1CaPcpsOpenCIAcquisition:



Collaboration diagram for GpsL1CaPcpsOpenCIAcquisition:



## Public Member Functions

- **GpsL1CaPcpsOpenCIAcquisition** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override
  - Returns "GPS\_L1\_CA\_PCPS\_OpenCI\_Acquisition".
- size\_t [item\\_size](#) () override



- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_gnss\\_synchro](#) ([Gnss\\_Synchro](#) \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [set\\_channel](#) (unsigned int channel) override  
*Set acquisition channel unique ID.*
- void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm) override  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold) override  
*Set statistics threshold of PCPS algorithm.*
- void [set\\_doppler\\_max](#) (unsigned int doppler\_max) override  
*Set maximum Doppler off grid search.*
- void [set\\_doppler\\_step](#) (unsigned int doppler\_step) override  
*Set Doppler steps for the grid search.*
- void [init](#) () override  
*Initializes acquisition algorithm.*
- void [set\\_local\\_code](#) () override  
*Sets local code for GPS L1/CA PCPS acquisition algorithm.*
- signed int [mag](#) () override  
*Returns the maximum peak of grid search.*
- void [reset](#) () override  
*Restart acquisition algorithm.*
- void [set\\_state](#) (int state \_\_attribute\_\_((unused))) override
- void [stop\\_acquisition](#) () override  
*Stop running acquisition.*
- void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples \_\_attribute\_\_((unused))) override
- bool [openc1\\_ready](#) () const

### Public Member Functions inherited from [AcquisitionInterface](#)

- virtual void [set\\_doppler\\_center](#) (int doppler\_center \_\_attribute\_\_((unused)))
- virtual void [set\\_state](#) (int state)=0
- virtual void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples)=0

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

## 10.218.1 Detailed Description

This class adapts an OpenCL PCPS acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals. Definition at line 42 of file [gps\\_l1\\_ca\\_pcps\\_openc1\\_acquisition.h](#).

## 10.218.2 Member Function Documentation

### 10.218.2.1 [connect](#)()

```
void GpsL1CaPcpsOpenClAcquisition::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.218.2.2 disconnect()

```
void GpsL1CaPcpsOpenClAcquisition::disconnect (  
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.218.2.3 get\_left\_block()

```
gr::basic_block_sptr GpsL1CaPcpsOpenClAcquisition::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.218.2.4 get\_right\_block()

```
gr::basic_block_sptr GpsL1CaPcpsOpenClAcquisition::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.218.2.5 implementation()

```
std::string GpsL1CaPcpsOpenClAcquisition::implementation () [inline], [override], [virtual]
```

Returns "GPS\_L1\_CA\_PCPS\_OpenCl\_Acquisition".

Implements [GNSSBlockInterface](#).

Definition at line 60 of file [gps\\_l1\\_ca\\_pcps\\_openc1\\_acquisition.h](#).

#### 10.218.2.6 init()

```
void GpsL1CaPcpsOpenClAcquisition::init () [override], [virtual]
```

Initializes acquisition algorithm.

Implements [AcquisitionInterface](#).

#### 10.218.2.7 item\_size()

```
size_t GpsL1CaPcpsOpenClAcquisition::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 65 of file [gps\\_l1\\_ca\\_pcps\\_openc1\\_acquisition.h](#).

#### 10.218.2.8 mag()

```
signed int GpsL1CaPcpsOpenClAcquisition::mag () [override], [virtual]
```

Returns the maximum peak of grid search.

Implements [AcquisitionInterface](#).

#### 10.218.2.9 openc1\_ready()

```
bool GpsL1CaPcpsOpenClAcquisition::openc1_ready () const [inline]
```

Definition at line 143 of file [gps\\_l1\\_ca\\_pcps\\_openc1\\_acquisition.h](#).

#### 10.218.2.10 reset()

```
void GpsL1CaPcpsOpenClAcquisition::reset () [override], [virtual]
```

Restart acquisition algorithm.

Implements [AcquisitionInterface](#).

#### 10.218.2.11 role()

```
std::string GpsL1CaPcpsOpenClAcquisition::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 52 of file [gps\\_l1\\_ca\\_pcps\\_openc1\\_acquisition.h](#).

**10.218.2.12 set\_channel()**

```
void GpsL1CaPcpsOpenClAcquisition::set_channel (
    unsigned int channel) [inline], [override], [virtual]
```

Set acquisition channel unique ID.

Implements [AcquisitionInterface](#).

Definition at line 85 of file [gps\\_l1\\_ca\\_pcps\\_openc1\\_acquisition.h](#).

**10.218.2.13 set\_channel\_fsm()**

```
void GpsL1CaPcpsOpenClAcquisition::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline], [override], [virtual]
```

Set channel fsm associated to this acquisition instance.

Implements [AcquisitionInterface](#).

Definition at line 94 of file [gps\\_l1\\_ca\\_pcps\\_openc1\\_acquisition.h](#).

**10.218.2.14 set\_doppler\_max()**

```
void GpsL1CaPcpsOpenClAcquisition::set_doppler_max (
    unsigned int doppler_max) [override], [virtual]
```

Set maximum Doppler off grid search.

Implements [AcquisitionInterface](#).

**10.218.2.15 set\_doppler\_step()**

```
void GpsL1CaPcpsOpenClAcquisition::set_doppler_step (
    unsigned int doppler_step) [override], [virtual]
```

Set Doppler steps for the grid search.

Implements [AcquisitionInterface](#).

**10.218.2.16 set\_gnss\_synchro()**

```
void GpsL1CaPcpsOpenClAcquisition::set_gnss_synchro (
    Gnss\_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [AcquisitionInterface](#).

**10.218.2.17 set\_local\_code()**

```
void GpsL1CaPcpsOpenClAcquisition::set_local_code () [override], [virtual]
```

Sets local code for GPS L1/CA PCPS acquisition algorithm.

Implements [AcquisitionInterface](#).

**10.218.2.18 set\_resampler\_latency()**

```
void GpsL1CaPcpsOpenClAcquisition::set_resampler_latency (
    uint32_t latency_samples __attribute__((unused))) [inline], [override]
```

Definition at line 141 of file [gps\\_l1\\_ca\\_pcps\\_openc1\\_acquisition.h](#).

**10.218.2.19 set\_state()**

```
void GpsL1CaPcpsOpenClAcquisition::set_state (
    int state __attribute__((unused))) [inline], [override]
```

Definition at line 134 of file [gps\\_l1\\_ca\\_pcps\\_openc1\\_acquisition.h](#).

**10.218.2.20 set\_threshold()**

```
void GpsL1CaPcpsOpenClAcquisition::set_threshold (
    float threshold) [override], [virtual]
```

Set statistics threshold of PCPS algorithm.  
Implements [AcquisitionInterface](#).

#### 10.218.2.21 stop\_acquisition()

```
void GpsL1CaPcpsOpenClAcquisition::stop_acquisition () [override], [virtual]
```

Stop running acquisition.

Implements [AcquisitionInterface](#).

The documentation for this class was generated from the following file:

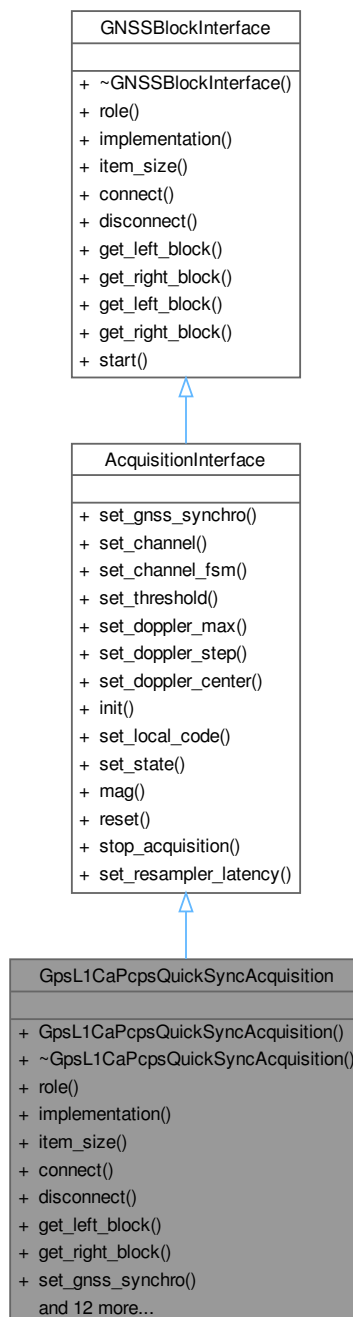
- [gps\\_l1\\_ca\\_pcps\\_openc1\\_acquisition.h](#)

## 10.219 GpsL1CaPcpsQuickSyncAcquisition Class Reference

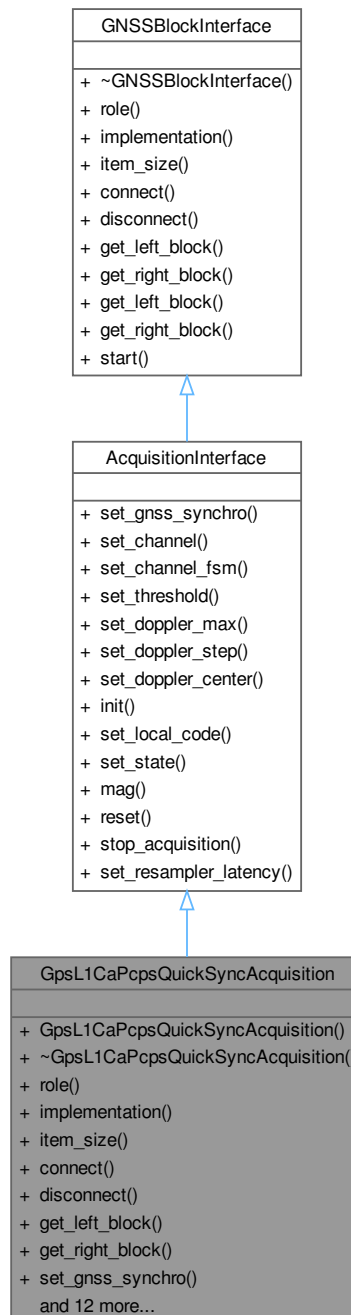
This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.

```
#include <gps_l1_ca_pcps_quicksync_acquisition.h>
```

Inheritance diagram for GpsL1CaPcpsQuickSyncAcquisition:



Collaboration diagram for GpsL1CaPcpsQuickSyncAcquisition:



## Public Member Functions

- **GpsL1CaPcpsQuickSyncAcquisition** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override
- Returns "GPS\_L1\_CA\_PCPS\_QuickSync\_Acquisition".
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [set\\_channel](#) (unsigned int channel) override  
*Set acquisition channel unique ID.*
- void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm) override  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold) override  
*Set statistics threshold of PCPS algorithm.*
- void [set\\_doppler\\_max](#) (unsigned int doppler\_max) override  
*Set maximum Doppler off grid search.*
- void [set\\_doppler\\_step](#) (unsigned int doppler\_step) override  
*Set Doppler steps for the grid search.*
- void [init](#) () override  
*Initializes acquisition algorithm.*
- void [set\\_local\\_code](#) () override  
*Sets local code for GPS L1/CA PCPS acquisition algorithm.*
- signed int [mag](#) () override  
*Returns the maximum peak of grid search.*
- void [reset](#) () override  
*Restart acquisition algorithm.*
- void [set\\_state](#) (int state) override  
*If state = 1, it forces the block to start acquiring from the first sample.*
- void [stop\\_acquisition](#) () override  
*Stop running acquisition.*
- void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples \_\_attribute\_\_((unused))) override

### Public Member Functions inherited from [AcquisitionInterface](#)

- virtual void [set\\_doppler\\_center](#) (int doppler\_center \_\_attribute\_\_((unused)))
- virtual void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples)=0

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

## 10.219.1 Detailed Description

This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals. Definition at line 44 of file [gps\\_l1\\_ca\\_pcps\\_quicksync\\_acquisition.h](#).

## 10.219.2 Member Function Documentation

### 10.219.2.1 [connect\(\)](#)

```
void GpsL1CaPcpsQuickSyncAcquisition::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.219.2.2 disconnect()

```
void GpsL1CaPcpsQuickSyncAcquisition::disconnect (  
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.219.2.3 get\_left\_block()

```
gr::basic_block_sptr GpsL1CaPcpsQuickSyncAcquisition::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.219.2.4 get\_right\_block()

```
gr::basic_block_sptr GpsL1CaPcpsQuickSyncAcquisition::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.219.2.5 implementation()

```
std::string GpsL1CaPcpsQuickSyncAcquisition::implementation () [inline], [override], [virtual]
```

Returns "GPS\_L1\_CA\_PCPS\_QuickSync\_Acquisition".

Implements [GNSSBlockInterface](#).

Definition at line 63 of file [gps\\_l1\\_ca\\_pcps\\_quicksync\\_acquisition.h](#).

### 10.219.2.6 init()

```
void GpsL1CaPcpsQuickSyncAcquisition::init () [override], [virtual]
```

Initializes acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.219.2.7 item\_size()

```
size_t GpsL1CaPcpsQuickSyncAcquisition::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 68 of file [gps\\_l1\\_ca\\_pcps\\_quicksync\\_acquisition.h](#).

### 10.219.2.8 mag()

```
signed int GpsL1CaPcpsQuickSyncAcquisition::mag () [override], [virtual]
```

Returns the maximum peak of grid search.

Implements [AcquisitionInterface](#).

### 10.219.2.9 reset()

```
void GpsL1CaPcpsQuickSyncAcquisition::reset () [override], [virtual]
```

Restart acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.219.2.10 role()

```
std::string GpsL1CaPcpsQuickSyncAcquisition::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 55 of file [gps\\_l1\\_ca\\_pcps\\_quicksync\\_acquisition.h](#).

### 10.219.2.11 set\_channel()

```
void GpsL1CaPcpsQuickSyncAcquisition::set_channel (  
    unsigned int channel) [inline], [override], [virtual]
```

Set acquisition channel unique ID.

Implements [AcquisitionInterface](#).

Definition at line 88 of file [gps\\_l1\\_ca\\_pcps\\_quicksync\\_acquisition.h](#).



**10.219.2.12 set\_channel\_fsm()**

```
void GpsL1CaPcpsQuickSyncAcquisition::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline], [override], [virtual]
```

Set channel fsm associated to this acquisition instance.

Implements [AcquisitionInterface](#).

Definition at line 97 of file [gps\\_l1\\_ca\\_pcps\\_quicksync\\_acquisition.h](#).

**10.219.2.13 set\_doppler\_max()**

```
void GpsL1CaPcpsQuickSyncAcquisition::set_doppler_max (
    unsigned int doppler_max) [override], [virtual]
```

Set maximum Doppler off grid search.

Implements [AcquisitionInterface](#).

**10.219.2.14 set\_doppler\_step()**

```
void GpsL1CaPcpsQuickSyncAcquisition::set_doppler_step (
    unsigned int doppler_step) [override], [virtual]
```

Set Doppler steps for the grid search.

Implements [AcquisitionInterface](#).

**10.219.2.15 set\_gnss\_synchro()**

```
void GpsL1CaPcpsQuickSyncAcquisition::set_gnss_synchro (
    Gnss\_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [AcquisitionInterface](#).

**10.219.2.16 set\_local\_code()**

```
void GpsL1CaPcpsQuickSyncAcquisition::set_local_code () [override], [virtual]
```

Sets local code for GPS L1/CA PCPS acquisition algorithm.

Implements [AcquisitionInterface](#).

**10.219.2.17 set\_resampler\_latency()**

```
void GpsL1CaPcpsQuickSyncAcquisition::set_resampler_latency (
    uint32_t latency_samples __attribute__((unused))) [inline], [override]
```

Definition at line 148 of file [gps\\_l1\\_ca\\_pcps\\_quicksync\\_acquisition.h](#).

**10.219.2.18 set\_state()**

```
void GpsL1CaPcpsQuickSyncAcquisition::set_state (
    int state) [override], [virtual]
```

If state = 1, it forces the block to start acquiring from the first sample.

Implements [AcquisitionInterface](#).

**10.219.2.19 set\_threshold()**

```
void GpsL1CaPcpsQuickSyncAcquisition::set_threshold (
    float threshold) [override], [virtual]
```

Set statistics threshold of PCPS algorithm.

Implements [AcquisitionInterface](#).

**10.219.2.20 stop\_acquisition()**

```
void GpsL1CaPcpsQuickSyncAcquisition::stop_acquisition () [override], [virtual]
```

Stop running acquisition.

Implements [AcquisitionInterface](#).

The documentation for this class was generated from the following file:

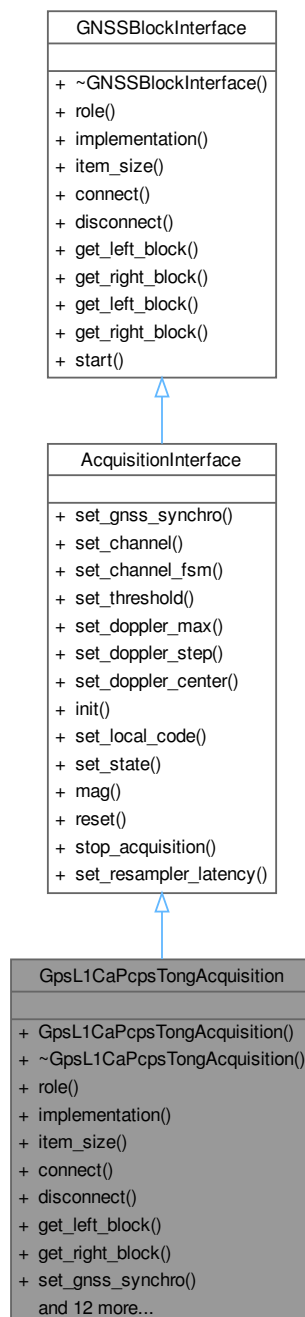
- [gps\\_l1\\_ca\\_pcps\\_quicksync\\_acquisition.h](#)

## 10.220 GpsL1CaPcpsTongAcquisition Class Reference

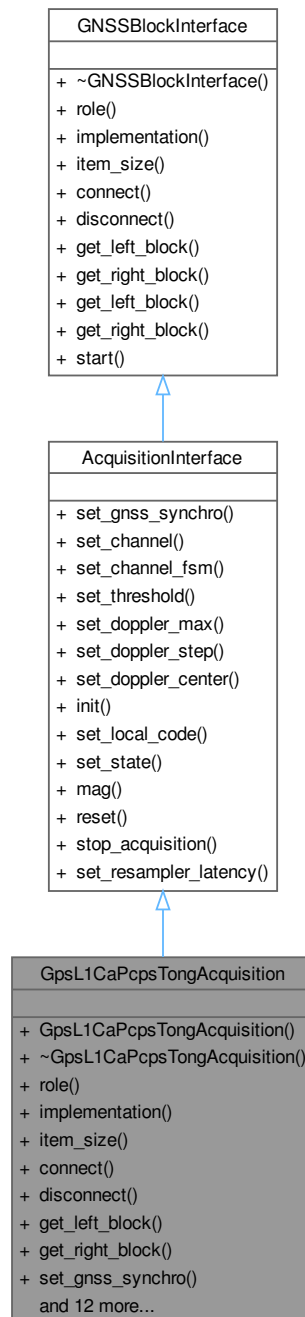
This class adapts a PCPS Tong acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.

```
#include <gps_l1_ca_pcps_tong_acquisition.h>
```

Inheritance diagram for GpsL1CaPcpsTongAcquisition:



Collaboration diagram for GpsL1CaPcpsTongAcquisition:



## Public Member Functions

- **GpsL1CaPcpsTongAcquisition** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override
  - Returns "GPS\_L1\_CA\_PCPS\_Tong\_Acquisition".
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [set\\_channel](#) (unsigned int channel) override  
*Set acquisition channel unique ID.*
- void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm) override  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold) override  
*Set statistics threshold of TONG algorithm.*
- void [set\\_doppler\\_max](#) (unsigned int doppler\_max) override  
*Set maximum Doppler off grid search.*
- void [set\\_doppler\\_step](#) (unsigned int doppler\_step) override  
*Set Doppler steps for the grid search.*
- void [init](#) () override  
*Initializes acquisition algorithm.*
- void [set\\_local\\_code](#) () override  
*Sets local code for GPS L1/CA TONG acquisition algorithm.*
- signed int [mag](#) () override  
*Returns the maximum peak of grid search.*
- void [reset](#) () override  
*Restart acquisition algorithm.*
- void [set\\_state](#) (int state) override  
*If state = 1, it forces the block to start acquiring from the first sample.*
- void [stop\\_acquisition](#) () override  
*Stop running acquisition.*
- void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples \_\_attribute\_\_((unused))) override

### Public Member Functions inherited from [AcquisitionInterface](#)

- virtual void [set\\_doppler\\_center](#) (int doppler\_center \_\_attribute\_\_((unused)))
- virtual void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples)=0

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

## 10.220.1 Detailed Description

This class adapts a PCPS Tong acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.  
Definition at line 43 of file [gps\\_l1\\_ca\\_pcps\\_tong\\_acquisition.h](#).

## 10.220.2 Member Function Documentation

### 10.220.2.1 [connect\(\)](#)

```
void GpsL1CaPcpsTongAcquisition::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.220.2.2 disconnect()

```
void GpsL1CaPcpsTongAcquisition::disconnect (  
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.220.2.3 get\_left\_block()

```
gr::basic_block_sptr GpsL1CaPcpsTongAcquisition::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.220.2.4 get\_right\_block()

```
gr::basic_block_sptr GpsL1CaPcpsTongAcquisition::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.220.2.5 implementation()

```
std::string GpsL1CaPcpsTongAcquisition::implementation () [inline], [override], [virtual]
```

Returns "GPS\_L1\_CA\_PCPS\_Tong\_Acquisition".

Implements [GNSSBlockInterface](#).

Definition at line 61 of file [gps\\_l1\\_ca\\_pcps\\_tong\\_acquisition.h](#).

### 10.220.2.6 init()

```
void GpsL1CaPcpsTongAcquisition::init () [override], [virtual]
```

Initializes acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.220.2.7 item\_size()

```
size_t GpsL1CaPcpsTongAcquisition::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 66 of file [gps\\_l1\\_ca\\_pcps\\_tong\\_acquisition.h](#).

### 10.220.2.8 mag()

```
signed int GpsL1CaPcpsTongAcquisition::mag () [override], [virtual]
```

Returns the maximum peak of grid search.

Implements [AcquisitionInterface](#).

### 10.220.2.9 reset()

```
void GpsL1CaPcpsTongAcquisition::reset () [override], [virtual]
```

Restart acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.220.2.10 role()

```
std::string GpsL1CaPcpsTongAcquisition::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 53 of file [gps\\_l1\\_ca\\_pcps\\_tong\\_acquisition.h](#).

### 10.220.2.11 set\_channel()

```
void GpsL1CaPcpsTongAcquisition::set_channel (  
    unsigned int channel) [inline], [override], [virtual]
```

Set acquisition channel unique ID.

Implements [AcquisitionInterface](#).

Definition at line 86 of file [gps\\_l1\\_ca\\_pcps\\_tong\\_acquisition.h](#).

#### 10.220.2.12 set\_channel\_fsm()

```
void GpsL1CaPcpsTongAcquisition::set_channel_fsm (  
    std::weak_ptr< ChannelFsm > channel_fsm) [inline], [override], [virtual]
```

Set channel fsm associated to this acquisition instance.

Implements [AcquisitionInterface](#).

Definition at line 95 of file [gps\\_l1\\_ca\\_pcps\\_tong\\_acquisition.h](#).

#### 10.220.2.13 set\_doppler\_max()

```
void GpsL1CaPcpsTongAcquisition::set_doppler_max (  
    unsigned int doppler_max) [override], [virtual]
```

Set maximum Doppler off grid search.

Implements [AcquisitionInterface](#).

#### 10.220.2.14 set\_doppler\_step()

```
void GpsL1CaPcpsTongAcquisition::set_doppler_step (  
    unsigned int doppler_step) [override], [virtual]
```

Set Doppler steps for the grid search.

Implements [AcquisitionInterface](#).

#### 10.220.2.15 set\_gnss\_synchro()

```
void GpsL1CaPcpsTongAcquisition::set_gnss_synchro (  
    Gnss\_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [AcquisitionInterface](#).

#### 10.220.2.16 set\_local\_code()

```
void GpsL1CaPcpsTongAcquisition::set_local_code () [override], [virtual]
```

Sets local code for GPS L1/CA TONG acquisition algorithm.

Implements [AcquisitionInterface](#).

#### 10.220.2.17 set\_resampler\_latency()

```
void GpsL1CaPcpsTongAcquisition::set_resampler_latency (  
    uint32_t latency_samples __attribute__((unused)) [inline], [override]
```

Definition at line 146 of file [gps\\_l1\\_ca\\_pcps\\_tong\\_acquisition.h](#).

#### 10.220.2.18 set\_state()

```
void GpsL1CaPcpsTongAcquisition::set_state (  
    int state) [override], [virtual]
```

If state = 1, it forces the block to start acquiring from the first sample.

Implements [AcquisitionInterface](#).

#### 10.220.2.19 set\_threshold()

```
void GpsL1CaPcpsTongAcquisition::set_threshold (  
    float threshold) [override], [virtual]
```

Set statistics threshold of TONG algorithm.

Implements [AcquisitionInterface](#).

#### 10.220.2.20 stop\_acquisition()

```
void GpsL1CaPcpsTongAcquisition::stop_acquisition () [override], [virtual]
```

Stop running acquisition.

Implements [AcquisitionInterface](#).

The documentation for this class was generated from the following file:

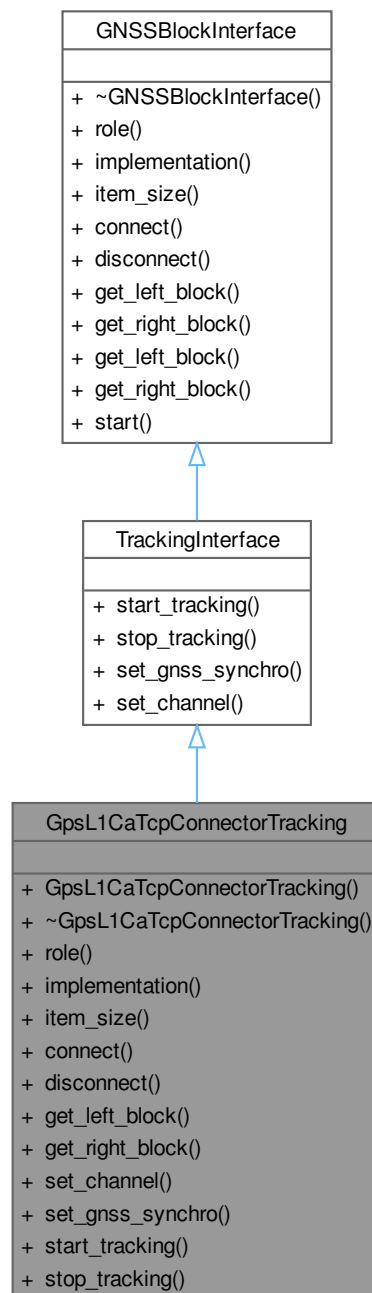
- [gps\\_l1\\_ca\\_pcps\\_tong\\_acquisition.h](#)

## 10.221 GpsL1CaTcpConnectorTracking Class Reference

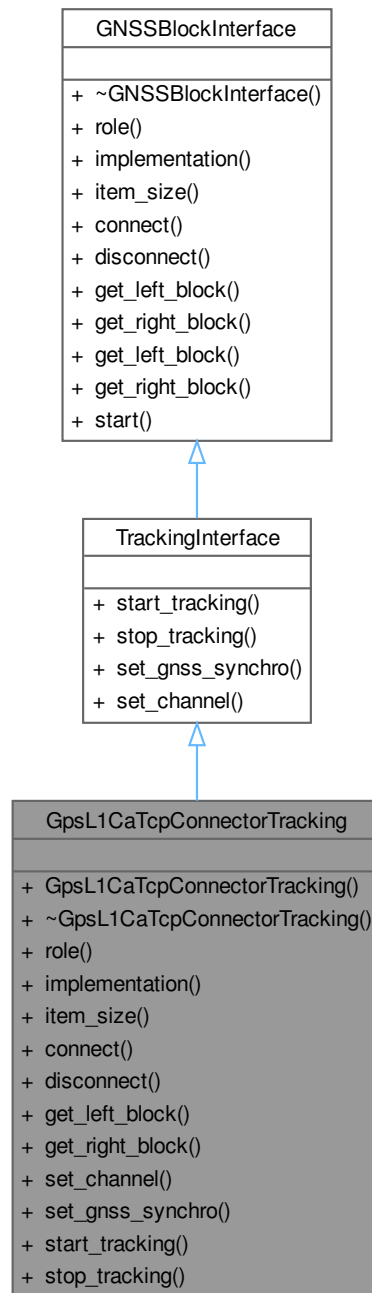
This class implements a code DLL + carrier PLL tracking loop.

```
#include <gps_l1_ca_tcp_connector_tracking.h>
```

Inheritance diagram for GpsL1CaTcpConnectorTracking:



Collaboration diagram for GpsL1CaTcpConnectorTracking:



### Public Member Functions

- **GpsL1CaTcpConnectorTracking** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override
  - Returns "GPS\_L1\_CA\_TCP\_CONNECTOR\_Tracking".*
- size\_t [item\\_size](#) () override



- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_channel](#) (unsigned int channel) override  
*Set tracking channel unique ID.*
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [start\\_tracking](#) () override
- void [stop\\_tracking](#) () override  
*Stop running tracking.*

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()  
*Start the flow of samples if needed.*

### 10.221.1 Detailed Description

This class implements a code DLL + carrier PLL tracking loop.  
Definition at line 42 of file [gps\\_l1\\_ca\\_tcp\\_connector\\_tracking.h](#).

### 10.221.2 Member Function Documentation

#### 10.221.2.1 connect()

```
void GpsL1CaTcpConnectorTracking::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.221.2.2 disconnect()

```
void GpsL1CaTcpConnectorTracking::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.221.2.3 get\_left\_block()

```
gr::basic_block_sptr GpsL1CaTcpConnectorTracking::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.221.2.4 get\_right\_block()

```
gr::basic_block_sptr GpsL1CaTcpConnectorTracking::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.221.2.5 implementation()

```
std::string GpsL1CaTcpConnectorTracking::implementation () [inline], [override], [virtual]
```

Returns "GPS\_L1\_CA\_TCP\_CONNECTOR\_Tracking".  
Implements [GNSSBlockInterface](#).  
Definition at line 59 of file [gps\\_l1\\_ca\\_tcp\\_connector\\_tracking.h](#).

**10.221.2.6 item\_size()**

```
size_t GpsL1CaTcpConnectorTracking::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 64 of file [gps\\_l1\\_ca\\_tcp\\_connector\\_tracking.h](#).

**10.221.2.7 role()**

```
std::string GpsL1CaTcpConnectorTracking::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 53 of file [gps\\_l1\\_ca\\_tcp\\_connector\\_tracking.h](#).

**10.221.2.8 set\_channel()**

```
void GpsL1CaTcpConnectorTracking::set_channel (
    unsigned int channel) [override], [virtual]
```

Set tracking channel unique ID.

Implements [TrackingInterface](#).

**10.221.2.9 set\_gnss\_synchro()**

```
void GpsL1CaTcpConnectorTracking::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [TrackingInterface](#).

**10.221.2.10 start\_tracking()**

```
void GpsL1CaTcpConnectorTracking::start_tracking () [override], [virtual]
```

Implements [TrackingInterface](#).

**10.221.2.11 stop\_tracking()**

```
void GpsL1CaTcpConnectorTracking::stop_tracking () [override], [virtual]
```

Stop running tracking.

Implements [TrackingInterface](#).

The documentation for this class was generated from the following file:

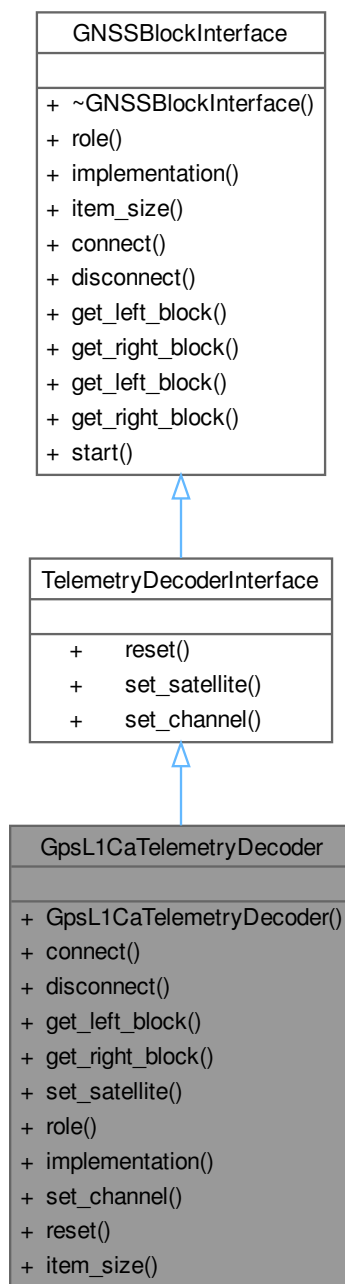
- [gps\\_l1\\_ca\\_tcp\\_connector\\_tracking.h](#)

**10.222 GpsL1CaTelemetryDecoder Class Reference**

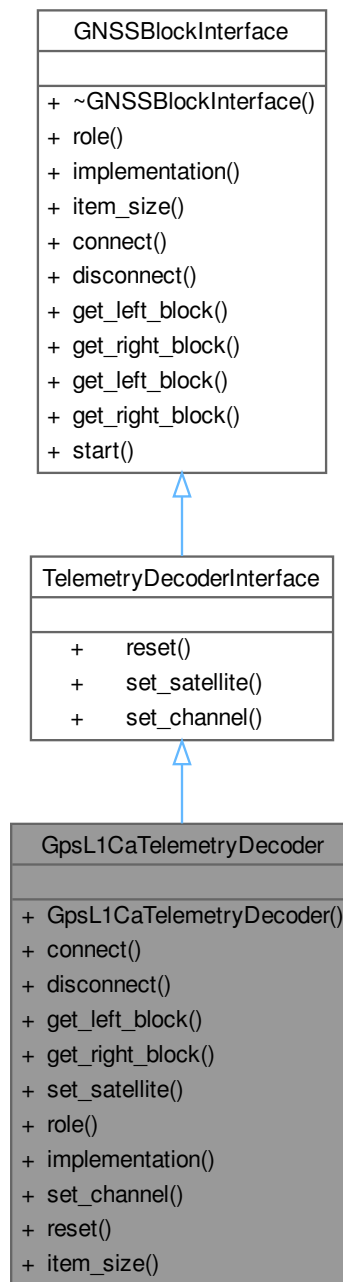
This class implements a NAV data decoder for GPS L1 C/A.

```
#include <gps_l1_ca_telemetry_decoder.h>
```

Inheritance diagram for GpsL1CaTelemetryDecoder:



Collaboration diagram for GpsL1CaTelemetryDecoder:



## Public Member Functions

- **GpsL1CaTelemetryDecoder** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override

- void [set\\_satellite](#) (const [Gnss\\_Satellite](#) &satellite) override
- std::string [role](#) () override
- std::string [implementation](#) () override
- *Returns "GPS\_L1\_CA\_Telemetry\_Decoder".*
- void [set\\_channel](#) (int channel) override
- void [reset](#) () override
- size\_t [item\\_size](#) () override

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

## 10.222.1 Detailed Description

This class implements a NAV data decoder for GPS L1 C/A.  
Definition at line 45 of file [gps\\_l1\\_ca\\_telemetry\\_decoder.h](#).

## 10.222.2 Member Function Documentation

### 10.222.2.1 connect()

```
void GpsL1CaTelemetryDecoder::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.222.2.2 disconnect()

```
void GpsL1CaTelemetryDecoder::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.222.2.3 get\_left\_block()

```
gr::basic_block_sptr GpsL1CaTelemetryDecoder::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.222.2.4 get\_right\_block()

```
gr::basic_block_sptr GpsL1CaTelemetryDecoder::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.222.2.5 implementation()

```
std::string GpsL1CaTelemetryDecoder::implementation () [inline], [override], [virtual]
```

Returns "GPS\_L1\_CA\_Telemetry\_Decoder".

Implements [GNSSBlockInterface](#).

Definition at line 67 of file [gps\\_l1\\_ca\\_telemetry\\_decoder.h](#).

### 10.222.2.6 item\_size()

```
size_t GpsL1CaTelemetryDecoder::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 79 of file [gps\\_l1\\_ca\\_telemetry\\_decoder.h](#).

#### 10.222.2.7 reset()

```
void GpsL1CaTelemetryDecoder::reset () [inline], [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

Definition at line 74 of file [gps\\_l1\\_ca\\_telemetry\\_decoder.h](#).

#### 10.222.2.8 role()

```
std::string GpsL1CaTelemetryDecoder::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 61 of file [gps\\_l1\\_ca\\_telemetry\\_decoder.h](#).

#### 10.222.2.9 set\_channel()

```
void GpsL1CaTelemetryDecoder::set_channel (  
    int channel) [inline], [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

Definition at line 72 of file [gps\\_l1\\_ca\\_telemetry\\_decoder.h](#).

#### 10.222.2.10 set\_satellite()

```
void GpsL1CaTelemetryDecoder::set_satellite (  
    const Gnss\_Satellite & satellite) [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

The documentation for this class was generated from the following file:

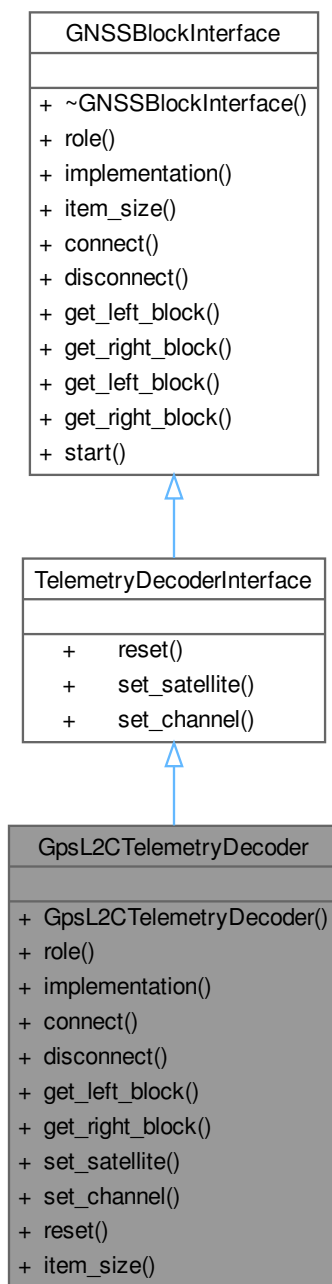
- [gps\\_l1\\_ca\\_telemetry\\_decoder.h](#)

## 10.223 GpsL2CTelemetryDecoder Class Reference

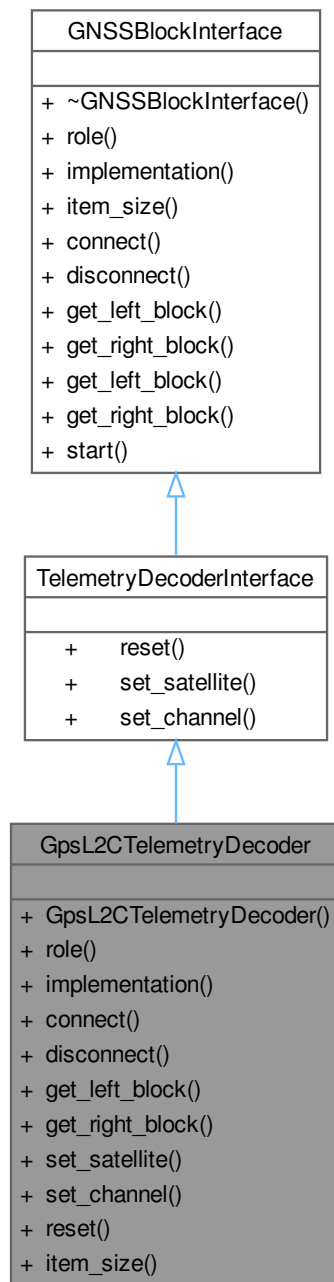
This class implements a NAV data decoder for GPS L2 M.

```
#include <gps_l2c_telemetry_decoder.h>
```

Inheritance diagram for GpsL2CTelemetryDecoder:



Collaboration diagram for GpsL2CTelemetryDecoder:



### Public Member Functions

- **GpsL2CTelemetryDecoder** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override
  - Returns "GPS\_L2C\_Telemetry\_Decoder".*
- void [connect](#) (gr::top\_block\_sptr top\_block) override



- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_satellite](#) (const [Gnss\\_Satellite](#) &satellite) override
- void [set\\_channel](#) (int channel) override
- void [reset](#) () override
- size\_t [item\\_size](#) () override

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

## 10.223.1 Detailed Description

This class implements a NAV data decoder for GPS L2 M.  
Definition at line 43 of file [gps\\_l2c\\_telemetry\\_decoder.h](#).

## 10.223.2 Member Function Documentation

### 10.223.2.1 connect()

```
void GpsL2CTelemetryDecoder::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.223.2.2 disconnect()

```
void GpsL2CTelemetryDecoder::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.223.2.3 get\_left\_block()

```
gr::basic_block_sptr GpsL2CTelemetryDecoder::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.223.2.4 get\_right\_block()

```
gr::basic_block_sptr GpsL2CTelemetryDecoder::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.223.2.5 implementation()

```
std::string GpsL2CTelemetryDecoder::implementation () [inline], [override], [virtual]
```

Returns "GPS\_L2C\_Telemetry\_Decoder".

Implements [GNSSBlockInterface](#).

Definition at line 58 of file [gps\\_l2c\\_telemetry\\_decoder.h](#).

### 10.223.2.6 item\_size()

```
size_t GpsL2CTelemetryDecoder::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 77 of file [gps\\_l2c\\_telemetry\\_decoder.h](#).

#### 10.223.2.7 reset()

```
void GpsL2CTelemetryDecoder::reset () [inline], [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

Definition at line 72 of file [gps\\_l2c\\_telemetry\\_decoder.h](#).

#### 10.223.2.8 role()

```
std::string GpsL2CTelemetryDecoder::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 52 of file [gps\\_l2c\\_telemetry\\_decoder.h](#).

#### 10.223.2.9 set\_channel()

```
void GpsL2CTelemetryDecoder::set_channel (
    int channel) [inline], [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

Definition at line 70 of file [gps\\_l2c\\_telemetry\\_decoder.h](#).

#### 10.223.2.10 set\_satellite()

```
void GpsL2CTelemetryDecoder::set_satellite (
    const Gnss\_Satellite & satellite) [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

The documentation for this class was generated from the following file:

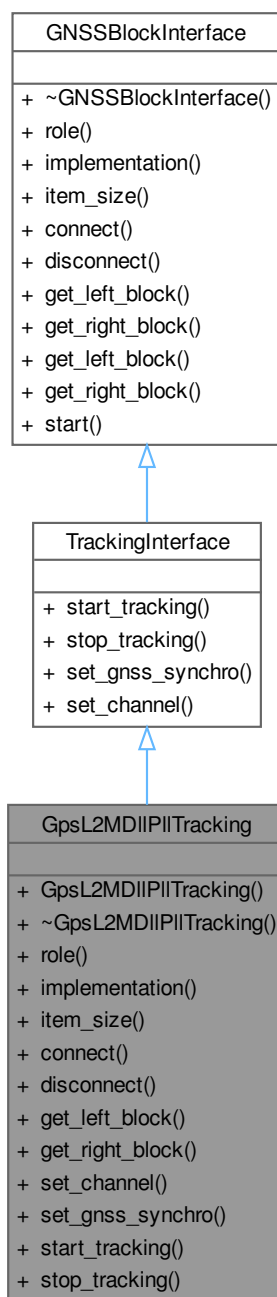
- [gps\\_l2c\\_telemetry\\_decoder.h](#)

## 10.224 GpsL2MDIIPITracking Class Reference

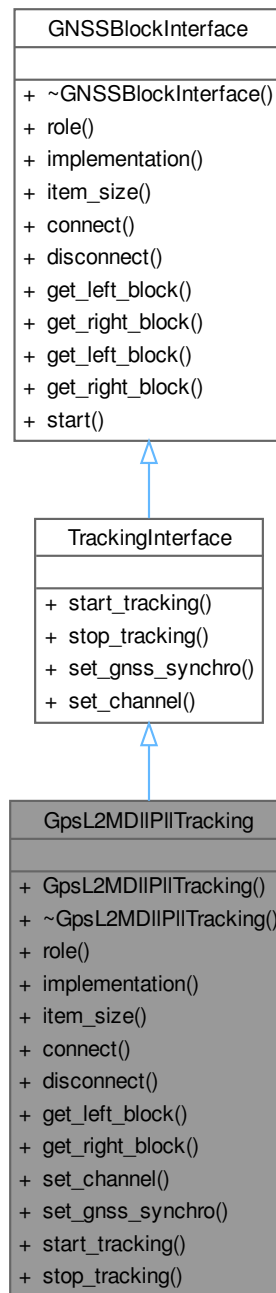
This class implements a code DLL + carrier PLL tracking loop.

```
#include <gps_l2_m_dll_pll_tracking.h>
```

Inheritance diagram for GpsL2MDIIPITracking:



Collaboration diagram for GpsL2MDIIPITracking:



### Public Member Functions

- **GpsL2MDIIPITracking** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in↔\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "GPS\_L2\_M\_DLL\_PLL\_Tracking".
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_channel](#) (unsigned int channel) override  
*Set tracking channel unique ID.*
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [start\\_tracking](#) () override
- void [stop\\_tracking](#) () override  
*Stop running tracking.*

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()  
*Start the flow of samples if needed.*

### 10.224.1 Detailed Description

This class implements a code DLL + carrier PLL tracking loop.

Definition at line 42 of file [gps\\_l2\\_m\\_dll\\_pll\\_tracking.h](#).

### 10.224.2 Member Function Documentation

#### 10.224.2.1 connect()

```
void GpsL2MD11P11Tracking::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.224.2.2 disconnect()

```
void GpsL2MD11P11Tracking::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.224.2.3 get\_left\_block()

```
gr::basic_block_sptr GpsL2MD11P11Tracking::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.224.2.4 get\_right\_block()

```
gr::basic_block_sptr GpsL2MD11P11Tracking::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.224.2.5 implementation()

```
std::string GpsL2MD11P11Tracking::implementation () [inline], [override], [virtual]
```

Returns "GPS\_L2\_M\_DLL\_PLL\_Tracking".

Implements [GNSSBlockInterface](#).

Definition at line 59 of file [gps\\_l2\\_m\\_dll\\_pll\\_tracking.h](#).

**10.224.2.6 item\_size()**

```
size_t GpsL2MD1lP1lTracking::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 64 of file [gps\\_l2\\_m\\_dll\\_pll\\_tracking.h](#).

**10.224.2.7 role()**

```
std::string GpsL2MD1lP1lTracking::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 53 of file [gps\\_l2\\_m\\_dll\\_pll\\_tracking.h](#).

**10.224.2.8 set\_channel()**

```
void GpsL2MD1lP1lTracking::set_channel (
    unsigned int channel) [override], [virtual]
```

Set tracking channel unique ID.

Implements [TrackingInterface](#).

**10.224.2.9 set\_gnss\_synchro()**

```
void GpsL2MD1lP1lTracking::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [TrackingInterface](#).

**10.224.2.10 start\_tracking()**

```
void GpsL2MD1lP1lTracking::start_tracking () [override], [virtual]
```

Implements [TrackingInterface](#).

**10.224.2.11 stop\_tracking()**

```
void GpsL2MD1lP1lTracking::stop_tracking () [override], [virtual]
```

Stop running tracking.

Implements [TrackingInterface](#).

The documentation for this class was generated from the following file:

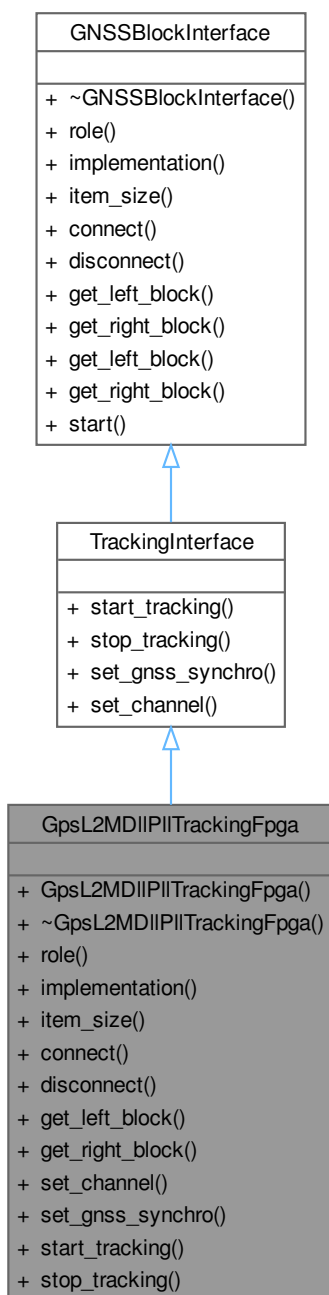
- [gps\\_l2\\_m\\_dll\\_pll\\_tracking.h](#)

**10.225 GpsL2MDIPIlTrackingFpga Class Reference**

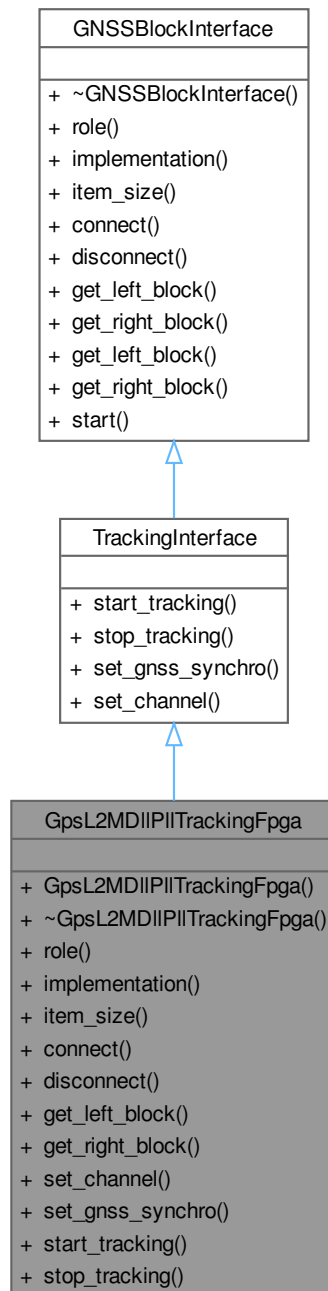
This class implements a code DLL + carrier PLL tracking loop.

```
#include <gps_l2_m_dll_pll_tracking_fpga.h>
```

Inheritance diagram for GpsL2MDIIPITrackingFpga:



Collaboration diagram for GpsL2MDIIPITrackingFpga:



## Public Member Functions

- **GpsL2MDIIPITrackingFpga** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "GPS\_L2\_M\_DLL\_PLL\_Tracking\_FPGA".
- size\_t [item\\_size](#) () override



- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_channel](#) (unsigned int channel) override  
*Set tracking channel unique ID.*
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [start\\_tracking](#) () override
- void [stop\\_tracking](#) () override  
*Stop running tracking.*

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()  
*Start the flow of samples if needed.*

### 10.225.1 Detailed Description

This class implements a code DLL + carrier PLL tracking loop.  
Definition at line 44 of file [gps\\_l2\\_m\\_dll\\_pll\\_tracking\\_fpga.h](#).

### 10.225.2 Member Function Documentation

#### 10.225.2.1 connect()

```
void GpsL2MD11P11TrackingFpga::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.225.2.2 disconnect()

```
void GpsL2MD11P11TrackingFpga::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.225.2.3 get\_left\_block()

```
gr::basic_block_sptr GpsL2MD11P11TrackingFpga::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.225.2.4 get\_right\_block()

```
gr::basic_block_sptr GpsL2MD11P11TrackingFpga::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.225.2.5 implementation()

```
std::string GpsL2MD11P11TrackingFpga::implementation () [inline], [override], [virtual]
```

Returns "GPS\_L2\_M\_DLL\_PLL\_Tracking\_FPGA".  
Implements [GNSSBlockInterface](#).  
Definition at line 61 of file [gps\\_l2\\_m\\_dll\\_pll\\_tracking\\_fpga.h](#).

**10.225.2.6 item\_size()**

```
size_t GpsL2MD11P11TrackingFpga::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 66 of file [gps\\_l2\\_m\\_dll\\_pll\\_tracking\\_fpga.h](#).

**10.225.2.7 role()**

```
std::string GpsL2MD11P11TrackingFpga::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 55 of file [gps\\_l2\\_m\\_dll\\_pll\\_tracking\\_fpga.h](#).

**10.225.2.8 set\_channel()**

```
void GpsL2MD11P11TrackingFpga::set_channel (
    unsigned int channel) [override], [virtual]
```

Set tracking channel unique ID.

Implements [TrackingInterface](#).

**10.225.2.9 set\_gnss\_synchro()**

```
void GpsL2MD11P11TrackingFpga::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [TrackingInterface](#).

**10.225.2.10 start\_tracking()**

```
void GpsL2MD11P11TrackingFpga::start_tracking () [override], [virtual]
```

Implements [TrackingInterface](#).

**10.225.2.11 stop\_tracking()**

```
void GpsL2MD11P11TrackingFpga::stop_tracking () [override], [virtual]
```

Stop running tracking.

Implements [TrackingInterface](#).

The documentation for this class was generated from the following file:

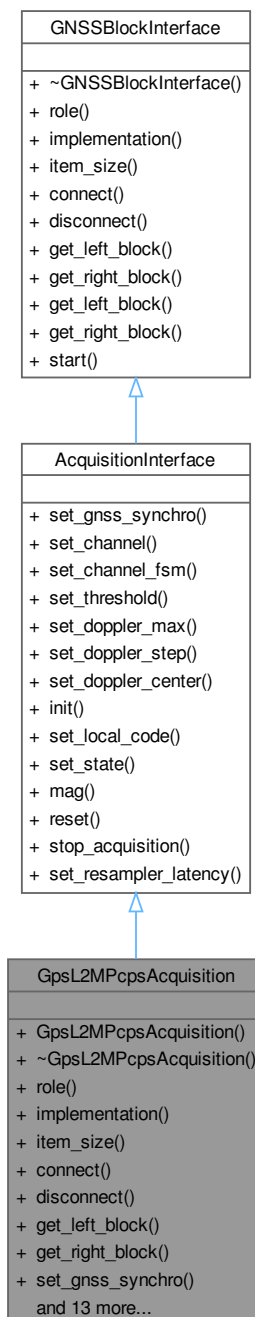
- [gps\\_l2\\_m\\_dll\\_pll\\_tracking\\_fpga.h](#)

**10.226 GpsL2MPcpsAcquisition Class Reference**

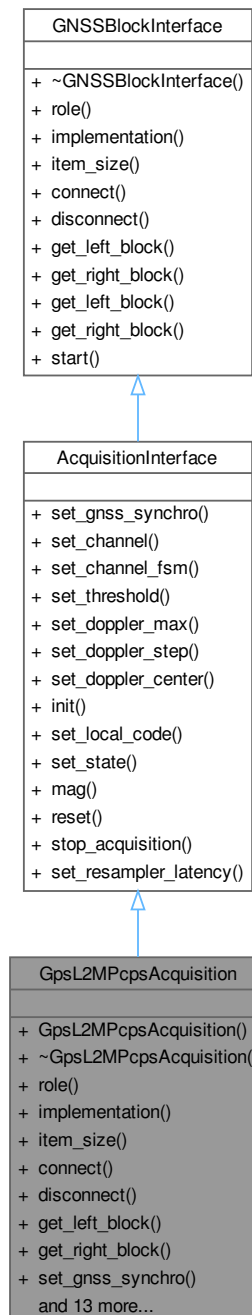
This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L2 M signals.

```
#include <gps_l2_m_pcps_acquisition.h>
```

Inheritance diagram for GpsL2MPcpsAcquisition:



Collaboration diagram for GpsL2MPcpsAcquisition:



## Public Member Functions

- **GpsL2MPcpsAcquisition** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "GPS\_L2\_M\_PCPS\_Acquisition".
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [set\\_channel](#) (unsigned int channel) override  
*Set acquisition channel unique ID.*
- void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm) override  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold) override  
*Set statistics threshold of PCPS algorithm.*
- void [set\\_doppler\\_max](#) (unsigned int doppler\_max) override  
*Set maximum Doppler off grid search.*
- void [set\\_doppler\\_step](#) (unsigned int doppler\_step) override  
*Set Doppler steps for the grid search.*
- void [set\\_doppler\\_center](#) (int doppler\_center) override  
*Set Doppler center for the grid search.*
- void [init](#) () override  
*Initializes acquisition algorithm.*
- void [set\\_local\\_code](#) () override  
*Sets local code for GPS L2/M PCPS acquisition algorithm.*
- signed int [mag](#) () override  
*Returns the maximum peak of grid search.*
- void [reset](#) () override  
*Restart acquisition algorithm.*
- void [set\\_state](#) (int state) override  
*If state = 1, it forces the block to start acquiring from the first sample.*
- void [stop\\_acquisition](#) () override  
*Stop running acquisition.*
- void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples) override  
*Sets the resampler latency to account it in the acquisition code delay estimation.*

### Public Member Functions inherited from [AcquisitionInterface](#)

- virtual void [set\\_doppler\\_center](#) (int doppler\_center \_\_attribute\_\_((unused)))

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

#### 10.226.1 Detailed Description

This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L2 M signals.  
 Definition at line 46 of file [gps\\_l2\\_m\\_pcps\\_acquisition.h](#).

## 10.226.2 Member Function Documentation

### 10.226.2.1 connect()

```
void GpsL2MPcpsAcquisition::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.226.2.2 disconnect()

```
void GpsL2MPcpsAcquisition::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.226.2.3 get\_left\_block()

```
gr::basic_block_sptr GpsL2MPcpsAcquisition::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.226.2.4 get\_right\_block()

```
gr::basic_block_sptr GpsL2MPcpsAcquisition::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.226.2.5 implementation()

```
std::string GpsL2MPcpsAcquisition::implementation () [inline], [override], [virtual]
```

Returns "GPS\_L2\_M\_PCPS\_Acquisition".

Implements [GNSSBlockInterface](#).

Definition at line 65 of file [gps\\_l2\\_m\\_pcps\\_acquisition.h](#).

### 10.226.2.6 init()

```
void GpsL2MPcpsAcquisition::init () [override], [virtual]
```

Initializes acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.226.2.7 item\_size()

```
size_t GpsL2MPcpsAcquisition::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 70 of file [gps\\_l2\\_m\\_pcps\\_acquisition.h](#).

### 10.226.2.8 mag()

```
signed int GpsL2MPcpsAcquisition::mag () [override], [virtual]
```

Returns the maximum peak of grid search.

Implements [AcquisitionInterface](#).

### 10.226.2.9 reset()

```
void GpsL2MPcpsAcquisition::reset () [override], [virtual]
```

Restart acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.226.2.10 role()

```
std::string GpsL2MPcpsAcquisition::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 57 of file [gps\\_l2\\_m\\_pcps\\_acquisition.h](#).

### 10.226.2.11 set\_channel()

```
void GpsL2MPcpsAcquisition::set_channel (
    unsigned int channel) [inline], [override], [virtual]
```

Set acquisition channel unique ID.

Implements [AcquisitionInterface](#).

Definition at line 90 of file [gps\\_l2\\_m\\_pcps\\_acquisition.h](#).

### 10.226.2.12 set\_channel\_fsm()

```
void GpsL2MPcpsAcquisition::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline], [override], [virtual]
```

Set channel fsm associated to this acquisition instance.

Implements [AcquisitionInterface](#).

Definition at line 99 of file [gps\\_l2\\_m\\_pcps\\_acquisition.h](#).

### 10.226.2.13 set\_doppler\_center()

```
void GpsL2MPcpsAcquisition::set_doppler_center (
    int doppler_center) [override]
```

Set Doppler center for the grid search.

### 10.226.2.14 set\_doppler\_max()

```
void GpsL2MPcpsAcquisition::set_doppler_max (
    unsigned int doppler_max) [override], [virtual]
```

Set maximum Doppler off grid search.

Implements [AcquisitionInterface](#).

### 10.226.2.15 set\_doppler\_step()

```
void GpsL2MPcpsAcquisition::set_doppler_step (
    unsigned int doppler_step) [override], [virtual]
```

Set Doppler steps for the grid search.

Implements [AcquisitionInterface](#).

### 10.226.2.16 set\_gnss\_synchro()

```
void GpsL2MPcpsAcquisition::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [AcquisitionInterface](#).

### 10.226.2.17 set\_local\_code()

```
void GpsL2MPcpsAcquisition::set_local_code () [override], [virtual]
```

Sets local code for GPS L2/M PCPS acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.226.2.18 set\_resampler\_latency()

```
void GpsL2MPcpsAcquisition::set_resampler_latency (
    uint32_t latency_samples) [override], [virtual]
```

Sets the resampler latency to account it in the acquisition code delay estimation.

Implements [AcquisitionInterface](#).

#### 10.226.2.19 `set_state()`

```
void GpsL2MPcpsAcquisition::set_state (  
    int state) [override], [virtual]
```

If state = 1, it forces the block to start acquiring from the first sample.

Implements [AcquisitionInterface](#).

#### 10.226.2.20 `set_threshold()`

```
void GpsL2MPcpsAcquisition::set_threshold (  
    float threshold) [override], [virtual]
```

Set statistics threshold of PCPS algorithm.

Implements [AcquisitionInterface](#).

#### 10.226.2.21 `stop_acquisition()`

```
void GpsL2MPcpsAcquisition::stop_acquisition () [override], [virtual]
```

Stop running acquisition.

Implements [AcquisitionInterface](#).

The documentation for this class was generated from the following file:

- [gps\\_l2\\_m\\_pcps\\_acquisition.h](#)

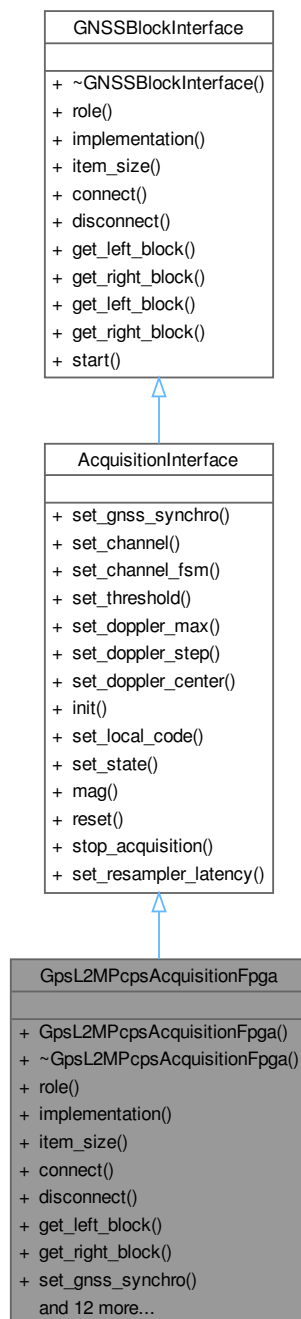
### 10.227 GpsL2MPcpsAcquisitionFpga Class Reference

This class adapts a PCPS acquisition block off-loaded on an FPGA to an [AcquisitionInterface](#) for GPS L2 M signals.

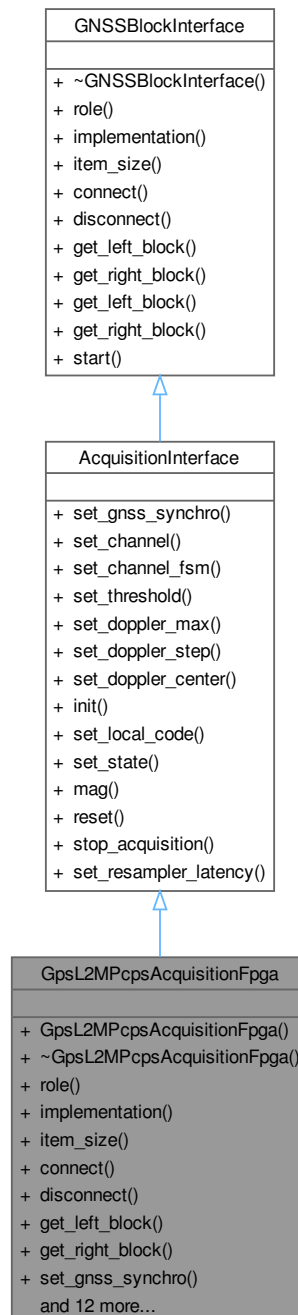
```
#include <gps_l2_m_pcps_acquisition_fpga.h>
```



Inheritance diagram for GpsL2MPcpsAcquisitionFpga:



Collaboration diagram for GpsL2MPcpsAcquisitionFpga:



## Public Member Functions

- **GpsL2MPcpsAcquisitionFpga** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override
  - Returns "GPS\_L2\_M\_PCPS\_Acquisition\_FPGA".*
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [set\\_channel](#) (unsigned int channel) override  
*Set acquisition channel unique ID.*
- void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm) override  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold) override  
*Set statistics threshold of PCPS algorithm.*
- void [set\\_doppler\\_max](#) (unsigned int doppler\_max) override  
*Set maximum Doppler off grid search.*
- void [set\\_doppler\\_step](#) (unsigned int doppler\_step) override  
*Set Doppler steps for the grid search.*
- void [init](#) () override  
*Initializes acquisition algorithm.*
- void [set\\_local\\_code](#) () override  
*Sets local code for GPS L2/M PCPS acquisition algorithm.*
- signed int [mag](#) () override  
*Returns the maximum peak of grid search.*
- void [reset](#) () override  
*Restart acquisition algorithm.*
- void [set\\_state](#) (int state) override  
*If state = 1, it forces the block to start acquiring from the first sample.*
- void [stop\\_acquisition](#) () override  
*Stop running acquisition.*
- void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples \_\_attribute\_\_((unused))) override

### Public Member Functions inherited from [AcquisitionInterface](#)

- virtual void [set\\_doppler\\_center](#) (int doppler\_center \_\_attribute\_\_((unused)))
- virtual void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples)=0

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

## 10.227.1 Detailed Description

This class adapts a PCPS acquisition block off-loaded on an FPGA to an [AcquisitionInterface](#) for GPS L2 M signals. Definition at line 46 of file [gps\\_l2\\_m\\_pcps\\_acquisition\\_fpga.h](#).

## 10.227.2 Member Function Documentation

### 10.227.2.1 [connect\(\)](#)

```
void GpsL2MPcpsAcquisitionFpga::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.227.2.2 disconnect()

```
void GpsL2MPcpsAcquisitionFpga::disconnect (  
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.227.2.3 get\_left\_block()

```
gr::basic_block_sptr GpsL2MPcpsAcquisitionFpga::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.227.2.4 get\_right\_block()

```
gr::basic_block_sptr GpsL2MPcpsAcquisitionFpga::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.227.2.5 implementation()

```
std::string GpsL2MPcpsAcquisitionFpga::implementation () [inline], [override], [virtual]
```

Returns "GPS\_L2\_M\_PCPS\_Acquisition\_FPGA".

Implements [GNSSBlockInterface](#).

Definition at line 65 of file [gps\\_l2\\_m\\_pcps\\_acquisition\\_fpga.h](#).

### 10.227.2.6 init()

```
void GpsL2MPcpsAcquisitionFpga::init () [override], [virtual]
```

Initializes acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.227.2.7 item\_size()

```
size_t GpsL2MPcpsAcquisitionFpga::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 70 of file [gps\\_l2\\_m\\_pcps\\_acquisition\\_fpga.h](#).

### 10.227.2.8 mag()

```
signed int GpsL2MPcpsAcquisitionFpga::mag () [override], [virtual]
```

Returns the maximum peak of grid search.

Implements [AcquisitionInterface](#).

### 10.227.2.9 reset()

```
void GpsL2MPcpsAcquisitionFpga::reset () [override], [virtual]
```

Restart acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.227.2.10 role()

```
std::string GpsL2MPcpsAcquisitionFpga::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 57 of file [gps\\_l2\\_m\\_pcps\\_acquisition\\_fpga.h](#).

### 10.227.2.11 set\_channel()

```
void GpsL2MPcpsAcquisitionFpga::set_channel (  
    unsigned int channel) [inline], [override], [virtual]
```

Set acquisition channel unique ID.

Implements [AcquisitionInterface](#).

Definition at line 90 of file [gps\\_l2\\_m\\_pcps\\_acquisition\\_fpga.h](#).

**10.227.2.12 set\_channel\_fsm()**

```
void GpsL2MPcpsAcquisitionFpga::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline], [override], [virtual]
```

Set channel fsm associated to this acquisition instance.

Implements [AcquisitionInterface](#).

Definition at line 99 of file [gps\\_l2\\_m\\_pcps\\_acquisition\\_fpga.h](#).

**10.227.2.13 set\_doppler\_max()**

```
void GpsL2MPcpsAcquisitionFpga::set_doppler_max (
    unsigned int doppler_max) [override], [virtual]
```

Set maximum Doppler off grid search.

Implements [AcquisitionInterface](#).

**10.227.2.14 set\_doppler\_step()**

```
void GpsL2MPcpsAcquisitionFpga::set_doppler_step (
    unsigned int doppler_step) [override], [virtual]
```

Set Doppler steps for the grid search.

Implements [AcquisitionInterface](#).

**10.227.2.15 set\_gnss\_synchro()**

```
void GpsL2MPcpsAcquisitionFpga::set_gnss_synchro (
    Gnss\_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [AcquisitionInterface](#).

**10.227.2.16 set\_local\_code()**

```
void GpsL2MPcpsAcquisitionFpga::set_local_code () [override], [virtual]
```

Sets local code for GPS L2/M PCPS acquisition algorithm.

Implements [AcquisitionInterface](#).

**10.227.2.17 set\_resampler\_latency()**

```
void GpsL2MPcpsAcquisitionFpga::set_resampler_latency (
    uint32_t latency_samples __attribute__((unused))) [inline], [override]
```

Definition at line 150 of file [gps\\_l2\\_m\\_pcps\\_acquisition\\_fpga.h](#).

**10.227.2.18 set\_state()**

```
void GpsL2MPcpsAcquisitionFpga::set_state (
    int state) [override], [virtual]
```

If state = 1, it forces the block to start acquiring from the first sample.

Implements [AcquisitionInterface](#).

**10.227.2.19 set\_threshold()**

```
void GpsL2MPcpsAcquisitionFpga::set_threshold (
    float threshold) [override], [virtual]
```

Set statistics threshold of PCPS algorithm.

Implements [AcquisitionInterface](#).

**10.227.2.20 stop\_acquisition()**

```
void GpsL2MPcpsAcquisitionFpga::stop_acquisition () [override], [virtual]
```

Stop running acquisition.

Implements [AcquisitionInterface](#).

The documentation for this class was generated from the following file:

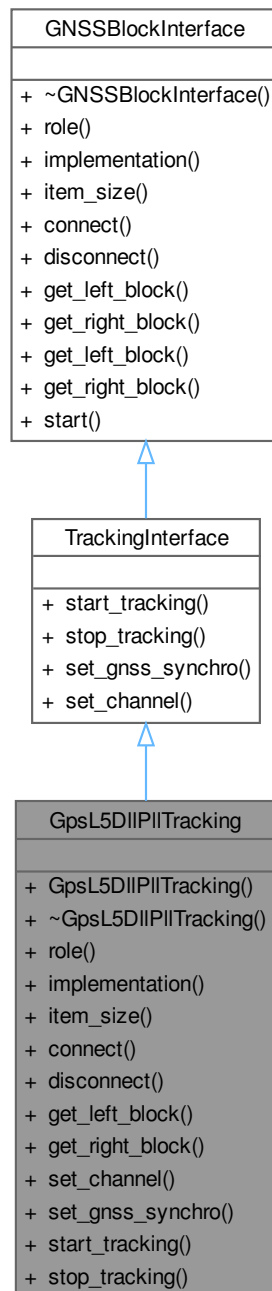
- [gps\\_l2\\_m\\_pcps\\_acquisition\\_fpga.h](#)

## 10.228 GpsL5DIIPITracking Class Reference

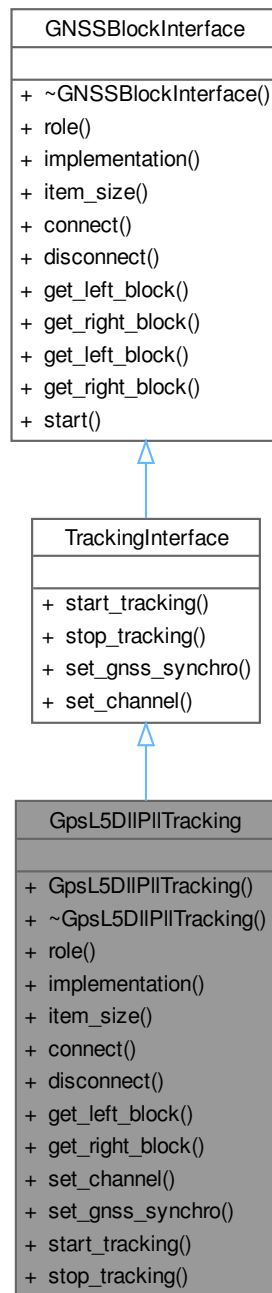
This class implements a code DLL + carrier PLL tracking loop.

```
#include <gps_l5_dll_pll_tracking.h>
```

Inheritance diagram for GpsL5DIIPITracking:



Collaboration diagram for GpsL5DIIPITracking:



### Public Member Functions

- **GpsL5DIIPITracking** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in↔  
\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "GPS\_L5\_DLL\_PLL\_Tracking".
- size\_t [item\\_size](#) () override

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_channel](#) (unsigned int channel) override
 

*Set tracking channel unique ID.*
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override
 

*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [start\\_tracking](#) () override
- void [stop\\_tracking](#) () override
 

*Stop running tracking.*

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()
 

*Start the flow of samples if needed.*

### 10.228.1 Detailed Description

This class implements a code DLL + carrier PLL tracking loop.  
 Definition at line 41 of file [gps\\_l5\\_dll\\_pll\\_tracking.h](#).

### 10.228.2 Member Function Documentation

#### 10.228.2.1 connect()

```
void GpsL5DllPllTracking::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.228.2.2 disconnect()

```
void GpsL5DllPllTracking::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.228.2.3 get\_left\_block()

```
gr::basic_block_sptr GpsL5DllPllTracking::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.228.2.4 get\_right\_block()

```
gr::basic_block_sptr GpsL5DllPllTracking::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.228.2.5 implementation()

```
std::string GpsL5DllPllTracking::implementation () [inline], [override], [virtual]
```

Returns "GPS\_L5\_DLL\_PLL\_Tracking".  
 Implements [GNSSBlockInterface](#).  
 Definition at line 58 of file [gps\\_l5\\_dll\\_pll\\_tracking.h](#).



**10.228.2.6 item\_size()**

```
size_t GpsL5DlIPllTracking::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 63 of file [gps\\_l5\\_dll\\_pll\\_tracking.h](#).

**10.228.2.7 role()**

```
std::string GpsL5DlIPllTracking::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 52 of file [gps\\_l5\\_dll\\_pll\\_tracking.h](#).

**10.228.2.8 set\_channel()**

```
void GpsL5DlIPllTracking::set_channel (
    unsigned int channel) [override], [virtual]
```

Set tracking channel unique ID.

Implements [TrackingInterface](#).

**10.228.2.9 set\_gnss\_synchro()**

```
void GpsL5DlIPllTracking::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [TrackingInterface](#).

**10.228.2.10 start\_tracking()**

```
void GpsL5DlIPllTracking::start_tracking () [override], [virtual]
```

Implements [TrackingInterface](#).

**10.228.2.11 stop\_tracking()**

```
void GpsL5DlIPllTracking::stop_tracking () [override], [virtual]
```

Stop running tracking.

Implements [TrackingInterface](#).

The documentation for this class was generated from the following file:

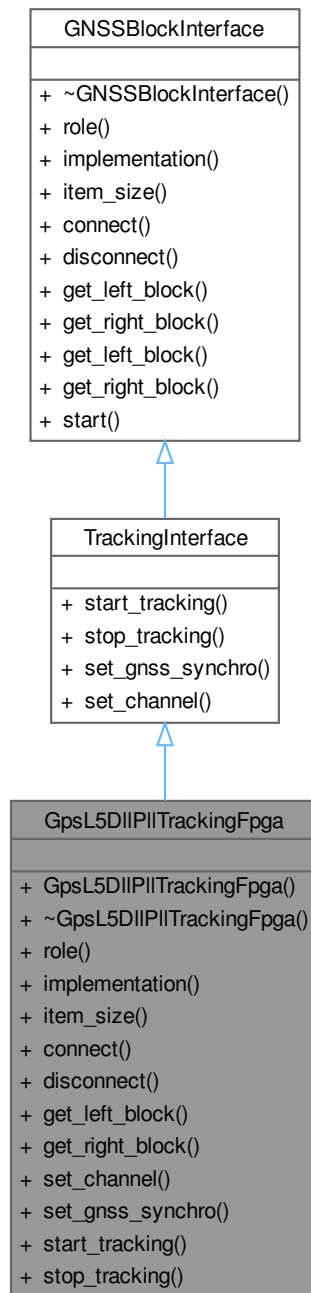
- [gps\\_l5\\_dll\\_pll\\_tracking.h](#)

**10.229 GpsL5DlIPllTrackingFpga Class Reference**

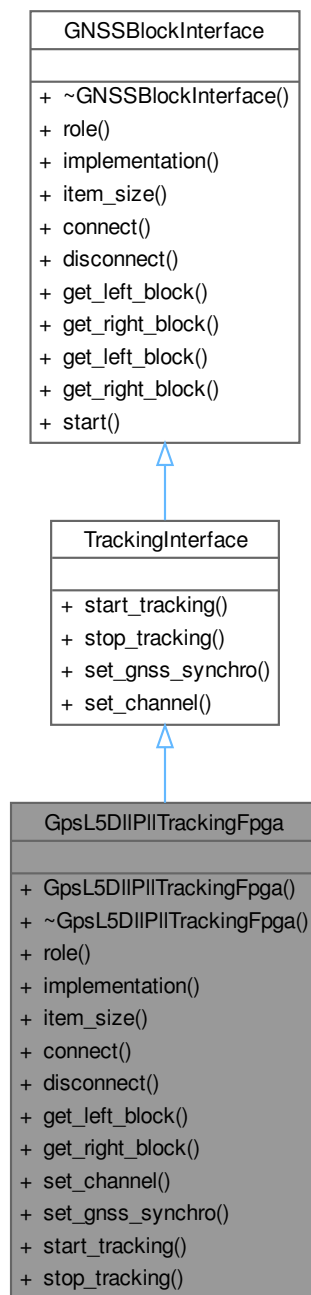
This class implements a code DLL + carrier PLL tracking loop.

```
#include <gps_l5_dll_pll_tracking_fpga.h>
```

Inheritance diagram for GpsL5DIPIITrackingFpga:



Collaboration diagram for GpsL5DIIPITrackingFpga:



## Public Member Functions

- [GpsL5DIIPITrackingFpga](#) (const [ConfigurationInterface](#) \*configuration, const std::string &[role](#), unsigned int in\_streams, unsigned int out\_streams)  
*Constructor.*
- virtual [~GpsL5DIIPITrackingFpga](#) ()  
*Destructor.*
- std::string [role](#) () override

- Role.*
- `std::string implementation ()` override  
*Returns "GPS\_L5\_DLL\_PLL\_Tracking\_FPGA".*
- `size_t item_size ()` override  
*Returns size of `lv_16sc_t`.*
- `void connect (gr::top_block_sptr top_block)` override  
*Connect.*
- `void disconnect (gr::top_block_sptr top_block)` override  
*Disconnect.*
- `gr::basic_block_sptr get_left_block ()` override  
*Get left block.*
- `gr::basic_block_sptr get_right_block ()` override  
*Get right block.*
- `void set_channel (unsigned int channel)` override  
*Set tracking channel unique ID.*
- `void set_gnss_synchro (Gnss_Synchro *p_gnss_synchro)` override  
*Set acquisition/tracking common `Gnss_Synchro` object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- `void start_tracking ()` override  
*Start the tracking process in the FPGA.*
- `void stop_tracking ()` override  
*Stop the tracking process in the FPGA.*

## Public Member Functions inherited from GNSSBlockInterface

- virtual `gr::basic_block_sptr get_left_block (int RF_channel)`
- virtual `gr::basic_block_sptr get_right_block (int RF_channel)`
- virtual `void start ()`  
*Start the flow of samples if needed.*

### 10.229.1 Detailed Description

This class implements a code DLL + carrier PLL tracking loop.  
Definition at line 42 of file [gps\\_l5\\_dll\\_pll\\_tracking\\_fpga.h](#).

### 10.229.2 Constructor & Destructor Documentation

#### 10.229.2.1 GpsL5DlIPllTrackingFpga()

```
GpsL5DlIPllTrackingFpga::GpsL5DlIPllTrackingFpga (
    const ConfigurationInterface * configuration,
    const std::string & role,
    unsigned int in_streams,
    unsigned int out_streams)
```

Constructor.

References [role\(\)](#).

Here is the call graph for this function:



**10.229.2.2 ~GpsL5DlIPllTrackingFpga()**

virtual GpsL5DlIPllTrackingFpga::~~GpsL5DlIPllTrackingFpga () [virtual]  
Destructor.

**10.229.3 Member Function Documentation****10.229.3.1 connect()**

```
void GpsL5DlIPllTrackingFpga::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Connect.

Implements [GNSSBlockInterface](#).

**10.229.3.2 disconnect()**

```
void GpsL5DlIPllTrackingFpga::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Disconnect.

Implements [GNSSBlockInterface](#).

**10.229.3.3 get\_left\_block()**

```
gr::basic_block_sptr GpsL5DlIPllTrackingFpga::get_left_block () [override], [virtual]
```

Get left block.

Implements [GNSSBlockInterface](#).

**10.229.3.4 get\_right\_block()**

```
gr::basic_block_sptr GpsL5DlIPllTrackingFpga::get_right_block () [override], [virtual]
```

Get right block.

Implements [GNSSBlockInterface](#).

**10.229.3.5 implementation()**

```
std::string GpsL5DlIPllTrackingFpga::implementation () [inline], [override], [virtual]
```

Returns "GPS\_L5\_DLL\_PLL\_Tracking\_FPGA".

Implements [GNSSBlockInterface](#).

Definition at line 70 of file [gps\\_l5\\_dll\\_pll\\_tracking\\_fpga.h](#).

**10.229.3.6 item\_size()**

```
size_t GpsL5DlIPllTrackingFpga::item_size () [inline], [override], [virtual]
```

Returns size of lv\_16sc\_t.

Implements [GNSSBlockInterface](#).

Definition at line 78 of file [gps\\_l5\\_dll\\_pll\\_tracking\\_fpga.h](#).

**10.229.3.7 role()**

```
std::string GpsL5DlIPllTrackingFpga::role () [inline], [override], [virtual]
```

Role.

Implements [GNSSBlockInterface](#).

Definition at line 62 of file [gps\\_l5\\_dll\\_pll\\_tracking\\_fpga.h](#).

Referenced by [GpsL5DlIPllTrackingFpga\(\)](#).

**10.229.3.8 set\_channel()**

```
void GpsL5DlIPllTrackingFpga::set_channel (
    unsigned int channel) [override], [virtual]
```

Set tracking channel unique ID.

Implements [TrackingInterface](#).

### 10.229.3.9 set\_gnss\_synchro()

```
void GpsL5D1lP1lTrackingFpga::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [TrackingInterface](#).

### 10.229.3.10 start\_tracking()

```
void GpsL5D1lP1lTrackingFpga::start_tracking () [override], [virtual]
```

Start the tracking process in the FPGA.

Implements [TrackingInterface](#).

### 10.229.3.11 stop\_tracking()

```
void GpsL5D1lP1lTrackingFpga::stop_tracking () [override], [virtual]
```

Stop the tracking process in the FPGA.

Implements [TrackingInterface](#).

The documentation for this class was generated from the following file:

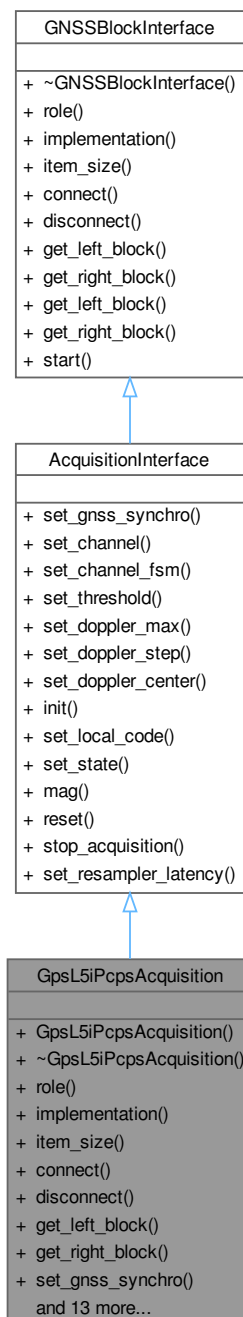
- [gps\\_l5\\_dll\\_pll\\_tracking\\_fpga.h](#)

## 10.230 GpsL5iPcpsAcquisition Class Reference

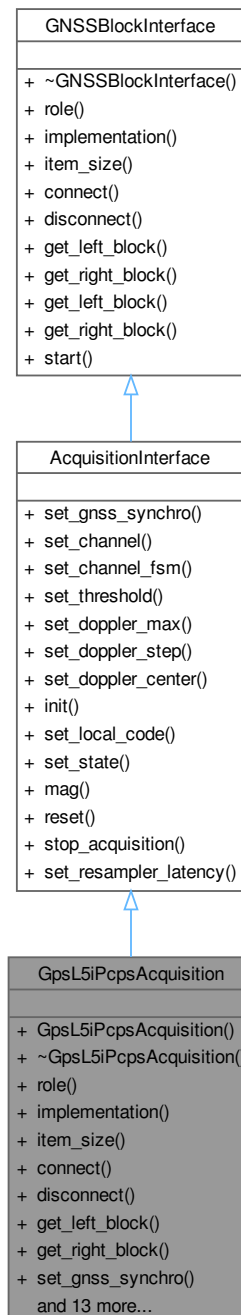
This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L5i signals.

```
#include <gps_l5i_pcps_acquisition.h>
```

Inheritance diagram for GpsL5iPcpsAcquisition:



Collaboration diagram for GpsL5iPcpsAcquisition:



## Public Member Functions

- **GpsL5iPcpsAcquisition** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "GPS\_L5i\_PCPS\_Acquisition".
- size\_t [item\\_size](#) () override



- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [set\\_channel](#) (unsigned int channel) override  
*Set acquisition channel unique ID.*
- void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm) override  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold) override  
*Set statistics threshold of PCPS algorithm.*
- void [set\\_doppler\\_max](#) (unsigned int doppler\_max) override  
*Set maximum Doppler off grid search.*
- void [set\\_doppler\\_step](#) (unsigned int doppler\_step) override  
*Set Doppler steps for the grid search.*
- void [set\\_doppler\\_center](#) (int doppler\_center) override  
*Set Doppler center for the grid search.*
- void [init](#) () override  
*Initializes acquisition algorithm.*
- void [set\\_local\\_code](#) () override  
*Sets local code for GPS L2/M PCPS acquisition algorithm.*
- signed int [mag](#) () override  
*Returns the maximum peak of grid search.*
- void [reset](#) () override  
*Restart acquisition algorithm.*
- void [set\\_state](#) (int state) override  
*If state = 1, it forces the block to start acquiring from the first sample.*
- void [stop\\_acquisition](#) () override  
*Stop running acquisition.*
- void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples) override  
*Sets the resampler latency to account it in the acquisition code delay estimation.*

### Public Member Functions inherited from [AcquisitionInterface](#)

- virtual void [set\\_doppler\\_center](#) (int doppler\_center \_\_attribute\_\_((unused)))

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

#### 10.230.1 Detailed Description

This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L5i signals.  
 Definition at line 45 of file [gps\\_l5i\\_pcps\\_acquisition.h](#).

## 10.230.2 Member Function Documentation

### 10.230.2.1 connect()

```
void GpsL5iPcpsAcquisition::connect (  
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.230.2.2 disconnect()

```
void GpsL5iPcpsAcquisition::disconnect (  
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.230.2.3 get\_left\_block()

```
gr::basic_block_sptr GpsL5iPcpsAcquisition::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.230.2.4 get\_right\_block()

```
gr::basic_block_sptr GpsL5iPcpsAcquisition::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.230.2.5 implementation()

```
std::string GpsL5iPcpsAcquisition::implementation () [inline], [override], [virtual]
```

Returns "GPS\_L5i\_PCPS\_Acquisition".

Implements [GNSSBlockInterface](#).

Definition at line 64 of file [gps\\_l5i\\_pcps\\_acquisition.h](#).

### 10.230.2.6 init()

```
void GpsL5iPcpsAcquisition::init () [override], [virtual]
```

Initializes acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.230.2.7 item\_size()

```
size_t GpsL5iPcpsAcquisition::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 69 of file [gps\\_l5i\\_pcps\\_acquisition.h](#).

### 10.230.2.8 mag()

```
signed int GpsL5iPcpsAcquisition::mag () [override], [virtual]
```

Returns the maximum peak of grid search.

Implements [AcquisitionInterface](#).

### 10.230.2.9 reset()

```
void GpsL5iPcpsAcquisition::reset () [override], [virtual]
```

Restart acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.230.2.10 role()

```
std::string GpsL5iPcpsAcquisition::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 56 of file [gps\\_l5i\\_pcps\\_acquisition.h](#).

### 10.230.2.11 set\_channel()

```
void GpsL5iPcpsAcquisition::set_channel (
    unsigned int channel) [inline], [override], [virtual]
```

Set acquisition channel unique ID.

Implements [AcquisitionInterface](#).

Definition at line 89 of file [gps\\_l5i\\_pcps\\_acquisition.h](#).

### 10.230.2.12 set\_channel\_fsm()

```
void GpsL5iPcpsAcquisition::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline], [override], [virtual]
```

Set channel fsm associated to this acquisition instance.

Implements [AcquisitionInterface](#).

Definition at line 98 of file [gps\\_l5i\\_pcps\\_acquisition.h](#).

### 10.230.2.13 set\_doppler\_center()

```
void GpsL5iPcpsAcquisition::set_doppler_center (
    int doppler_center) [override]
```

Set Doppler center for the grid search.

### 10.230.2.14 set\_doppler\_max()

```
void GpsL5iPcpsAcquisition::set_doppler_max (
    unsigned int doppler_max) [override], [virtual]
```

Set maximum Doppler off grid search.

Implements [AcquisitionInterface](#).

### 10.230.2.15 set\_doppler\_step()

```
void GpsL5iPcpsAcquisition::set_doppler_step (
    unsigned int doppler_step) [override], [virtual]
```

Set Doppler steps for the grid search.

Implements [AcquisitionInterface](#).

### 10.230.2.16 set\_gnss\_synchro()

```
void GpsL5iPcpsAcquisition::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [AcquisitionInterface](#).

### 10.230.2.17 set\_local\_code()

```
void GpsL5iPcpsAcquisition::set_local_code () [override], [virtual]
```

Sets local code for GPS L2/M PCPS acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.230.2.18 set\_resampler\_latency()

```
void GpsL5iPcpsAcquisition::set_resampler_latency (
    uint32_t latency_samples) [override], [virtual]
```

Sets the resampler latency to account it in the acquisition code delay estimation.

Implements [AcquisitionInterface](#).

#### 10.230.2.19 `set_state()`

```
void GpsL5iPcpsAcquisition::set_state (  
    int state) [override], [virtual]
```

If state = 1, it forces the block to start acquiring from the first sample.

Implements [AcquisitionInterface](#).

#### 10.230.2.20 `set_threshold()`

```
void GpsL5iPcpsAcquisition::set_threshold (  
    float threshold) [override], [virtual]
```

Set statistics threshold of PCPS algorithm.

Implements [AcquisitionInterface](#).

#### 10.230.2.21 `stop_acquisition()`

```
void GpsL5iPcpsAcquisition::stop_acquisition () [override], [virtual]
```

Stop running acquisition.

Implements [AcquisitionInterface](#).

The documentation for this class was generated from the following file:

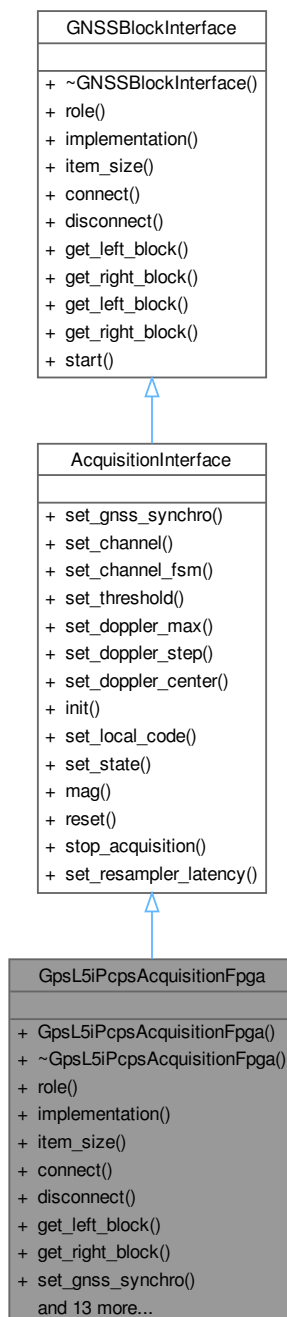
- [gps\\_l5i\\_pcps\\_acquisition.h](#)

### 10.231 `GpsL5iPcpsAcquisitionFpga` Class Reference

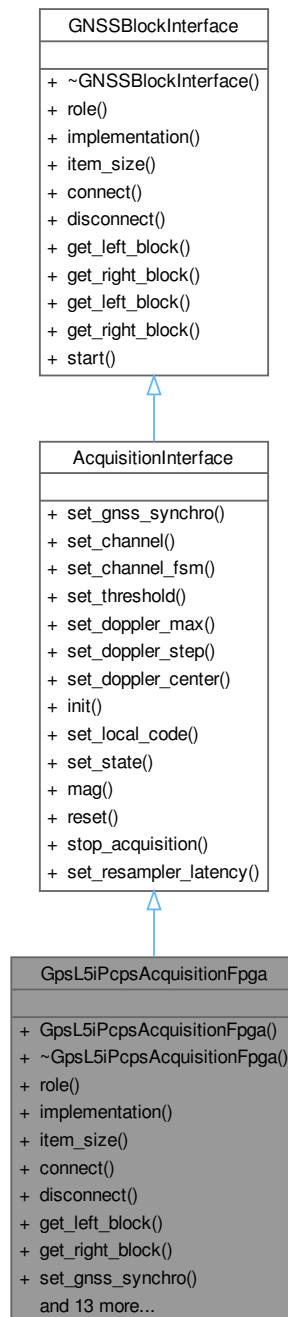
This class adapts a PCPS acquisition block off-loaded on an FPGA to an [AcquisitionInterface](#) for GPS L5i signals.

```
#include <gps_l5i_pcps_acquisition_fpga.h>
```

Inheritance diagram for GpsL5iPcpsAcquisitionFpga:



Collaboration diagram for GpsL5iPcpsAcquisitionFpga:



## Public Member Functions

- [GpsL5iPcpsAcquisitionFpga](#) (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)  
*Constructor.*
- [~GpsL5iPcpsAcquisitionFpga](#) ()=default  
*Destructor.*
- std::string [role](#) () override

*Role.*

- std::string [implementation](#) () override  
*Returns "GPS\_L5i\_PCPS\_Acquisition\_FPGA".*
- size\_t [item\\_size](#) () override  
*Returns size of lv\_16sc\_t.*
- void [connect](#) (gr::top\_block\_sptr top\_block) override  
*Connect.*
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override  
*Disconnect.*
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override  
*Get left block.*
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override  
*Get right block.*
- void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*p\_gnss\_synchro) override  
*Set acquisition/tracking common Gnss\_Synchro object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.*
- void [set\\_channel](#) (unsigned int channel) override  
*Set acquisition channel unique ID.*
- void [set\\_channel\\_fsm](#) (std::weak\_ptr< ChannelFsm > channel\_fsm) override  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold) override  
*Set statistics threshold of PCPS algorithm.*
- void [set\\_doppler\\_max](#) (unsigned int doppler\_max) override  
*Set maximum Doppler off grid search.*
- void [set\\_doppler\\_step](#) (unsigned int doppler\_step) override  
*Set Doppler steps for the grid search.*
- void [set\\_doppler\\_center](#) (int doppler\_center) override  
*Set Doppler center for the grid search.*
- void [init](#) () override  
*Initializes acquisition algorithm.*
- void [set\\_local\\_code](#) () override  
*Sets local code for GPS L5 PCPS acquisition algorithm.*
- signed int [mag](#) () override  
*Returns the maximum peak of grid search.*
- void [reset](#) () override  
*Restart acquisition algorithm.*
- void [set\\_state](#) (int state) override  
*If state = 1, it forces the block to start acquiring from the first sample.*
- void [stop\\_acquisition](#) () override  
*Stop running acquisition.*
- void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples \_\_attribute\_\_((unused))) override  
*Set resampler latency.*

### Public Member Functions inherited from [AcquisitionInterface](#)

- virtual void [set\\_doppler\\_center](#) (int doppler\_center \_\_attribute\_\_((unused)))
- virtual void [set\\_resampler\\_latency](#) (uint32\_t latency\_samples)=0

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()  
*Start the flow of samples if needed.*

### 10.231.1 Detailed Description

This class adapts a PCPS acquisition block off-loaded on an FPGA to an [AcquisitionInterface](#) for GPS L5i signals. Definition at line 46 of file [gps\\_l5i\\_pcps\\_acquisition\\_fpga.h](#).

### 10.231.2 Constructor & Destructor Documentation

#### 10.231.2.1 GpsL5iPcpsAcquisitionFpga()

```
GpsL5iPcpsAcquisitionFpga::GpsL5iPcpsAcquisitionFpga (
    const ConfigurationInterface * configuration,
    const std::string & role,
    unsigned int in_streams,
    unsigned int out_streams)
```

Constructor.

References [role\(\)](#).

Here is the call graph for this function:



#### 10.231.2.2 ~GpsL5iPcpsAcquisitionFpga()

```
GpsL5iPcpsAcquisitionFpga::~~GpsL5iPcpsAcquisitionFpga () [default]
```

Destructor.

### 10.231.3 Member Function Documentation

#### 10.231.3.1 connect()

```
void GpsL5iPcpsAcquisitionFpga::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Connect.

Implements [GNSSBlockInterface](#).

#### 10.231.3.2 disconnect()

```
void GpsL5iPcpsAcquisitionFpga::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Disconnect.

Implements [GNSSBlockInterface](#).

#### 10.231.3.3 get\_left\_block()

```
gr::basic_block_sptr GpsL5iPcpsAcquisitionFpga::get_left_block () [override], [virtual]
```

Get left block.

Implements [GNSSBlockInterface](#).

#### 10.231.3.4 get\_right\_block()

```
gr::basic_block_sptr GpsL5iPcpsAcquisitionFpga::get_right_block () [override], [virtual]
```

Get right block.

Implements [GNSSBlockInterface](#).



### 10.231.3.5 implementation()

```
std::string GpsL5iPcpsAcquisitionFpga::implementation () [inline], [override], [virtual]
```

Returns "GPS\_L5i\_PCPS\_Acquisition\_FPGA".

Implements [GNSSBlockInterface](#).

Definition at line 74 of file [gps\\_l5i\\_pcps\\_acquisition\\_fpga.h](#).

### 10.231.3.6 init()

```
void GpsL5iPcpsAcquisitionFpga::init () [override], [virtual]
```

Initializes acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.231.3.7 item\_size()

```
size_t GpsL5iPcpsAcquisitionFpga::item_size () [inline], [override], [virtual]
```

Returns size of `lv_16sc_t`.

Implements [GNSSBlockInterface](#).

Definition at line 82 of file [gps\\_l5i\\_pcps\\_acquisition\\_fpga.h](#).

### 10.231.3.8 mag()

```
signed int GpsL5iPcpsAcquisitionFpga::mag () [override], [virtual]
```

Returns the maximum peak of grid search.

Implements [AcquisitionInterface](#).

### 10.231.3.9 reset()

```
void GpsL5iPcpsAcquisitionFpga::reset () [override], [virtual]
```

Restart acquisition algorithm.

Implements [AcquisitionInterface](#).

### 10.231.3.10 role()

```
std::string GpsL5iPcpsAcquisitionFpga::role () [inline], [override], [virtual]
```

Role.

Implements [GNSSBlockInterface](#).

Definition at line 66 of file [gps\\_l5i\\_pcps\\_acquisition\\_fpga.h](#).

Referenced by [GpsL5iPcpsAcquisitionFpga\(\)](#).

### 10.231.3.11 set\_channel()

```
void GpsL5iPcpsAcquisitionFpga::set_channel (  
    unsigned int channel) [inline], [override], [virtual]
```

Set acquisition channel unique ID.

Implements [AcquisitionInterface](#).

Definition at line 117 of file [gps\\_l5i\\_pcps\\_acquisition\\_fpga.h](#).

### 10.231.3.12 set\_channel\_fsm()

```
void GpsL5iPcpsAcquisitionFpga::set_channel_fsm (  
    std::weak_ptr< ChannelFsm > channel_fsm) [inline], [override], [virtual]
```

Set channel fsm associated to this acquisition instance.

Implements [AcquisitionInterface](#).

Definition at line 126 of file [gps\\_l5i\\_pcps\\_acquisition\\_fpga.h](#).

### 10.231.3.13 set\_doppler\_center()

```
void GpsL5iPcpsAcquisitionFpga::set_doppler_center (  
    int doppler_center) [override]
```

Set Doppler center for the grid search.

#### 10.231.3.14 `set_doppler_max()`

```
void GpsL5iPcpsAcquisitionFpga::set_doppler_max (
    unsigned int doppler_max) [override], [virtual]
```

Set maximum Doppler off grid search.

Implements [AcquisitionInterface](#).

#### 10.231.3.15 `set_doppler_step()`

```
void GpsL5iPcpsAcquisitionFpga::set_doppler_step (
    unsigned int doppler_step) [override], [virtual]
```

Set Doppler steps for the grid search.

Implements [AcquisitionInterface](#).

#### 10.231.3.16 `set_gnss_synchro()`

```
void GpsL5iPcpsAcquisitionFpga::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [override], [virtual]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to efficiently exchange synchronization data between acquisition and tracking blocks.

Implements [AcquisitionInterface](#).

#### 10.231.3.17 `set_local_code()`

```
void GpsL5iPcpsAcquisitionFpga::set_local_code () [override], [virtual]
```

Sets local code for GPS L5 PCPS acquisition algorithm.

Implements [AcquisitionInterface](#).

#### 10.231.3.18 `set_resampler_latency()`

```
void GpsL5iPcpsAcquisitionFpga::set_resampler_latency (
    uint32_t latency_samples __attribute__((unused))) [inline], [override]
```

Set resampler latency.

Definition at line 185 of file [gps\\_l5i\\_pcps\\_acquisition\\_fpga.h](#).

#### 10.231.3.19 `set_state()`

```
void GpsL5iPcpsAcquisitionFpga::set_state (
    int state) [override], [virtual]
```

If state = 1, it forces the block to start acquiring from the first sample.

Implements [AcquisitionInterface](#).

#### 10.231.3.20 `set_threshold()`

```
void GpsL5iPcpsAcquisitionFpga::set_threshold (
    float threshold) [override], [virtual]
```

Set statistics threshold of PCPS algorithm.

Implements [AcquisitionInterface](#).

#### 10.231.3.21 `stop_acquisition()`

```
void GpsL5iPcpsAcquisitionFpga::stop_acquisition () [override], [virtual]
```

Stop running acquisition.

Implements [AcquisitionInterface](#).

The documentation for this class was generated from the following file:

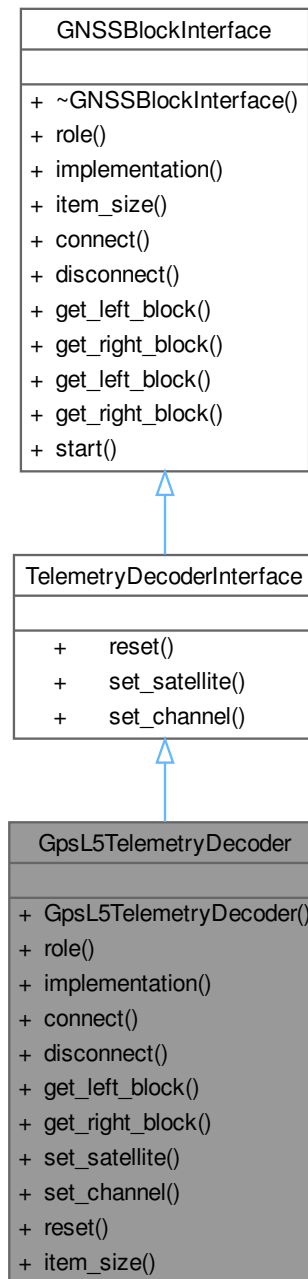
- [gps\\_l5i\\_pcps\\_acquisition\\_fpga.h](#)

## 10.232 GpsL5TelemetryDecoder Class Reference

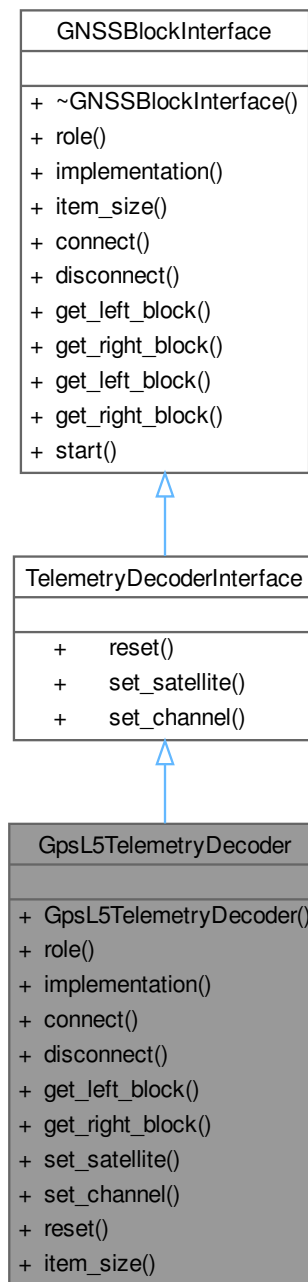
This class implements a NAV data decoder for GPS L5.

```
#include <gps_l5_telemetry_decoder.h>
```

Inheritance diagram for GpsL5TelemetryDecoder:



Collaboration diagram for GpsL5TelemetryDecoder:



### Public Member Functions

- **GpsL5TelemetryDecoder** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override  
*Returns "GPS\_L5\_Telemetry\_Decoder".*
- void [connect](#) (gr::top\_block\_sptr top\_block) override

- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_satellite](#) (const [Gnss\\_Satellite](#) &satellite) override
- void [set\\_channel](#) (int channel) override
- void [reset](#) () override
- size\_t [item\\_size](#) () override

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

## 10.232.1 Detailed Description

This class implements a NAV data decoder for GPS L5.  
Definition at line 43 of file [gps\\_l5\\_telemetry\\_decoder.h](#).

## 10.232.2 Member Function Documentation

### 10.232.2.1 connect()

```
void GpsL5TelemetryDecoder::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.232.2.2 disconnect()

```
void GpsL5TelemetryDecoder::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.232.2.3 get\_left\_block()

```
gr::basic_block_sptr GpsL5TelemetryDecoder::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.232.2.4 get\_right\_block()

```
gr::basic_block_sptr GpsL5TelemetryDecoder::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.232.2.5 implementation()

```
std::string GpsL5TelemetryDecoder::implementation () [inline], [override], [virtual]
```

Returns "GPS\_L5\_Telemetry\_Decoder".

Implements [GNSSBlockInterface](#).

Definition at line 58 of file [gps\\_l5\\_telemetry\\_decoder.h](#).

### 10.232.2.6 item\_size()

```
size_t GpsL5TelemetryDecoder::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 77 of file [gps\\_l5\\_telemetry\\_decoder.h](#).

**10.232.2.7 reset()**

```
void GpsL5TelemetryDecoder::reset () [inline], [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

Definition at line 72 of file [gps\\_l5\\_telemetry\\_decoder.h](#).

**10.232.2.8 role()**

```
std::string GpsL5TelemetryDecoder::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 52 of file [gps\\_l5\\_telemetry\\_decoder.h](#).

**10.232.2.9 set\_channel()**

```
void GpsL5TelemetryDecoder::set_channel (
    int channel) [inline], [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

Definition at line 70 of file [gps\\_l5\\_telemetry\\_decoder.h](#).

**10.232.2.10 set\_satellite()**

```
void GpsL5TelemetryDecoder::set_satellite (
    const Gnss_Satellite & satellite) [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

The documentation for this class was generated from the following file:

- [gps\\_l5\\_telemetry\\_decoder.h](#)

**10.233 GPU\_Complex Struct Reference**

Collaboration diagram for GPU\_Complex:

GPU_Complex
+ r
+ i
+ GPU_Complex()
+ GPU_Complex()
+ magnitude2()
+ operator*()
+ operator+()
+ operator+=()
+ multiply_acc()

**Public Member Functions**

- CUDA\_CALLABLE\_MEMBER\_DEVICE [GPU\\_Complex](#) (float a, float b)
- CUDA\_CALLABLE\_MEMBER\_DEVICE float [magnitude2](#) (void)
- CUDA\_CALLABLE\_MEMBER\_DEVICE [GPU\\_Complex operator\\*](#) (const [GPU\\_Complex](#) &a)

- CUDA\_CALLABLE\_MEMBER\_DEVICE [GPU\\_Complex operator+](#) (const [GPU\\_Complex](#) &a)
- CUDA\_CALLABLE\_MEMBER\_DEVICE void [operator+=](#) (const [GPU\\_Complex](#) &a)
- CUDA\_CALLABLE\_MEMBER\_DEVICE void [multiply\\_acc](#) (const [GPU\\_Complex](#) &a, const [GPU\\_Complex](#) &b)

#### Public Attributes

- float [r](#)
- float [i](#)

### 10.233.1 Detailed Description

Definition at line 45 of file [cuda\\_multicorrelator.h](#).

### 10.233.2 Constructor & Destructor Documentation

#### 10.233.2.1 GPU\_Complex() [1/2]

```
CUDA_CALLABLE_MEMBER_DEVICE GPU_Complex::GPU_Complex () [inline]
```

Definition at line 49 of file [cuda\\_multicorrelator.h](#).

#### 10.233.2.2 GPU\_Complex() [2/2]

```
CUDA_CALLABLE_MEMBER_DEVICE GPU_Complex::GPU_Complex (
    float a,
    float b) [inline]
```

Definition at line 50 of file [cuda\\_multicorrelator.h](#).

### 10.233.3 Member Function Documentation

#### 10.233.3.1 magnitude2()

```
CUDA_CALLABLE_MEMBER_DEVICE float GPU_Complex::magnitude2 (
    void ) [inline]
```

Definition at line 51 of file [cuda\\_multicorrelator.h](#).

#### 10.233.3.2 multiply\_acc()

```
CUDA_CALLABLE_MEMBER_DEVICE void GPU_Complex::multiply_acc (
    const GPU_Complex & a,
    const GPU_Complex & b) [inline]
```

Definition at line 69 of file [cuda\\_multicorrelator.h](#).

#### 10.233.3.3 operator\*()

```
CUDA_CALLABLE_MEMBER_DEVICE GPU_Complex GPU_Complex::operator* (
    const GPU_Complex & a) [inline]
```

Definition at line 52 of file [cuda\\_multicorrelator.h](#).

#### 10.233.3.4 operator+()

```
CUDA_CALLABLE_MEMBER_DEVICE GPU_Complex GPU_Complex::operator+ (
    const GPU_Complex & a) [inline]
```

Definition at line 60 of file [cuda\\_multicorrelator.h](#).

#### 10.233.3.5 operator+=()

```
CUDA_CALLABLE_MEMBER_DEVICE void GPU_Complex::operator+= (
    const GPU_Complex & a) [inline]
```

Definition at line 64 of file [cuda\\_multicorrelator.h](#).

## 10.233.4 Member Data Documentation

### 10.233.4.1 `i`

`float GPU_Complex::i`

Definition at line 48 of file [cuda\\_multicorrelator.h](#).

### 10.233.4.2 `r`

`float GPU_Complex::r`

Definition at line 47 of file [cuda\\_multicorrelator.h](#).

The documentation for this struct was generated from the following file:

- [cuda\\_multicorrelator.h](#)

## 10.234 GPU\_Complex\_Short Struct Reference

Collaboration diagram for GPU\_Complex\_Short:

GPU_Complex_Short
+ <code>r</code>
+ <code>i</code>
+ <code>GPU_Complex_Short()</code>
+ <code>magnitude2()</code>
+ <code>operator*()</code>
+ <code>operator+()</code>

### Public Member Functions

- CUDA\_CALLABLE\_MEMBER\_DEVICE [GPU\\_Complex\\_Short](#) (short int a, short int b)
- CUDA\_CALLABLE\_MEMBER\_DEVICE float [magnitude2](#) (void)
- CUDA\_CALLABLE\_MEMBER\_DEVICE [GPU\\_Complex\\_Short operator\\*](#) (const [GPU\\_Complex\\_Short](#) &a)
- CUDA\_CALLABLE\_MEMBER\_DEVICE [GPU\\_Complex\\_Short operator+](#) (const [GPU\\_Complex\\_Short](#) &a)

### Public Attributes

- float [r](#)
- float [i](#)

### 10.234.1 Detailed Description

Definition at line 88 of file [cuda\\_multicorrelator.h](#).

### 10.234.2 Constructor & Destructor Documentation

#### 10.234.2.1 `GPU_Complex_Short()`

```
CUDA_CALLABLE_MEMBER_DEVICE GPU_Complex_Short::GPU_Complex_Short (
    short int a,
    short int b) [inline]
```

Definition at line 92 of file [cuda\\_multicorrelator.h](#).



### 10.234.3 Member Function Documentation

#### 10.234.3.1 magnitude2()

```
CUDA_CALLABLE_MEMBER_DEVICE float GPU_Complex_Short::magnitude2 (
    void ) [inline]
```

Definition at line 93 of file [cuda\\_multicorrelator.h](#).

#### 10.234.3.2 operator\*()

```
CUDA_CALLABLE_MEMBER_DEVICE GPU_Complex_Short GPU_Complex_Short::operator* (
    const GPU_Complex_Short & a) [inline]
```

Definition at line 97 of file [cuda\\_multicorrelator.h](#).

#### 10.234.3.3 operator+()

```
CUDA_CALLABLE_MEMBER_DEVICE GPU_Complex_Short GPU_Complex_Short::operator+ (
    const GPU_Complex_Short & a) [inline]
```

Definition at line 101 of file [cuda\\_multicorrelator.h](#).

### 10.234.4 Member Data Documentation

#### 10.234.4.1 i

```
float GPU_Complex_Short::i
```

Definition at line 91 of file [cuda\\_multicorrelator.h](#).

#### 10.234.4.2 r

```
float GPU_Complex_Short::r
```

Definition at line 90 of file [cuda\\_multicorrelator.h](#).

The documentation for this struct was generated from the following file:

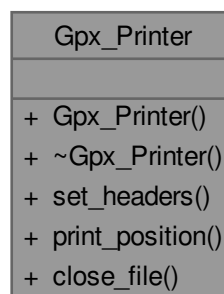
- [cuda\\_multicorrelator.h](#)

## 10.235 Gpx\_Printer Class Reference

Prints PVT information to GPX format file.

```
#include <gpx_printer.h>
```

Collaboration diagram for Gpx\_Printer:



## Public Member Functions

- **Gpx\_Printer** (const std::string &base\_path=".")
- bool **set\_headers** (const std::string &filename, bool time\_tag\_name=true)
- bool **print\_position** (const [Pvt\\_Solution](#) \*const position)
- bool **close\_file** ()

### 10.235.1 Detailed Description

Prints PVT information to GPX format file.

See <https://www.topografix.com/gpx.asp>

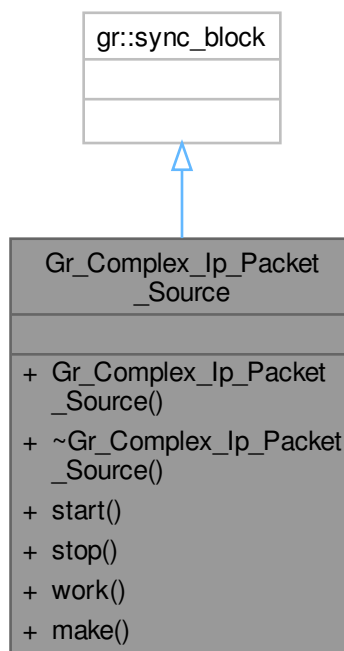
Definition at line 39 of file [gpx\\_printer.h](#).

The documentation for this class was generated from the following file:

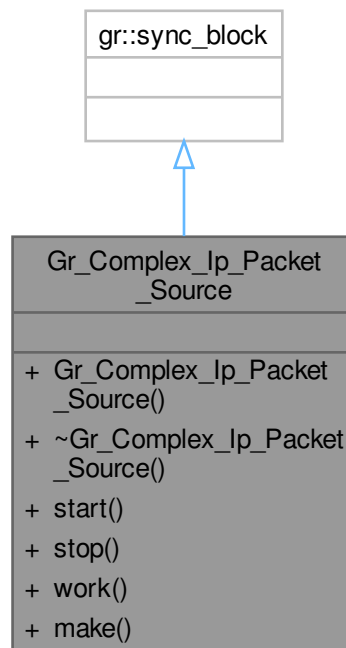
- [gpx\\_printer.h](#)

## 10.236 Gr\_Complex\_Ip\_Packet\_Source Class Reference

Inheritance diagram for Gr\_Complex\_Ip\_Packet\_Source:



Collaboration diagram for Gr\_Complex\_Ip\_Packet\_Source:



### Public Types

- using `sptr` = `gnss_shared_ptr<Gr_Complex_Ip_Packet_Source>`

### Public Member Functions

- **Gr\_Complex\_Ip\_Packet\_Source** (`std::string src_device`, `const std::string &origin_address`, `int udp_port`, `int udp_packet_size`, `int n_baseband_channels`, `const std::string &wire_sample_type`, `size_t item_size`, `bool IQ_swap_`)
- `bool start ()`
- `bool stop ()`
- `int work (int noutput_items, gr_vector_const_void_star &input_items, gr_vector_void_star &output_items)`

### Static Public Member Functions

- static `sptr make` (`std::string src_device`, `const std::string &origin_address`, `int udp_port`, `int udp_packet_size`, `int n_baseband_channels`, `const std::string &wire_sample_type`, `size_t item_size`, `bool IQ_swap_`)

## 10.236.1 Detailed Description

Definition at line 41 of file [gr\\_complex\\_ip\\_packet\\_source.h](#).

## 10.236.2 Member Typedef Documentation

### 10.236.2.1 `sptr`

using `Gr_Complex_Ip_Packet_Source::sptr` = `gnss_shared_ptr<Gr_Complex_Ip_Packet_Source>`

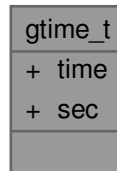
Definition at line 44 of file [gr\\_complex\\_ip\\_packet\\_source.h](#).

The documentation for this class was generated from the following file:

- [gr\\_complex\\_ip\\_packet\\_source.h](#)

## 10.237 gtime\_t Struct Reference

Collaboration diagram for gtime\_t:



### Public Attributes

- `time_t` [time](#)
- `double` [sec](#)

### 10.237.1 Detailed Description

Definition at line [355](#) of file [rtklib.h](#).

### 10.237.2 Member Data Documentation

#### 10.237.2.1 sec

```
double gtime_t::sec
```

Definition at line [358](#) of file [rtklib.h](#).

#### 10.237.2.2 time

```
time_t gtime_t::time
```

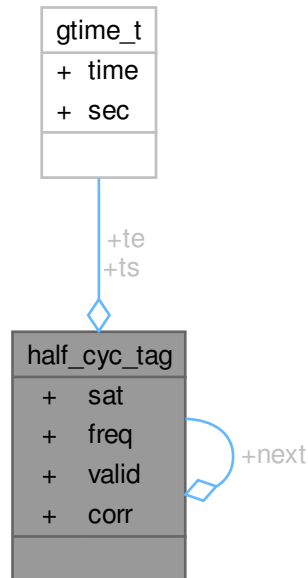
Definition at line [357](#) of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.238 half\_cyc\_tag Struct Reference

Collaboration diagram for half\_cyc\_tag:



### Public Attributes

- unsigned char [sat](#)
- unsigned char [freq](#)
- unsigned char [valid](#)
- char [corr](#)
- [gtime\\_t](#) [ts](#)
- [gtime\\_t](#) [te](#)
- struct [half\\_cyc\\_tag](#) \* [next](#)

### 10.238.1 Detailed Description

Definition at line [1083](#) of file [rtklib.h](#).

### 10.238.2 Member Data Documentation

#### 10.238.2.1 corr

```
char half_cyc_tag::corr
```

Definition at line [1088](#) of file [rtklib.h](#).

#### 10.238.2.2 freq

```
unsigned char half_cyc_tag::freq
```

Definition at line [1086](#) of file [rtklib.h](#).

**10.238.2.3 next**

struct [half\\_cyc\\_tag](#)\* half\_cyc\_tag::next  
 Definition at line [1090](#) of file [rtklib.h](#).

**10.238.2.4 sat**

unsigned char half\_cyc\_tag::sat  
 Definition at line [1085](#) of file [rtklib.h](#).

**10.238.2.5 te**

[gtime\\_t](#) half\_cyc\_tag::te  
 Definition at line [1089](#) of file [rtklib.h](#).

**10.238.2.6 ts**

[gtime\\_t](#) half\_cyc\_tag::ts  
 Definition at line [1089](#) of file [rtklib.h](#).

**10.238.2.7 valid**

unsigned char half\_cyc\_tag::valid  
 Definition at line [1087](#) of file [rtklib.h](#).  
 The documentation for this struct was generated from the following file:

- [rtklib.h](#)

**10.239 HAS\_clock\_corrections Class Reference**

Collaboration diagram for HAS\_clock\_corrections:

HAS_clock_corrections
+ clock_correction_m
+ valid_until
+ HAS_clock_corrections()

**Public Attributes**

- float [clock\\_correction\\_m](#) {}
- uint32\_t [valid\\_until](#) {}

**10.239.1 Detailed Description**

Definition at line [41](#) of file [rtklib\\_conversions.h](#).

## 10.239.2 Member Data Documentation

### 10.239.2.1 clock\_correction\_m

float HAS\_clock\_corrections::clock\_correction\_m {}

Definition at line 45 of file [rtklib\\_conversions.h](#).

### 10.239.2.2 valid\_until

uint32\_t HAS\_clock\_corrections::valid\_until {}

Definition at line 46 of file [rtklib\\_conversions.h](#).

The documentation for this class was generated from the following file:

- [rtklib\\_conversions.h](#)

## 10.240 HAS\_obs\_corrections Class Reference

Collaboration diagram for HAS\_obs\_corrections:

HAS_obs_corrections
+ code_bias_m
+ phase_bias_cycle
+ HAS_obs_corrections()

### Public Attributes

- float [code\\_bias\\_m](#) {}
- float [phase\\_bias\\_cycle](#) {}

### 10.240.1 Detailed Description

Definition at line 60 of file [rtklib\\_conversions.h](#).

## 10.240.2 Member Data Documentation

### 10.240.2.1 code\_bias\_m

float HAS\_obs\_corrections::code\_bias\_m {}

Definition at line 64 of file [rtklib\\_conversions.h](#).

### 10.240.2.2 phase\_bias\_cycle

float HAS\_obs\_corrections::phase\_bias\_cycle {}

Definition at line 65 of file [rtklib\\_conversions.h](#).

The documentation for this class was generated from the following file:

- [rtklib\\_conversions.h](#)

## 10.241 HAS\_orbit\_corrections Class Reference

Collaboration diagram for HAS\_orbit\_corrections:

HAS_orbit_corrections
+ radial_m
+ in_track_m
+ cross_track_m
+ valid_until
+ iod
+ HAS_orbit_corrections()

### Public Attributes

- float [radial\\_m](#) {}
- float [in\\_track\\_m](#) {}
- float [cross\\_track\\_m](#) {}
- uint32\_t [valid\\_until](#) {}
- uint16\_t [iod](#) {}

### 10.241.1 Detailed Description

Definition at line 49 of file [rtklib\\_conversions.h](#).

### 10.241.2 Member Data Documentation

#### 10.241.2.1 cross\_track\_m

```
float HAS_orbit_corrections::cross_track_m {}
```

Definition at line 55 of file [rtklib\\_conversions.h](#).

#### 10.241.2.2 in\_track\_m

```
float HAS_orbit_corrections::in_track_m {}
```

Definition at line 54 of file [rtklib\\_conversions.h](#).

#### 10.241.2.3 iod

```
uint16_t HAS_orbit_corrections::iod {}
```

Definition at line 57 of file [rtklib\\_conversions.h](#).

#### 10.241.2.4 radial\_m

```
float HAS_orbit_corrections::radial_m {}
```

Definition at line 53 of file [rtklib\\_conversions.h](#).



### 10.241.2.5 valid\_until

```
uint32_t HAS_orbit_corrections::valid_until {}
```

Definition at line 56 of file [rtklib\\_conversions.h](#).

The documentation for this class was generated from the following file:

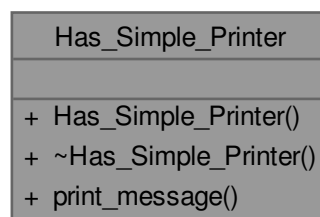
- [rtklib\\_conversions.h](#)

## 10.242 Has\_Simple\_Printer Class Reference

Prints HAS messages content in a txt file. See HAS-SIS-ICD for a message description.

```
#include <has_simple_printer.h>
```

Collaboration diagram for Has\_Simple\_Printer:



### Public Member Functions

- **Has\_Simple\_Printer** (const std::string &base\_path=std::string("."), const std::string &filename=std::string("HAS\_Messages"), bool time\_tag\_name=true)
- bool **print\_message** (const [Galileo\\_HAS\\_data](#) \*const has\_data)

### 10.242.1 Detailed Description

Prints HAS messages content in a txt file. See HAS-SIS-ICD for a message description.

Definition at line 37 of file [has\\_simple\\_printer.h](#).

The documentation for this class was generated from the following file:

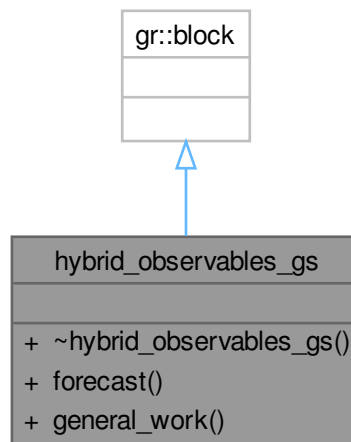
- [has\\_simple\\_printer.h](#)

## 10.243 hybrid\_observables\_gs Class Reference

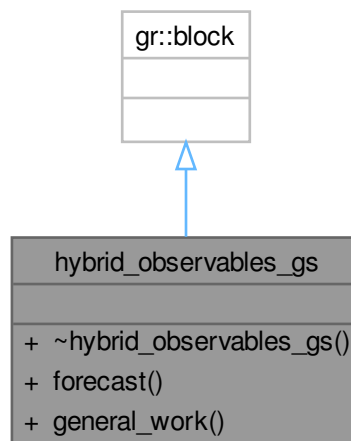
This class implements a block that computes observables.

```
#include <hybrid_observables_gs.h>
```

Inheritance diagram for `hybrid_observables_gs`:



Collaboration diagram for `hybrid_observables_gs`:



## Public Member Functions

- void **forecast** (int noutput\_items, gr\_vector\_int &ninput\_items\_required)
- int **general\_work** (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)

## Friends

- `hybrid_observables_gs_sptr` **hybrid\_observables\_gs\_make** (const [Obs\\_Conf](#) &conf\_)

### 10.243.1 Detailed Description

This class implements a block that computes observables.

Definition at line 58 of file [hybrid\\_observables\\_gs.h](#).

The documentation for this class was generated from the following file:

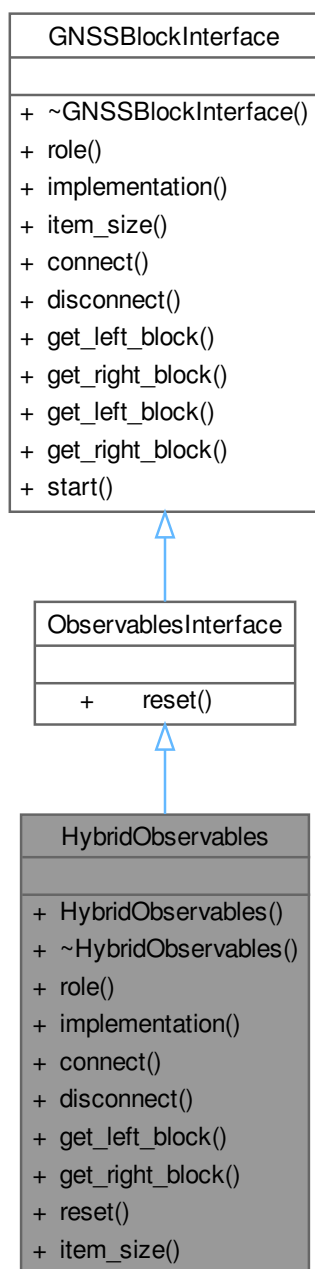
- [hybrid\\_observables\\_gs.h](#)

## 10.244 HybridObservables Class Reference

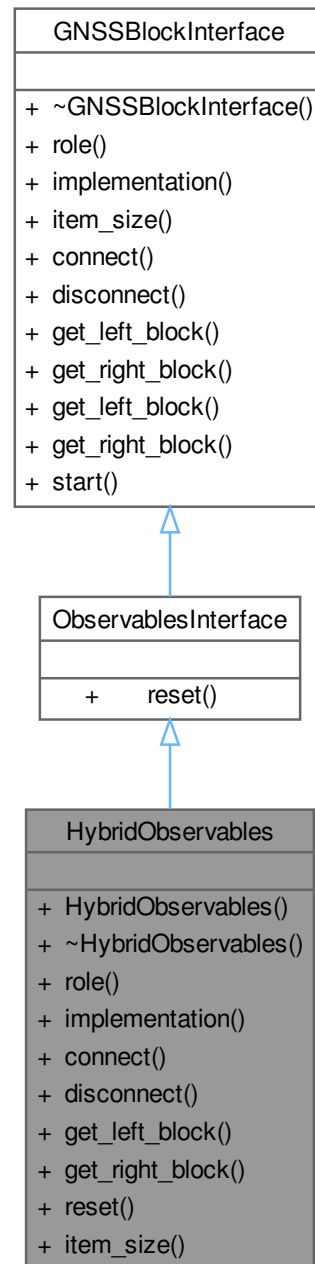
This class implements an [ObservablesInterface](#) for observables of all kind of GNSS signals.

```
#include <hybrid_observables.h>
```

Inheritance diagram for HybridObservables:



Collaboration diagram for HybridObservables:



## Public Member Functions

- **HybridObservables** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "Hybrid\_Observables".
- void [connect](#) (gr::top\_block\_sptr top\_block) override

- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [reset](#) () override
- size\_t [item\\_size](#) () override

*All blocks must have an [item\\_size\(\)](#) function implementation.*

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.244.1 Detailed Description

This class implements an [ObservablesInterface](#) for observables of all kind of GNSS signals.  
Definition at line 43 of file [hybrid\\_observables.h](#).

### 10.244.2 Member Function Documentation

#### 10.244.2.1 connect()

```
void HybridObservables::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.244.2.2 disconnect()

```
void HybridObservables::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.244.2.3 get\_left\_block()

```
gr::basic_block_sptr HybridObservables::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.244.2.4 get\_right\_block()

```
gr::basic_block_sptr HybridObservables::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.244.2.5 implementation()

```
std::string HybridObservables::implementation () [inline], [override], [virtual]
```

Returns "Hybrid\_Observables".

Implements [GNSSBlockInterface](#).

Definition at line 59 of file [hybrid\\_observables.h](#).

#### 10.244.2.6 item\_size()

```
size_t HybridObservables::item_size () [inline], [override], [virtual]
```

All blocks must have an [item\\_size\(\)](#) function implementation.

Implements [GNSSBlockInterface](#).

Definition at line 75 of file [hybrid\\_observables.h](#).

#### 10.244.2.7 reset()

`void HybridObservables::reset () [inline], [override], [virtual]`

Implements [ObservablesInterface](#).

Definition at line 69 of file [hybrid\\_observables.h](#).

#### 10.244.2.8 role()

`std::string HybridObservables::role () [inline], [override], [virtual]`

Implements [GNSSBlockInterface](#).

Definition at line 53 of file [hybrid\\_observables.h](#).

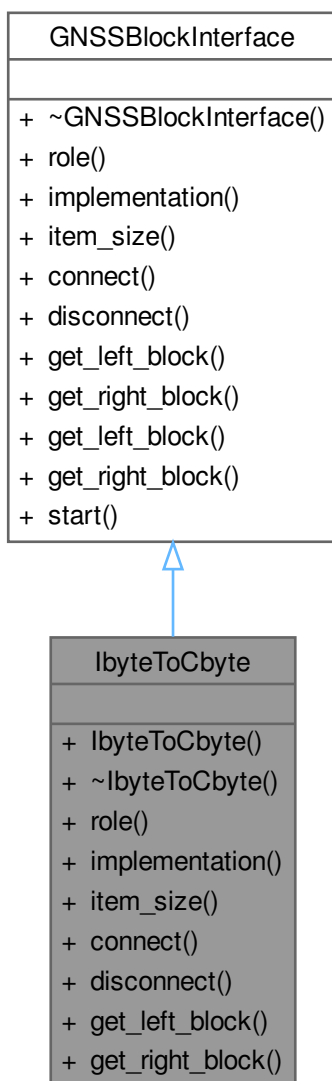
The documentation for this class was generated from the following file:

- [hybrid\\_observables.h](#)

## 10.245 IbyteToCbyte Class Reference

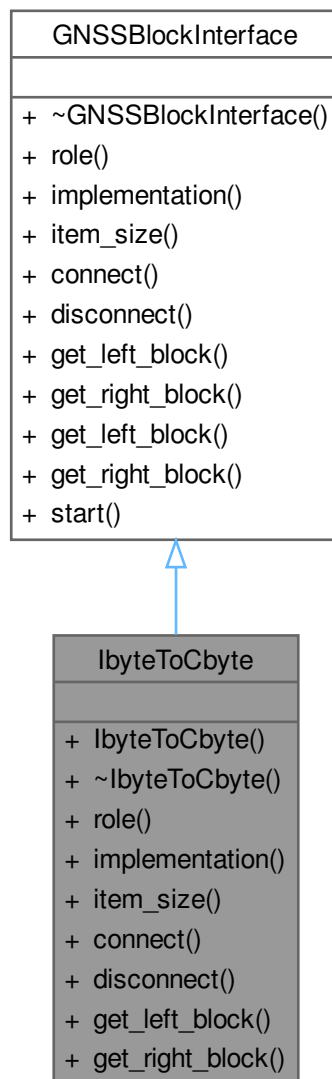
```
#include <ibyte_to_cbyte.h>
```

Inheritance diagram for lbyteToCbyte:





Collaboration diagram for lbyteToCbyte:



### Public Member Functions

- **lbyteToCbyte** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override  
*Returns "lbyte\_To\_Cbyte".*
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual `gr::basic_block_sptr` [get\\_left\\_block](#) (int RF\_channel)
- virtual `gr::basic_block_sptr` [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.245.1 Detailed Description

\brief Adapts an I/Q interleaved byte (unsigned char) sample stream into a `std::complex<unsigned char>` stream  
Definition at line 40 of file [ibyte\\_to\\_cbyte.h](#).

### 10.245.2 Member Function Documentation

#### 10.245.2.1 `connect()`

```
void IbyteToCbyte::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.245.2.2 `disconnect()`

```
void IbyteToCbyte::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.245.2.3 `get_left_block()`

```
gr::basic_block_sptr IbyteToCbyte::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.245.2.4 `get_right_block()`

```
gr::basic_block_sptr IbyteToCbyte::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.245.2.5 `implementation()`

```
std::string IbyteToCbyte::implementation () [inline], [override], [virtual]
```

Returns "Ibyte\_To\_Cbyte".

Implements [GNSSBlockInterface](#).

Definition at line 55 of file [ibyte\\_to\\_cbyte.h](#).

#### 10.245.2.6 `item_size()`

```
size_t IbyteToCbyte::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 60 of file [ibyte\\_to\\_cbyte.h](#).

#### 10.245.2.7 `role()`

```
std::string IbyteToCbyte::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 49 of file [ibyte\\_to\\_cbyte.h](#).

The documentation for this class was generated from the following file:

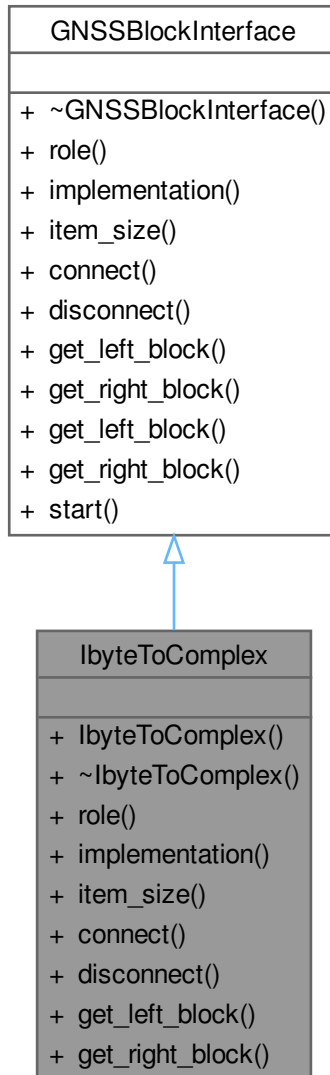
- [ibyte\\_to\\_cbyte.h](#)

## 10.246 lbyteToComplex Class Reference

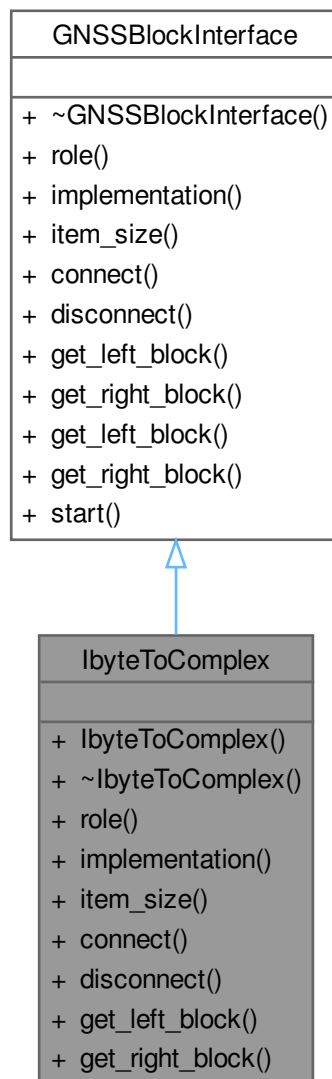
Adapts an I/Q interleaved byte integer sample stream to a gr\_complex (float) stream.

```
#include <lbyte_to_complex.h>
```

Inheritance diagram for lbyteToComplex:



Collaboration diagram for IbyteToComplex:



## Public Member Functions

- **IbyteToComplex** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "Ibyte\_To\_Complex".
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override

## Public Member Functions inherited from GNSSBlockInterface

- virtual `gr::basic_block_sptr` [get\\_left\\_block](#) (int RF\_channel)
- virtual `gr::basic_block_sptr` [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.246.1 Detailed Description

Adapts an I/Q interleaved byte integer sample stream to a `gr_complex` (float) stream.  
Definition at line 39 of file [ibyte\\_to\\_complex.h](#).

### 10.246.2 Member Function Documentation

#### 10.246.2.1 connect()

```
void IbyteToComplex::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.246.2.2 disconnect()

```
void IbyteToComplex::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.246.2.3 get\_left\_block()

```
gr::basic_block_sptr IbyteToComplex::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.246.2.4 get\_right\_block()

```
gr::basic_block_sptr IbyteToComplex::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.246.2.5 implementation()

```
std::string IbyteToComplex::implementation () [inline], [override], [virtual]
```

Returns "Ibyte\_To\_Complex".

Implements [GNSSBlockInterface](#).

Definition at line 54 of file [ibyte\\_to\\_complex.h](#).

#### 10.246.2.6 item\_size()

```
size_t IbyteToComplex::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 59 of file [ibyte\\_to\\_complex.h](#).

#### 10.246.2.7 role()

```
std::string IbyteToComplex::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 48 of file [ibyte\\_to\\_complex.h](#).

The documentation for this class was generated from the following file:

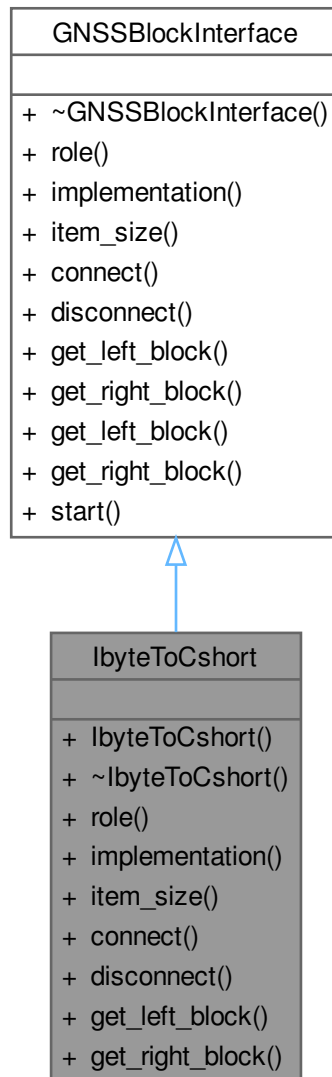
- [ibyte\\_to\\_complex.h](#)

## 10.247 lbyteToCshort Class Reference

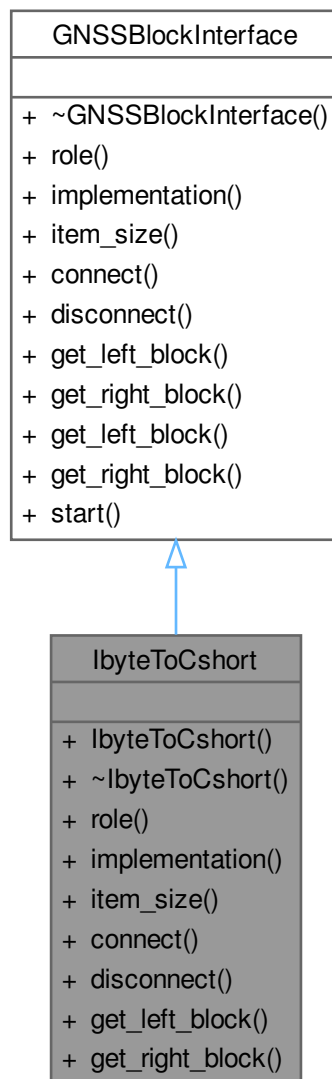
Adapts a short integer (16 bits) interleaved sample stream into a `std::complex<short>` stream.

```
#include <lbyte_to_cshort.h>
```

Inheritance diagram for lbyteToCshort:



Collaboration diagram for IbyteToCshort:



## Public Member Functions

- **IbyteToCshort** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "Ibyte\_To\_Cshort".
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual `gr::basic_block_sptr` [get\\_left\\_block](#) (int RF\_channel)
- virtual `gr::basic_block_sptr` [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.247.1 Detailed Description

Adapts a short integer (16 bits) interleaved sample stream into a `std::complex<short>` stream.  
Definition at line 39 of file [ibyte\\_to\\_cshort.h](#).

### 10.247.2 Member Function Documentation

#### 10.247.2.1 `connect()`

```
void IbyteToCshort::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.247.2.2 `disconnect()`

```
void IbyteToCshort::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.247.2.3 `get_left_block()`

```
gr::basic_block_sptr IbyteToCshort::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.247.2.4 `get_right_block()`

```
gr::basic_block_sptr IbyteToCshort::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.247.2.5 `implementation()`

```
std::string IbyteToCshort::implementation () [inline], [override], [virtual]
```

Returns "Ibyte\_To\_Cshort".

Implements [GNSSBlockInterface](#).

Definition at line 54 of file [ibyte\\_to\\_cshort.h](#).

#### 10.247.2.6 `item_size()`

```
size_t IbyteToCshort::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 59 of file [ibyte\\_to\\_cshort.h](#).

#### 10.247.2.7 `role()`

```
std::string IbyteToCshort::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 48 of file [ibyte\\_to\\_cshort.h](#).

The documentation for this class was generated from the following file:

- [ibyte\\_to\\_cshort.h](#)

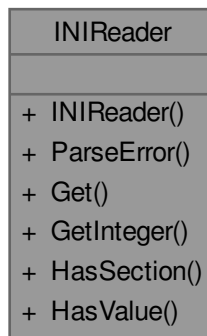


## 10.248 INIReader Class Reference

Read an INI file into easy-to-access name/value pairs. (Note that I've gone for simplicity here rather than speed, but it should be pretty decent.)

```
#include <INIReader.h>
```

Collaboration diagram for INIReader:



### Public Member Functions

- **INIReader** (const std::string &filename)  
Construct **INIReader** and parse given filename. See [ini.h](#) for more info about the parsing.
- int **ParseError** () const  
Return the result of [ini\\_parse\(\)](#), i.e., 0 on success, line number of first error on parse error, or -1 on file open error.
- std::string **Get** (const std::string &section, const std::string &name, const std::string &default\_value)  
Get a string value from INI file, returning default\_value if not found.
- int64\_t **GetInteger** (const std::string &section, const std::string &name, int64\_t default\_value)  
Get an integer (long) value from INI file, returning default\_value if not found.
- bool **HasSection** (const std::string &section) const  
Return true if the given section exists (section must contain at least one name=value pair).
- bool **HasValue** (const std::string &section, const std::string &name) const  
Return true if a value exists with the given section and field names.

### 10.248.1 Detailed Description

Read an INI file into easy-to-access name/value pairs. (Note that I've gone for simplicity here rather than speed, but it should be pretty decent.)

Definition at line 45 of file [INIReader.h](#).

### 10.248.2 Constructor & Destructor Documentation

#### 10.248.2.1 INIReader()

```
INIReader::INIReader (
    const std::string & filename) [explicit]
```

Construct **INIReader** and parse given filename. See [ini.h](#) for more info about the parsing.

### 10.248.3 Member Function Documentation

#### 10.248.3.1 Get()

```
std::string INIReader::Get (
    const std::string & section,
    const std::string & name,
    const std::string & default_value)
```

Get a string value from INI file, returning default\_value if not found.

#### 10.248.3.2 GetInteger()

```
int64_t INIReader::GetInteger (
    const std::string & section,
    const std::string & name,
    int64_t default_value)
```

Get an integer (long) value from INI file, returning default\_value if not found.

#### 10.248.3.3 HasSection()

```
bool INIReader::HasSection (
    const std::string & section) const
```

Return true if the given section exists (section must contain at least one name=value pair).

#### 10.248.3.4 HasValue()

```
bool INIReader::HasValue (
    const std::string & section,
    const std::string & name) const
```

Return true if a value exists with the given section and field names.

#### 10.248.3.5 ParseError()

```
int INIReader::ParseError () const
```

Return the result of [ini\\_parse\(\)](#), i.e., 0 on success, line number of first error on parse error, or -1 on file open error. The documentation for this class was generated from the following file:

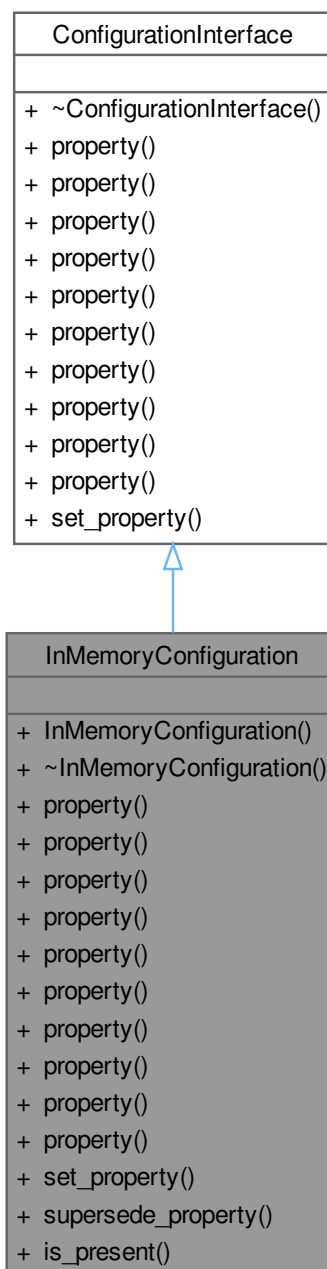
- [INIReader.h](#)

## 10.249 InMemoryConfiguration Class Reference

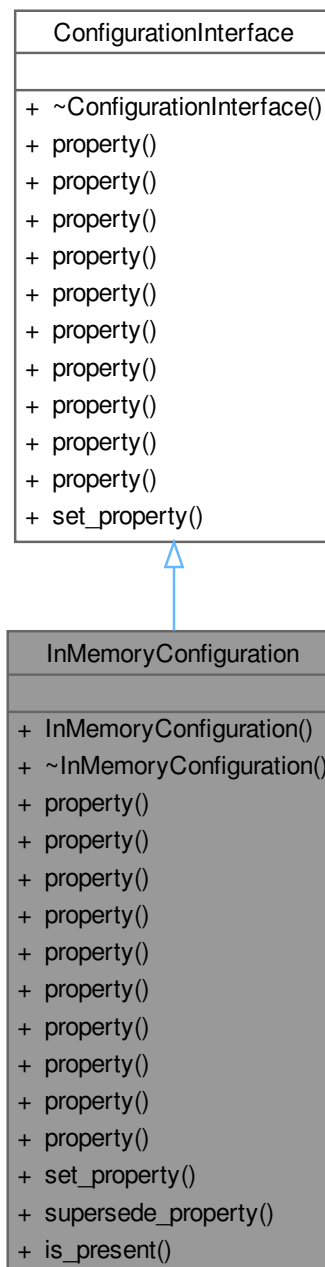
This class is an implementation of the interface [ConfigurationInterface](#).

```
#include <in_memory_configuration.h>
```

Inheritance diagram for InMemoryConfiguration:



Collaboration diagram for InMemoryConfiguration:



### Public Member Functions

- `std::string property (std::string property_name, std::string default_value) const` override
- `bool property (std::string property_name, bool default_value) const` override
- `int64_t property (std::string property_name, int64_t default_value) const` override
- `uint64_t property (std::string property_name, uint64_t default_value) const` override
- `int32_t property (std::string property_name, int32_t default_value) const` override
- `uint32_t property (std::string property_name, uint32_t default_value) const` override

- `int16_t` [property](#) (`std::string property_name`, `int16_t default_value`) `const` override
- `uint16_t` [property](#) (`std::string property_name`, `uint16_t default_value`) `const` override
- `float` [property](#) (`std::string property_name`, `float default_value`) `const` override
- `double` [property](#) (`std::string property_name`, `double default_value`) `const` override
- `void` [set\\_property](#) (`std::string property_name`, `std::string value`) override
- `void` [supersede\\_property](#) (`const std::string &property_name`, `const std::string &value`)
- `bool` [is\\_present](#) (`const std::string &property_name`) `const`

### 10.249.1 Detailed Description

This class is an implementation of the interface [ConfigurationInterface](#).

This implementation accepts configuration parameters upon instantiation and it is intended to be used in unit testing.

Definition at line 43 of file [in\\_memory\\_configuration.h](#).

### 10.249.2 Member Function Documentation

#### 10.249.2.1 [property\(\)](#) [1/10]

```
bool InMemoryConfiguration::property (
    std::string property_name,
    bool default_value) const [override], [virtual]
```

Implements [ConfigurationInterface](#).

#### 10.249.2.2 [property\(\)](#) [2/10]

```
double InMemoryConfiguration::property (
    std::string property_name,
    double default_value) const [override], [virtual]
```

Implements [ConfigurationInterface](#).

#### 10.249.2.3 [property\(\)](#) [3/10]

```
float InMemoryConfiguration::property (
    std::string property_name,
    float default_value) const [override], [virtual]
```

Implements [ConfigurationInterface](#).

#### 10.249.2.4 [property\(\)](#) [4/10]

```
int16_t InMemoryConfiguration::property (
    std::string property_name,
    int16_t default_value) const [override], [virtual]
```

Implements [ConfigurationInterface](#).

#### 10.249.2.5 [property\(\)](#) [5/10]

```
int32_t InMemoryConfiguration::property (
    std::string property_name,
    int32_t default_value) const [override], [virtual]
```

Implements [ConfigurationInterface](#).

#### 10.249.2.6 [property\(\)](#) [6/10]

```
int64_t InMemoryConfiguration::property (
    std::string property_name,
    int64_t default_value) const [override], [virtual]
```

Implements [ConfigurationInterface](#).

**10.249.2.7 property()** [7/10]

```
std::string InMemoryConfiguration::property (  
    std::string property_name,  
    std::string default_value) const [override], [virtual]
```

Implements [ConfigurationInterface](#).

**10.249.2.8 property()** [8/10]

```
uint16_t InMemoryConfiguration::property (  
    std::string property_name,  
    uint16_t default_value) const [override], [virtual]
```

Implements [ConfigurationInterface](#).

**10.249.2.9 property()** [9/10]

```
uint32_t InMemoryConfiguration::property (  
    std::string property_name,  
    uint32_t default_value) const [override], [virtual]
```

Implements [ConfigurationInterface](#).

**10.249.2.10 property()** [10/10]

```
uint64_t InMemoryConfiguration::property (  
    std::string property_name,  
    uint64_t default_value) const [override], [virtual]
```

Implements [ConfigurationInterface](#).

**10.249.2.11 set\_property()**

```
void InMemoryConfiguration::set_property (  
    std::string property_name,  
    std::string value) [override], [virtual]
```

Implements [ConfigurationInterface](#).

The documentation for this class was generated from the following file:

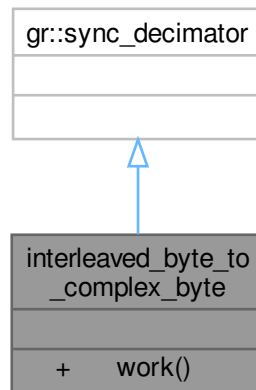
- [in\\_memory\\_configuration.h](#)

**10.250 interleaved\_byte\_to\_complex\_byte Class Reference**

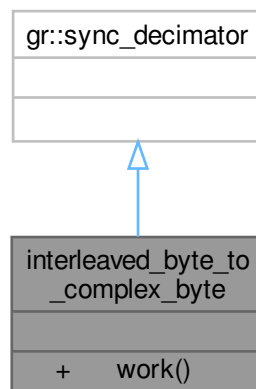
This class adapts an 8-bits interleaved sample stream into a 16-bits complex stream (`std::complex<unsigned char>`)

```
#include <interleaved_byte_to_complex_byte.h>
```

Inheritance diagram for interleaved\_byte\_to\_complex\_byte:



Collaboration diagram for interleaved\_byte\_to\_complex\_byte:



### Public Member Functions

- `int` **work** (`int` noutput\_items, `gr_vector_const_void_star` &input\_items, `gr_vector_void_star` &output\_items)

### Friends

- `interleaved_byte_to_complex_byte_sptr` **make\_interleaved\_byte\_to\_complex\_byte** ()

### 10.250.1 Detailed Description

This class adapts an 8-bits interleaved sample stream into a 16-bits complex stream (`std::complex<unsigned char>`)

Definition at line 40 of file [interleaved\\_byte\\_to\\_complex\\_byte.h](#).

The documentation for this class was generated from the following file:

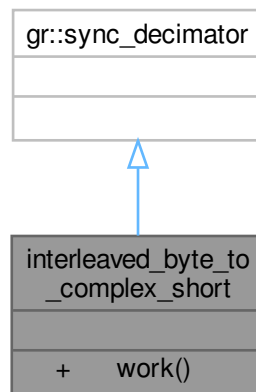
- [interleaved\\_byte\\_to\\_complex\\_byte.h](#)

## 10.251 interleaved\_byte\_to\_complex\_short Class Reference

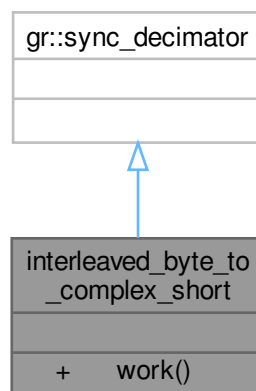
This class adapts a short (16-bits) interleaved sample stream into a `std::complex<short>` stream.

```
#include <interleaved_byte_to_complex_short.h>
```

Inheritance diagram for `interleaved_byte_to_complex_short`:



Collaboration diagram for `interleaved_byte_to_complex_short`:



### Public Member Functions

- `int` **work** (`int` noutput\_items, `gr_vector_const_void_star` &input\_items, `gr_vector_void_star` &output\_items)



## Friends

- `interleaved_byte_to_complex_short_sptr make_interleaved_byte_to_complex_short ()`

### 10.251.1 Detailed Description

This class adapts a short (16-bits) interleaved sample stream into a `std::complex<short>` stream.

Definition at line 40 of file [interleaved\\_byte\\_to\\_complex\\_short.h](#).

The documentation for this class was generated from the following file:

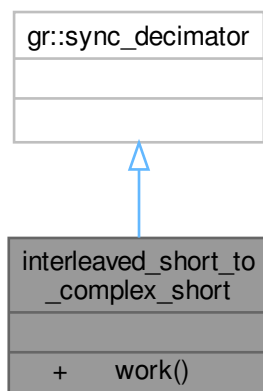
- [interleaved\\_byte\\_to\\_complex\\_short.h](#)

## 10.252 interleaved\_short\_to\_complex\_short Class Reference

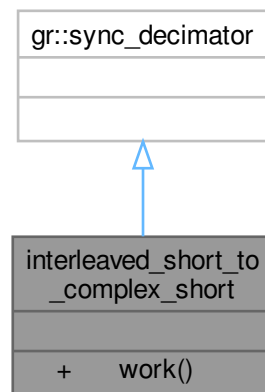
This class adapts a short (16-bits) interleaved sample stream into a `std::complex<short>` stream.

`#include <interleaved_short_to_complex_short.h>`

Inheritance diagram for `interleaved_short_to_complex_short`:



Collaboration diagram for `interleaved_short_to_complex_short`:



#### Public Member Functions

- `int` **work** (`int` noutput\_items, `gr_vector_const_void_star` &input\_items, `gr_vector_void_star` &output\_items)

#### Friends

- `interleaved_short_to_complex_short_sptr` **make\_interleaved\_short\_to\_complex\_short** ()

### 10.252.1 Detailed Description

This class adapts a short (16-bits) interleaved sample stream into a `std::complex<short>` stream.

Definition at line 39 of file [interleaved\\_short\\_to\\_complex\\_short.h](#).

The documentation for this class was generated from the following file:

- [interleaved\\_short\\_to\\_complex\\_short.h](#)

## 10.253 IONGSMSChunkData Class Reference

Collaboration diagram for IONGSMSChunkData:

IONGSMSChunkData
<ul style="list-style-type: none"> <li>+ IONGSMSChunkData()</li> <li>+ ~IONGSMSChunkData()</li> <li>+ IONGSMSChunkData()</li> <li>+ operator=()</li> <li>+ IONGSMSChunkData()</li> <li>+ operator=()</li> <li>+ read_from_buffer()</li> <li>+ write_to_output()</li> <li>+ output_stream_count()</li> <li>+ output_stream_item_size()</li> <li>+ output_stream_item_rate()</li> </ul>

### Public Member Functions

- **IONGSMSChunkData** (const GnssMetadata::Chunk &chunk, const std::vector< std::string > &stream\_ids, std::size\_t output\_stream\_offset)
- **IONGSMSChunkData** (const [IONGSMSChunkData](#) &rhl)=delete
- [IONGSMSChunkData](#) & **operator=** (const [IONGSMSChunkData](#) &rhl)=delete
- **IONGSMSChunkData** ([IONGSMSChunkData](#) &&rhl)=delete
- [IONGSMSChunkData](#) & **operator=** ([IONGSMSChunkData](#) &&rhl)=delete
- std::size\_t **read\_from\_buffer** (uint8\_t \*buffer, std::size\_t offset)
- void **write\_to\_output** (gr\_vector\_void\_star &outputs, std::vector< int > &output\_items)
- std::size\_t **output\_stream\_count** () const
- std::size\_t **output\_stream\_item\_size** (std::size\_t stream\_index) const
- std::size\_t **output\_stream\_item\_rate** (std::size\_t stream\_index) const

### 10.253.1 Detailed Description

Definition at line 112 of file [ion\\_gsms\\_chunk\\_data.h](#).

The documentation for this class was generated from the following file:

- [ion\\_gsms\\_chunk\\_data.h](#)

## 10.254 IONGSMSChunkUnpackingCtx< WT > Struct Template Reference

Collaboration diagram for IONGSMSChunkUnpackingCtx< WT >:

IONGSMSChunkUnpackingCtx < WT >
+ word_shift_direction_ + iterator_ + current_word_ + bitshift_ + word_bitsize_ + IONGSMSChunkUnpackingCtx() + advance_word() + shift_current_word() + shift_padding() + shift_sample()

### Public Member Functions

- [IONGSMSChunkUnpackingCtx](#) (const GnssMetadata::Chunk::WordShift word\_shift, WT \*data\_buffer, uint8\_t data\_buffer\_word\_count)
- void [advance\\_word](#) ()
- void [shift\\_current\\_word](#) (uint8\_t n)
- void [shift\\_padding](#) (uint8\_t n\_bits)
- template<typename OT>  
void [shift\\_sample](#) (uint8\_t sample\_bitsize, OT \*output, uint8\_t output\_bit\_offset=0)

### Public Attributes

- const GnssMetadata::Chunk::WordShift [word\\_shift\\_direction\\_](#)
- WT \* [iterator\\_](#) = nullptr
- WT [current\\_word\\_](#) {}
- uint8\_t [bitshift\\_](#) = 0

### Static Public Attributes

- static constexpr uint8\_t [word\\_bitsize\\_](#) = sizeof(WT) \* 8

### 10.254.1 Detailed Description

```
template<typename WT>
struct IONGSMSChunkUnpackingCtx< WT >
```

Definition at line 32 of file [ion\\_gsms\\_chunk\\_unpacking\\_ctx.h](#).

## 10.254.2 Constructor & Destructor Documentation

### 10.254.2.1 IONGSMSChunkUnpackingCtx()

```
template<typename WT>
IONGSMSChunkUnpackingCtx< WT >::IONGSMSChunkUnpackingCtx (
    const GnssMetadata::Chunk::WordShift word_shift,
    WT * data_buffer,
    uint8_t data_buffer_word_count) [inline]
```

Definition at line 41 of file [ion\\_gsms\\_chunk\\_unpacking\\_ctx.h](#).

## 10.254.3 Member Function Documentation

### 10.254.3.1 advance\_word()

```
template<typename WT>
void IONGSMSChunkUnpackingCtx< WT >::advance_word () [inline]
```

Definition at line 60 of file [ion\\_gsms\\_chunk\\_unpacking\\_ctx.h](#).

### 10.254.3.2 shift\_current\_word()

```
template<typename WT>
void IONGSMSChunkUnpackingCtx< WT >::shift_current_word (
    uint8_t n) [inline]
```

Definition at line 75 of file [ion\\_gsms\\_chunk\\_unpacking\\_ctx.h](#).

### 10.254.3.3 shift\_padding()

```
template<typename WT>
void IONGSMSChunkUnpackingCtx< WT >::shift_padding (
    uint8_t n_bits) [inline]
```

Definition at line 103 of file [ion\\_gsms\\_chunk\\_unpacking\\_ctx.h](#).

### 10.254.3.4 shift\_sample()

```
template<typename WT>
template<typename OT>
void IONGSMSChunkUnpackingCtx< WT >::shift_sample (
    uint8_t sample_bitsize,
    OT * output,
    uint8_t output_bit_offset = 0) [inline]
```

Definition at line 124 of file [ion\\_gsms\\_chunk\\_unpacking\\_ctx.h](#).

## 10.254.4 Member Data Documentation

### 10.254.4.1 bitshift\_

```
template<typename WT>
uint8_t IONGSMSChunkUnpackingCtx< WT >::bitshift_ = 0
```

Definition at line 39 of file [ion\\_gsms\\_chunk\\_unpacking\\_ctx.h](#).

### 10.254.4.2 current\_word\_

```
template<typename WT>
WT IONGSMSChunkUnpackingCtx< WT >::current_word_ {}
```

Definition at line 38 of file [ion\\_gsms\\_chunk\\_unpacking\\_ctx.h](#).

### 10.254.4.3 iterator\_

```
template<typename WT>
WT* IONGSMSChunkUnpackingCtx< WT >::iterator_ = nullptr
```

Definition at line 37 of file [ion\\_gsms\\_chunk\\_unpacking\\_ctx.h](#).

#### 10.254.4.4 word\_bitsize\_

```
template<typename WT>
uint8_t IONGSMSChunkUnpackingCtx< WT >::word_bitsize_ = sizeof(WT) * 8 [static], [constexpr]
Definition at line 34 of file ion\_gsms\_chunk\_unpacking\_ctx.h.
```

#### 10.254.4.5 word\_shift\_direction\_

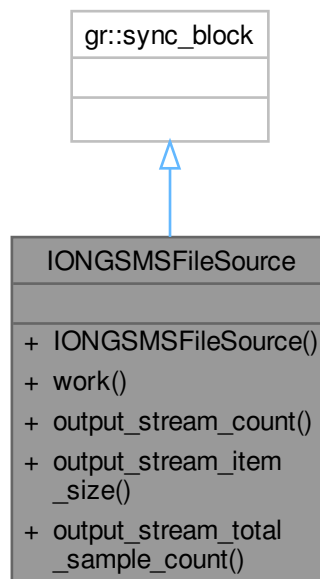
```
template<typename WT>
const GnssMetadata::Chunk::WordShift IONGSMSChunkUnpackingCtx< WT >::word_shift_direction_
Definition at line 36 of file ion\_gsms\_chunk\_unpacking\_ctx.h.
```

The documentation for this struct was generated from the following file:

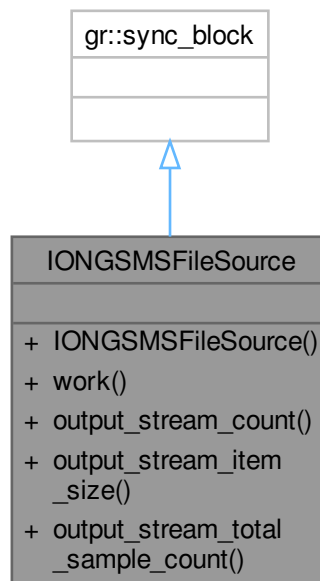
- [ion\\_gsms\\_chunk\\_unpacking\\_ctx.h](#)

## 10.255 IONGSMSFileSource Class Reference

Inheritance diagram for IONGSMSFileSource:



Collaboration diagram for IONGSMSFileSource:



## Public Types

- using `sptr` = `gnss_shared_ptr<IONGSMSFileSource>`

## Public Member Functions

- **IONGSMSFileSource** (const fs::path &metadata\_filepath, const GnssMetadata::File &file, const GnssMetadata::Block &block, const std::vector< std::string > &stream\_ids)
- int **work** (int noutput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items) override
- std::size\_t **output\_stream\_count** () const
- std::size\_t **output\_stream\_item\_size** (std::size\_t stream\_index) const
- std::size\_t **output\_stream\_total\_sample\_count** (std::size\_t stream\_index) const

### 10.255.1 Detailed Description

Definition at line 36 of file [ion\\_gsms.h](#).

### 10.255.2 Member Typedef Documentation

#### 10.255.2.1 sptr

using `IONGSMSFileSource::sptr` = `gnss_shared_ptr<IONGSMSFileSource>`

Definition at line 39 of file [ion\\_gsms.h](#).

The documentation for this class was generated from the following file:

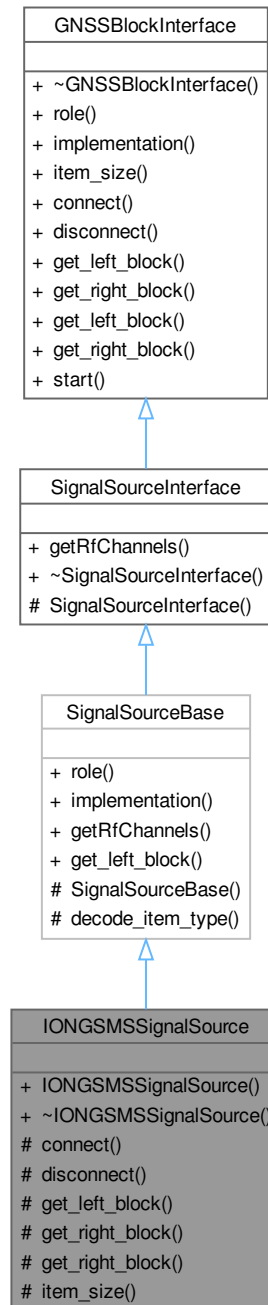
- [ion\\_gsms.h](#)

## 10.256 IONGSMSSignalSource Class Reference

Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).

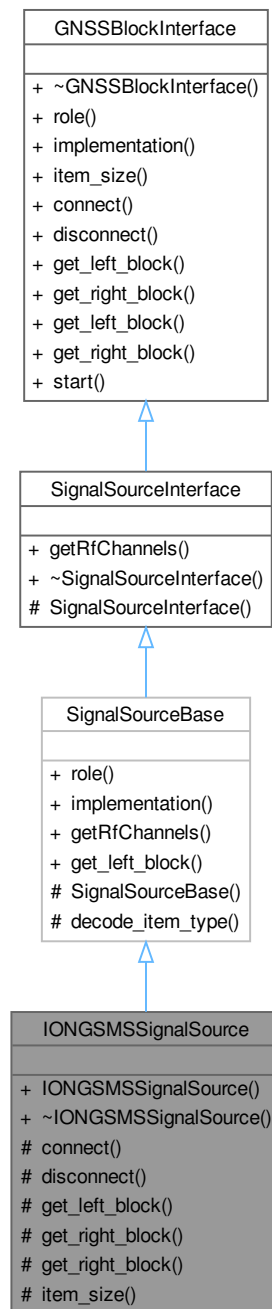
#include <ion\_gsms\_signal\_source.h>

Inheritance diagram for IONGSMSSignalSource:





Collaboration diagram for IONGSMSSignalSource:



### Public Member Functions

- **IONGSMSSignalSource** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)

### Public Member Functions inherited from [SignalSourceBase](#)

- std::string [role](#) () final
- std::string [implementation](#) () final

- `size_t getRfChannels ()` const override
- `gr::basic_block_sptr get_left_block ()` override

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual `gr::basic_block_sptr get_left_block (int RF_channel)`
- virtual void `start ()`

*Start the flow of samples if needed.*

## Protected Member Functions

- void `connect (gr::top_block_sptr top_block)` override
- void `disconnect (gr::top_block_sptr top_block)` override
- `gr::basic_block_sptr get_left_block ()` override
- `gr::basic_block_sptr get_right_block ()` override
- `gr::basic_block_sptr get_right_block (int RF_channel)` override
- `size_t item_size ()` override

## Protected Member Functions inherited from [SignalSourceBase](#)

- [SignalSourceBase](#) ([ConfigurationInterface](#) const \*configuration, std::string role, std::string impl)  
*Constructor.*
- `size_t decode_item_type (std::string const &item_type, bool *is_interleaved=nullptr, bool throw_on_↵ error=false)`  
*utility for decoding passed ".item\_type" values*

### 10.256.1 Detailed Description

Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).

Definition at line 40 of file [ion\\_gsms\\_signal\\_source.h](#).

### 10.256.2 Member Function Documentation

#### 10.256.2.1 connect()

```
void IONGSMSSignalSource::connect (
    gr::top_block_sptr top_block) [override], [protected], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.256.2.2 disconnect()

```
void IONGSMSSignalSource::disconnect (
    gr::top_block_sptr top_block) [override], [protected], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.256.2.3 get\_left\_block()

```
gr::basic_block_sptr IONGSMSSignalSource::get_left_block () [override], [protected], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.256.2.4 get\_right\_block() [1/2]

```
gr::basic_block_sptr IONGSMSSignalSource::get_right_block () [override], [protected], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.256.2.5 `get_right_block()` [2/2]

```
gr::basic_block_sptr IONGSMSSignalSource::get_right_block (
    int RF_channel) [override], [protected], [virtual]
```

Reimplemented from [GNSSBlockInterface](#).

### 10.256.2.6 `item_size()`

```
size_t IONGSMSSignalSource::item_size () [inline], [override], [protected], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 57 of file [ion\\_gsms\\_signal\\_source.h](#).

The documentation for this class was generated from the following file:

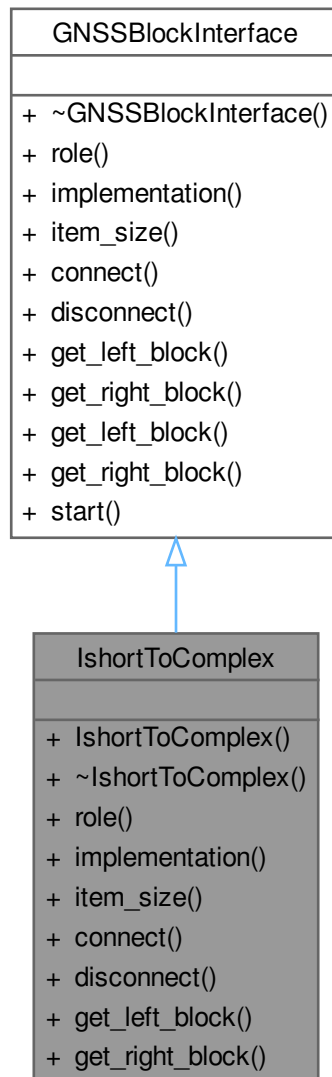
- [ion\\_gsms\\_signal\\_source.h](#)

## 10.257 IshortToComplex Class Reference

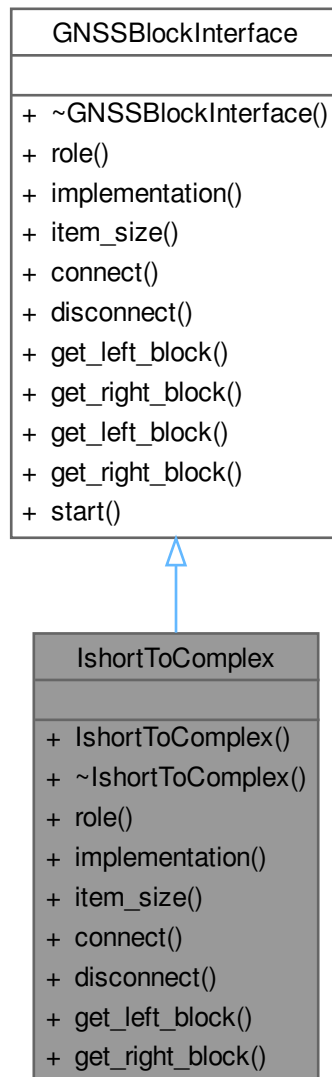
Adapts an I/Q interleaved short integer sample stream to a `gr_complex` (float) stream.

```
#include <ishort_to_complex.h>
```

Inheritance diagram for IshortToComplex:



Collaboration diagram for IshortToComplex:



## Public Member Functions

- **IshortToComplex** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "Ishort\_To\_Complex".
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual `gr::basic_block_sptr` [get\\_left\\_block](#) (int RF\_channel)
- virtual `gr::basic_block_sptr` [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.257.1 Detailed Description

Adapts an I/Q interleaved short integer sample stream to a `gr_complex` (float) stream.  
Definition at line 39 of file [ishort\\_to\\_complex.h](#).

### 10.257.2 Member Function Documentation

#### 10.257.2.1 `connect()`

```
void IshortToComplex::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.257.2.2 `disconnect()`

```
void IshortToComplex::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.257.2.3 `get_left_block()`

```
gr::basic_block_sptr IshortToComplex::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.257.2.4 `get_right_block()`

```
gr::basic_block_sptr IshortToComplex::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.257.2.5 `implementation()`

```
std::string IshortToComplex::implementation () [inline], [override], [virtual]
```

Returns "Ishort\_To\_Complex".

Implements [GNSSBlockInterface](#).

Definition at line 54 of file [ishort\\_to\\_complex.h](#).

#### 10.257.2.6 `item_size()`

```
size_t IshortToComplex::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 59 of file [ishort\\_to\\_complex.h](#).

#### 10.257.2.7 `role()`

```
std::string IshortToComplex::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 48 of file [ishort\\_to\\_complex.h](#).

The documentation for this class was generated from the following file:

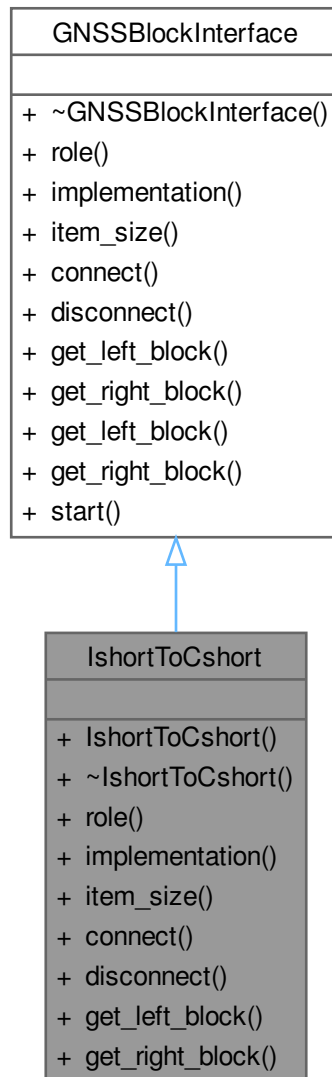
- [ishort\\_to\\_complex.h](#)

## 10.258 IshortToCshort Class Reference

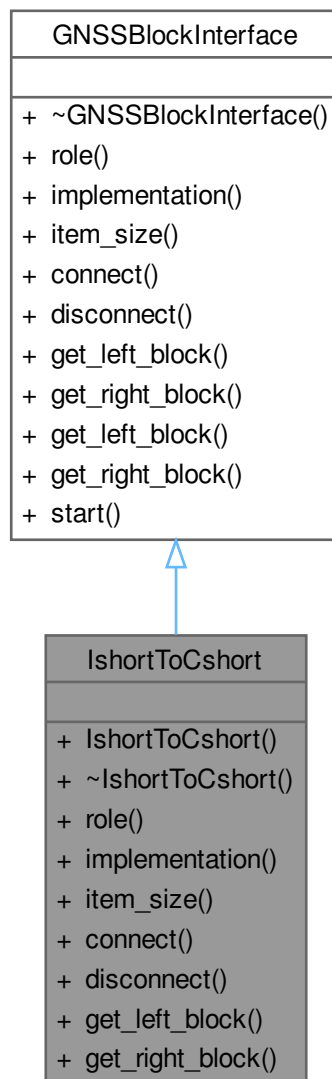
Adapts a short integer (16 bits) interleaved sample stream into a `std::complex<short>` stream.

```
#include <ishort_to_cshort.h>
```

Inheritance diagram for IshortToCshort:



Collaboration diagram for IshortToCshort:



### Public Member Functions

- **IshortToCshort** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "Ishort\_To\_Cshort".
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override



## Public Member Functions inherited from GNSSBlockInterface

- virtual `gr::basic_block_sptr` [get\\_left\\_block](#) (int RF\_channel)
- virtual `gr::basic_block_sptr` [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.258.1 Detailed Description

Adapts a short integer (16 bits) interleaved sample stream into a `std::complex<short>` stream.  
Definition at line 39 of file [ishort\\_to\\_cshort.h](#).

### 10.258.2 Member Function Documentation

#### 10.258.2.1 connect()

```
void IshortToCshort::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.258.2.2 disconnect()

```
void IshortToCshort::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.258.2.3 get\_left\_block()

```
gr::basic_block_sptr IshortToCshort::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.258.2.4 get\_right\_block()

```
gr::basic_block_sptr IshortToCshort::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.258.2.5 implementation()

```
std::string IshortToCshort::implementation () [inline], [override], [virtual]
```

Returns "Ishort\_To\_Cshort".

Implements [GNSSBlockInterface](#).

Definition at line 54 of file [ishort\\_to\\_cshort.h](#).

#### 10.258.2.6 item\_size()

```
size_t IshortToCshort::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 59 of file [ishort\\_to\\_cshort.h](#).

#### 10.258.2.7 role()

```
std::string IshortToCshort::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

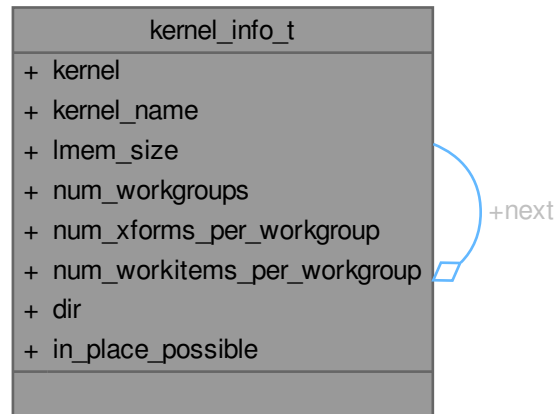
Definition at line 48 of file [ishort\\_to\\_cshort.h](#).

The documentation for this class was generated from the following file:

- [ishort\\_to\\_cshort.h](#)

## 10.259 kernel\_info\_t Struct Reference

Collaboration diagram for kernel\_info\_t:



### Public Attributes

- cl\_kernel [kernel](#)
- char \* [kernel\\_name](#)
- unsigned [lmem\\_size](#)
- unsigned [num\\_workgroups](#)
- unsigned [num\\_xforms\\_per\\_workgroup](#)
- unsigned [num\\_workitems\\_per\\_workgroup](#)
- cl\_fft\_kernel\_dir [dir](#)
- int [in\\_place\\_possible](#)
- [kernel\\_info\\_t](#) \* [next](#)

### 10.259.1 Detailed Description

Definition at line 32 of file [fft\\_internal.h](#).

### 10.259.2 Member Data Documentation

#### 10.259.2.1 dir

cl\_fft\_kernel\_dir kernel\_info\_t::dir

Definition at line 40 of file [fft\\_internal.h](#).

#### 10.259.2.2 in\_place\_possible

int kernel\_info\_t::in\_place\_possible

Definition at line 41 of file [fft\\_internal.h](#).

#### 10.259.2.3 kernel

cl\_kernel kernel\_info\_t::kernel

Definition at line 34 of file [fft\\_internal.h](#).

#### 10.259.2.4 kernel\_name

char\* kernel\_info\_t::kernel\_name

Definition at line 35 of file [fft\\_internal.h](#).

#### 10.259.2.5 lmem\_size

unsigned kernel\_info\_t::lmem\_size

Definition at line 36 of file [fft\\_internal.h](#).

#### 10.259.2.6 next

[kernel\\_info\\_t\\*](#) kernel\_info\_t::next

Definition at line 42 of file [fft\\_internal.h](#).

#### 10.259.2.7 num\_workgroups

unsigned kernel\_info\_t::num\_workgroups

Definition at line 37 of file [fft\\_internal.h](#).

#### 10.259.2.8 num\_workitems\_per\_workgroup

unsigned kernel\_info\_t::num\_workitems\_per\_workgroup

Definition at line 39 of file [fft\\_internal.h](#).

#### 10.259.2.9 num\_xforms\_per\_workgroup

unsigned kernel\_info\_t::num\_xforms\_per\_workgroup

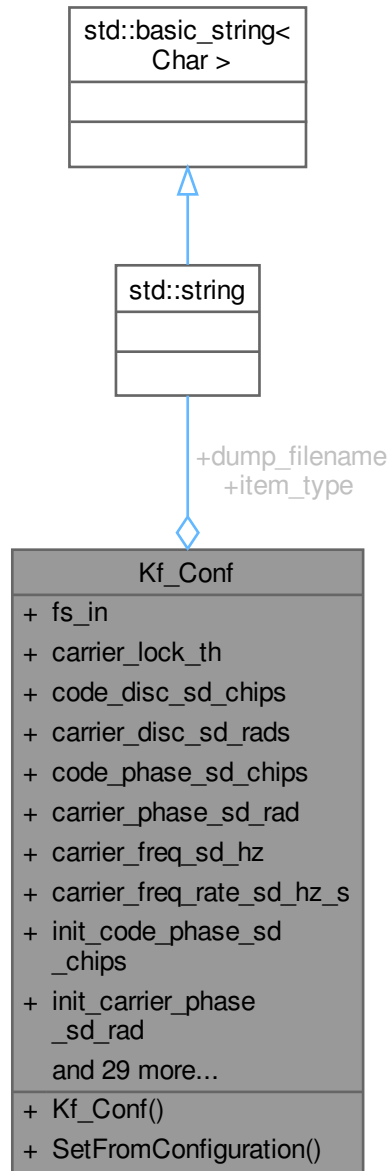
Definition at line 38 of file [fft\\_internal.h](#).

The documentation for this struct was generated from the following file:

- [fft\\_internal.h](#)

## 10.260 Kf\_Conf Class Reference

Collaboration diagram for Kf\_Conf:



### Public Member Functions

- void **SetFromConfiguration** (const [ConfigurationInterface](#) \*configuration, const std::string &role)

### Public Attributes

- std::string [item\\_type](#)
- std::string [dump\\_filename](#)
- double [fs\\_in](#)

- double [carrier\\_lock\\_th](#)
- double [code\\_disc\\_sd\\_chips](#)
- double [carrier\\_disc\\_sd\\_rads](#)
- double [code\\_phase\\_sd\\_chips](#)
- double [carrier\\_phase\\_sd\\_rad](#)
- double [carrier\\_freq\\_sd\\_hz](#)
- double [carrier\\_freq\\_rate\\_sd\\_hz\\_s](#)
- double [init\\_code\\_phase\\_sd\\_chips](#)
- double [init\\_carrier\\_phase\\_sd\\_rad](#)
- double [init\\_carrier\\_freq\\_sd\\_hz](#)
- double [init\\_carrier\\_freq\\_rate\\_sd\\_hz\\_s](#)
- float [early\\_late\\_space\\_chips](#)
- float [very\\_early\\_late\\_space\\_chips](#)
- float [early\\_late\\_space\\_narrow\\_chips](#)
- float [very\\_early\\_late\\_space\\_narrow\\_chips](#)
- float [slope](#)
- float [spc](#)
- float [y\\_intercept](#)
- float [cn0\\_smoother\\_alpha](#)
- float [carrier\\_lock\\_test\\_smoother\\_alpha](#)
- uint32\_t [pull\\_in\\_time\\_s](#)
- uint32\_t [bit\\_synchronization\\_time\\_limit\\_s](#)
- uint32\_t [vector\\_length](#)
- uint32\_t [smoother\\_length](#)
- int32\_t [extend\\_correlation\\_symbols](#)
- int32\_t [cn0\\_samples](#)
- int32\_t [cn0\\_smoother\\_samples](#)
- int32\_t [carrier\\_lock\\_test\\_smoother\\_samples](#)
- int32\_t [cn0\\_min](#)
- int32\_t [max\\_code\\_lock\\_fail](#)
- int32\_t [max\\_carrier\\_lock\\_fail](#)
- char [signal](#) [3] {}
- char [system](#)
- bool [track\\_pilot](#)
- bool [enable\\_doppler\\_correction](#)
- bool [high\\_dyn](#)
- bool [dump](#)
- bool [dump\\_mat](#)

### 10.260.1 Detailed Description

Definition at line 29 of file [kf\\_conf.h](#).

### 10.260.2 Member Data Documentation

#### 10.260.2.1 [bit\\_synchronization\\_time\\_limit\\_s](#)

uint32\_t Kf\_Conf::bit\_synchronization\_time\_limit\_s

Definition at line 67 of file [kf\\_conf.h](#).

#### 10.260.2.2 [carrier\\_disc\\_sd\\_rads](#)

double Kf\_Conf::carrier\_disc\_sd\_rads

Definition at line 43 of file [kf\\_conf.h](#).

#### 10.260.2.3 carrier\_freq\_rate\_sd\_hz\_s

double Kf\_Conf::carrier\_freq\_rate\_sd\_hz\_s  
Definition at line 49 of file [kf\\_conf.h](#).

#### 10.260.2.4 carrier\_freq\_sd\_hz

double Kf\_Conf::carrier\_freq\_sd\_hz  
Definition at line 48 of file [kf\\_conf.h](#).

#### 10.260.2.5 carrier\_lock\_test\_smoother\_alpha

float Kf\_Conf::carrier\_lock\_test\_smoother\_alpha  
Definition at line 65 of file [kf\\_conf.h](#).

#### 10.260.2.6 carrier\_lock\_test\_smoother\_samples

int32\_t Kf\_Conf::carrier\_lock\_test\_smoother\_samples  
Definition at line 73 of file [kf\\_conf.h](#).

#### 10.260.2.7 carrier\_lock\_th

double Kf\_Conf::carrier\_lock\_th  
Definition at line 38 of file [kf\\_conf.h](#).

#### 10.260.2.8 carrier\_phase\_sd\_rad

double Kf\_Conf::carrier\_phase\_sd\_rad  
Definition at line 47 of file [kf\\_conf.h](#).

#### 10.260.2.9 cn0\_min

int32\_t Kf\_Conf::cn0\_min  
Definition at line 74 of file [kf\\_conf.h](#).

#### 10.260.2.10 cn0\_samples

int32\_t Kf\_Conf::cn0\_samples  
Definition at line 71 of file [kf\\_conf.h](#).

#### 10.260.2.11 cn0\_smoother\_alpha

float Kf\_Conf::cn0\_smoother\_alpha  
Definition at line 64 of file [kf\\_conf.h](#).

#### 10.260.2.12 cn0\_smoother\_samples

int32\_t Kf\_Conf::cn0\_smoother\_samples  
Definition at line 72 of file [kf\\_conf.h](#).

#### 10.260.2.13 code\_disc\_sd\_chips

double Kf\_Conf::code\_disc\_sd\_chips  
Definition at line 42 of file [kf\\_conf.h](#).

#### 10.260.2.14 code\_phase\_sd\_chips

double Kf\_Conf::code\_phase\_sd\_chips  
Definition at line 46 of file [kf\\_conf.h](#).

### 10.260.2.15 dump

bool Kf\_Conf::dump

Definition at line 82 of file [kf\\_conf.h](#).

### 10.260.2.16 dump\_filename

std::string Kf\_Conf::dump\_filename

Definition at line 36 of file [kf\\_conf.h](#).

### 10.260.2.17 dump\_mat

bool Kf\_Conf::dump\_mat

Definition at line 83 of file [kf\\_conf.h](#).

### 10.260.2.18 early\_late\_space\_chips

float Kf\_Conf::early\_late\_space\_chips

Definition at line 57 of file [kf\\_conf.h](#).

### 10.260.2.19 early\_late\_space\_narrow\_chips

float Kf\_Conf::early\_late\_space\_narrow\_chips

Definition at line 59 of file [kf\\_conf.h](#).

### 10.260.2.20 enable\_doppler\_correction

bool Kf\_Conf::enable\_doppler\_correction

Definition at line 80 of file [kf\\_conf.h](#).

### 10.260.2.21 extend\_correlation\_symbols

int32\_t Kf\_Conf::extend\_correlation\_symbols

Definition at line 70 of file [kf\\_conf.h](#).

### 10.260.2.22 fs\_in

double Kf\_Conf::fs\_in

Definition at line 37 of file [kf\\_conf.h](#).

### 10.260.2.23 high\_dyn

bool Kf\_Conf::high\_dyn

Definition at line 81 of file [kf\\_conf.h](#).

### 10.260.2.24 init\_carrier\_freq\_rate\_sd\_hz\_s

double Kf\_Conf::init\_carrier\_freq\_rate\_sd\_hz\_s

Definition at line 55 of file [kf\\_conf.h](#).

### 10.260.2.25 init\_carrier\_freq\_sd\_hz

double Kf\_Conf::init\_carrier\_freq\_sd\_hz

Definition at line 54 of file [kf\\_conf.h](#).

### 10.260.2.26 init\_carrier\_phase\_sd\_rad

double Kf\_Conf::init\_carrier\_phase\_sd\_rad

Definition at line 53 of file [kf\\_conf.h](#).

#### 10.260.2.27 init\_code\_phase\_sd\_chips

double Kf\_Conf::init\_code\_phase\_sd\_chips  
Definition at line 52 of file [kf\\_conf.h](#).

#### 10.260.2.28 item\_type

std::string Kf\_Conf::item\_type  
Definition at line 35 of file [kf\\_conf.h](#).

#### 10.260.2.29 max\_carrier\_lock\_fail

int32\_t Kf\_Conf::max\_carrier\_lock\_fail  
Definition at line 76 of file [kf\\_conf.h](#).

#### 10.260.2.30 max\_code\_lock\_fail

int32\_t Kf\_Conf::max\_code\_lock\_fail  
Definition at line 75 of file [kf\\_conf.h](#).

#### 10.260.2.31 pull\_in\_time\_s

uint32\_t Kf\_Conf::pull\_in\_time\_s  
Definition at line 66 of file [kf\\_conf.h](#).

#### 10.260.2.32 signal

char Kf\_Conf::signal[3] {}  
Definition at line 77 of file [kf\\_conf.h](#).

#### 10.260.2.33 slope

float Kf\_Conf::slope  
Definition at line 61 of file [kf\\_conf.h](#).

#### 10.260.2.34 smoother\_length

uint32\_t Kf\_Conf::smoother\_length  
Definition at line 69 of file [kf\\_conf.h](#).

#### 10.260.2.35 spc

float Kf\_Conf::spc  
Definition at line 62 of file [kf\\_conf.h](#).

#### 10.260.2.36 system

char Kf\_Conf::system  
Definition at line 78 of file [kf\\_conf.h](#).

#### 10.260.2.37 track\_pilot

bool Kf\_Conf::track\_pilot  
Definition at line 79 of file [kf\\_conf.h](#).

#### 10.260.2.38 vector\_length

uint32\_t Kf\_Conf::vector\_length  
Definition at line 68 of file [kf\\_conf.h](#).



### 10.260.2.39 very\_early\_late\_space\_chips

float Kf\_Conf::very\_early\_late\_space\_chips

Definition at line 58 of file [kf\\_conf.h](#).

### 10.260.2.40 very\_early\_late\_space\_narrow\_chips

float Kf\_Conf::very\_early\_late\_space\_narrow\_chips

Definition at line 60 of file [kf\\_conf.h](#).

### 10.260.2.41 y\_intercept

float Kf\_Conf::y\_intercept

Definition at line 63 of file [kf\\_conf.h](#).

The documentation for this class was generated from the following file:

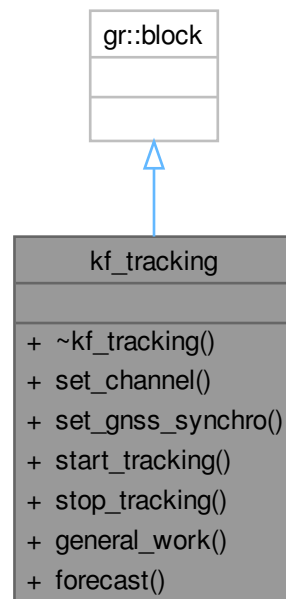
- [kf\\_conf.h](#)

## 10.261 kf\_tracking Class Reference

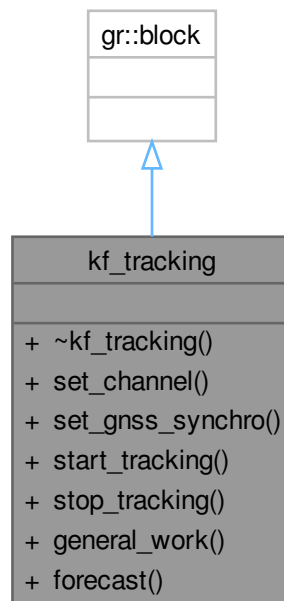
This class implements a code DLL + carrier PLL tracking block.

```
#include <kf_tracking.h>
```

Inheritance diagram for kf\_tracking:



Collaboration diagram for `kf_tracking`:



### Public Member Functions

- void **set\_channel** (uint32\_t channel)
- void **set\_gnss\_synchro** ([Gnss\\_Synchro](#) \*p\_gnss\_synchro)
- void **start\_tracking** ()
- void **stop\_tracking** ()
- int **general\_work** (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)
- void **forecast** (int noutput\_items, gr\_vector\_int &ninput\_items\_required)

### Friends

- `kf_tracking_sptr` **kf\_make\_tracking** (const [Kf\\_Conf](#) &conf\_)

### 10.261.1 Detailed Description

This class implements a code DLL + carrier PLL tracking block.

Definition at line 60 of file [kf\\_tracking.h](#).

The documentation for this class was generated from the following file:

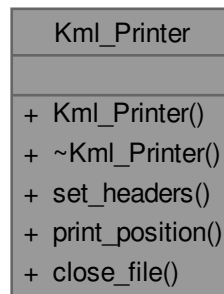
- `kf_tracking.h`

## 10.262 Kml\_Printer Class Reference

Prints PVT information to OGC KML format file (can be viewed with Google Earth)

```
#include <kml_printer.h>
```

Collaboration diagram for Kml\_Printer:



#### Public Member Functions

- **Kml\_Printer** (const std::string &base\_path=std::string("."))
- bool **set\_headers** (const std::string &filename, bool time\_tag\_name=true)
- bool **print\_position** (const [Pvt\\_Solution](#) \*const position)
- bool **close\_file** ()

#### 10.262.1 Detailed Description

Prints PVT information to OGC KML format file (can be viewed with Google Earth)

See <https://www.opengeospatial.org/standards/kml>

Definition at line 38 of file [kml\\_printer.h](#).

The documentation for this class was generated from the following file:

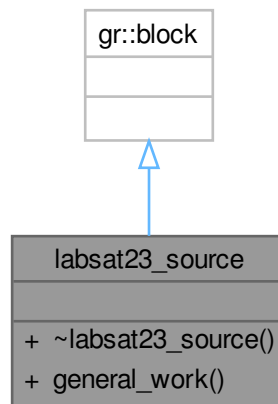
- [kml\\_printer.h](#)

## 10.263 labsat23\_source Class Reference

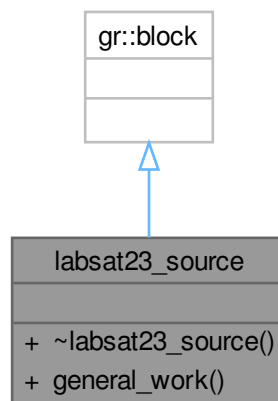
This class implements conversion between Labsat 2, 3 and 3 Wideband formats to gr\_complex.

```
#include <labsat23_source.h>
```

Inheritance diagram for labsat23\_source:



Collaboration diagram for labsat23\_source:



### Public Member Functions

- int **general\_work** (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)

### Friends

- labsat23\_source\_sptr **labsat23\_make\_source\_sptr** (const char \*signal\_file\_basename, const std::vector< int > &channel\_selector, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue, bool digital\_io\_enabled)

### 10.263.1 Detailed Description

This class implements conversion between Labsat 2, 3 and 3 Wideband formats to `gr_complex`.

Definition at line 52 of file [labsat23\\_source.h](#).

The documentation for this class was generated from the following file:

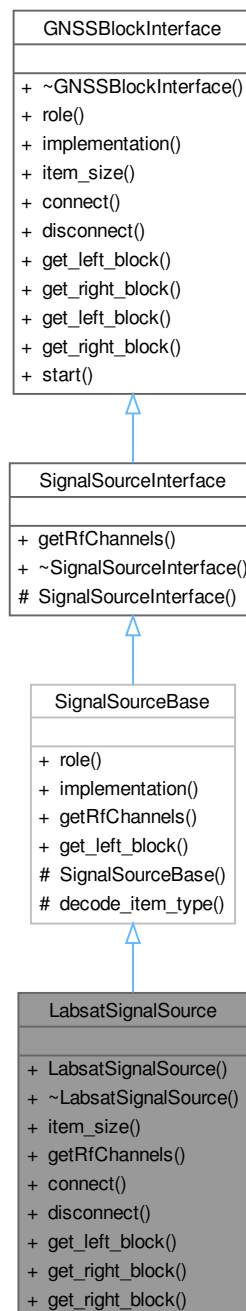
- [labsat23\\_source.h](#)

## 10.264 LabsatSignalSource Class Reference

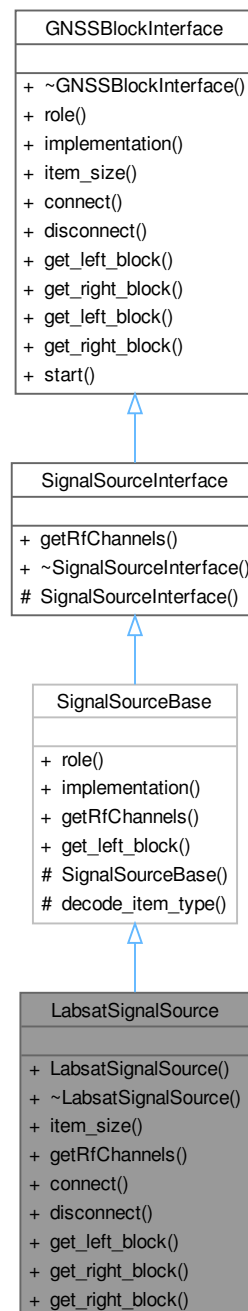
This class reads samples stored in LabSat version 2, 3, and 3 Wideband format.

```
#include <labsat_signal_source.h>
```

Inheritance diagram for LabsatSignalSource:



Collaboration diagram for LabsatSignalSource:



## Public Member Functions

- **LabsatSignalSource** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_stream, unsigned int out\_stream, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)
- size\_t [item\\_size](#) () override
- size\_t [getRfChannels](#) () const override
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override

- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) (int i) override

### Public Member Functions inherited from [SignalSourceBase](#)

- std::string [role](#) () final
- std::string [implementation](#) () final
- size\_t [getRfChannels](#) () const override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### Additional Inherited Members

### Protected Member Functions inherited from [SignalSourceBase](#)

- [SignalSourceBase](#) ([ConfigurationInterface](#) const \*configuration, std::string role, std::string impl)  
*Constructor.*
- size\_t [decode\\_item\\_type](#) (std::string const &item\_type, bool \*is\_interleaved=nullptr, bool throw\_on\_error=false)  
*utility for decoding passed ".item\_type" values*

## 10.264.1 Detailed Description

This class reads samples stored in LabSat version 2, 3, and 3 Wideband format.  
Definition at line 44 of file [labsat\\_signal\\_source.h](#).

## 10.264.2 Member Function Documentation

### 10.264.2.1 connect()

```
void LabsatSignalSource::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.264.2.2 disconnect()

```
void LabsatSignalSource::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.264.2.3 get\_left\_block()

```
gr::basic_block_sptr LabsatSignalSource::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.264.2.4 get\_right\_block() [1/2]

```
gr::basic_block_sptr LabsatSignalSource::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

**10.264.2.5 get\_right\_block()** [2/2]

```
gr::basic_block_sptr LabsatSignalSource::get_right_block (
    int i) [override], [virtual]
```

Reimplemented from [GNSSBlockInterface](#).

**10.264.2.6 getRfChannels()**

```
size_t LabsatSignalSource::getRfChannels () const [override], [virtual]
```

Implements [SignalSourceInterface](#).

**10.264.2.7 item\_size()**

```
size_t LabsatSignalSource::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

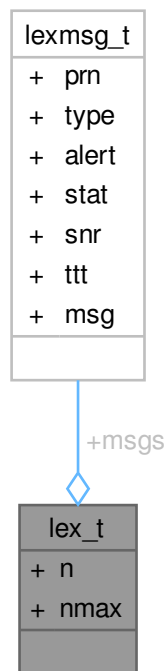
Definition at line 53 of file [labsat\\_signal\\_source.h](#).

The documentation for this class was generated from the following file:

- [labsat\\_signal\\_source.h](#)

**10.265 lex\_t Struct Reference**

Collaboration diagram for `lex_t`:

**Public Attributes**

- `int` [n](#)
- `int` [nmax](#)
- `lexmsg_t *` [msgs](#)



## 10.265.1 Detailed Description

Definition at line 688 of file [rtklib.h](#).

## 10.265.2 Member Data Documentation

### 10.265.2.1 msgs

[lexmsg\\_t\\*](#) [lex\\_t::msgs](#)

Definition at line 691 of file [rtklib.h](#).

### 10.265.2.2 n

int [lex\\_t::n](#)

Definition at line 690 of file [rtklib.h](#).

### 10.265.2.3 nmax

int [lex\\_t::nmax](#)

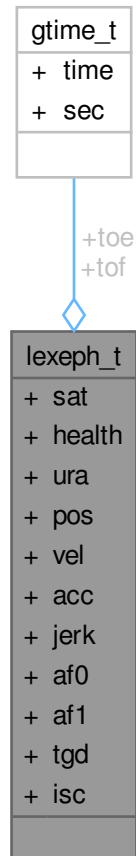
Definition at line 690 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.266 lexeph\_t Struct Reference

Collaboration diagram for `lexeph_t`:



### Public Attributes

- [gtime\\_t](#) `toe`
- [gtime\\_t](#) `tof`
- int `sat`
- unsigned char `health`
- unsigned char `ura`
- double `pos` [3]
- double `vel` [3]
- double `acc` [3]
- double `jerk` [3]
- double `af0`
- double `af1`
- double `tgd`
- double `isc` [8]

### 10.266.1 Detailed Description

Definition at line 695 of file [rtklib.h](#).

## 10.266.2 Member Data Documentation

### 10.266.2.1 acc

```
double lexeph_t::acc[3]
```

Definition at line 704 of file [rtklib.h](#).

### 10.266.2.2 af0

```
double lexeph_t::af0
```

Definition at line 706 of file [rtklib.h](#).

### 10.266.2.3 af1

```
double lexeph_t::af1
```

Definition at line 706 of file [rtklib.h](#).

### 10.266.2.4 health

```
unsigned char lexeph_t::health
```

Definition at line 700 of file [rtklib.h](#).

### 10.266.2.5 isc

```
double lexeph_t::isc[8]
```

Definition at line 708 of file [rtklib.h](#).

### 10.266.2.6 jerk

```
double lexeph_t::jerk[3]
```

Definition at line 705 of file [rtklib.h](#).

### 10.266.2.7 pos

```
double lexeph_t::pos[3]
```

Definition at line 702 of file [rtklib.h](#).

### 10.266.2.8 sat

```
int lexeph_t::sat
```

Definition at line 699 of file [rtklib.h](#).

### 10.266.2.9 tgd

```
double lexeph_t::tgd
```

Definition at line 707 of file [rtklib.h](#).

### 10.266.2.10 toe

```
gtime_t lexeph_t::toe
```

Definition at line 697 of file [rtklib.h](#).

### 10.266.2.11 tof

```
gtime_t lexeph_t::tof
```

Definition at line 698 of file [rtklib.h](#).

### 10.266.2.12 ura

```
unsigned char lexeph_t::ura
```

Definition at line 701 of file [rtklib.h](#).

### 10.266.2.13 vel

```
double lexeph_t::vel[3]
```

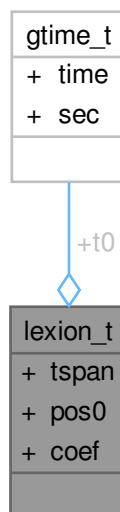
Definition at line 703 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.267 lexion\_t Struct Reference

Collaboration diagram for `lexion_t`:



### Public Attributes

- [gtime\\_t t0](#)
- double [tspan](#)
- double [pos0](#) [2]
- double [coef](#) [3][2]

### 10.267.1 Detailed Description

Definition at line 712 of file [rtklib.h](#).

### 10.267.2 Member Data Documentation

#### 10.267.2.1 coef

```
double lexion_t::coef[3][2]
```

Definition at line 717 of file [rtklib.h](#).

#### 10.267.2.2 pos0

```
double lexion_t::pos0[2]
```

Definition at line 716 of file [rtklib.h](#).

### 10.267.2.3 t0

`ptime_t lexion_t::t0`

Definition at line 714 of file [rtklib.h](#).

### 10.267.2.4 tspan

`double lexion_t::tspan`

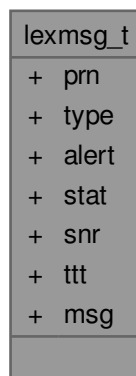
Definition at line 715 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.268 lexmsg\_t Struct Reference

Collaboration diagram for `lexmsg_t`:



### Public Attributes

- int [prn](#)
- int [type](#)
- int [alert](#)
- unsigned char [stat](#)
- unsigned char [snr](#)
- unsigned int [ttt](#)
- unsigned char [msg](#) [212]

### 10.268.1 Detailed Description

Definition at line 676 of file [rtklib.h](#).

### 10.268.2 Member Data Documentation

#### 10.268.2.1 alert

`int lexmsg_t::alert`

Definition at line 680 of file [rtklib.h](#).

#### 10.268.2.2 msg

unsigned char lexmsg\_t::msg[212]  
Definition at line 684 of file [rtklib.h](#).

#### 10.268.2.3 prn

int lexmsg\_t::prn  
Definition at line 678 of file [rtklib.h](#).

#### 10.268.2.4 snr

unsigned char lexmsg\_t::snr  
Definition at line 682 of file [rtklib.h](#).

#### 10.268.2.5 stat

unsigned char lexmsg\_t::stat  
Definition at line 681 of file [rtklib.h](#).

#### 10.268.2.6 ttt

unsigned int lexmsg\_t::ttt  
Definition at line 683 of file [rtklib.h](#).

#### 10.268.2.7 type

int lexmsg\_t::type  
Definition at line 679 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

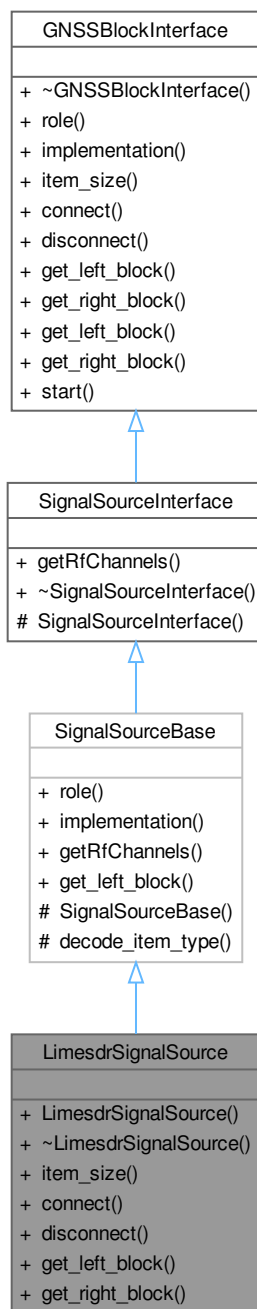
- [rtklib.h](#)

## 10.269 LimesdrSignalSource Class Reference

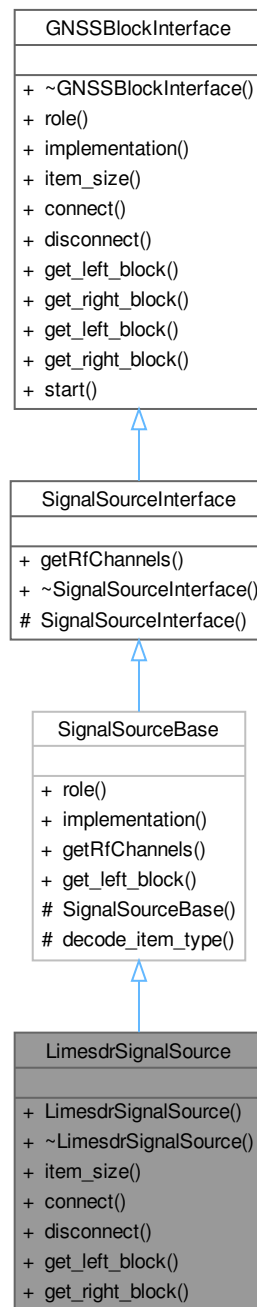
This class instantiates the LimeSDR gnuradio signal source. It has support also for a customized LimeSDR firmware and signal source to support PPS samplestamp reading.

```
#include <limesdr_signal_source.h>
```

Inheritance diagram for LimesdrSignalSource:



Collaboration diagram for LimesdrSignalSource:



## Public Member Functions

- **LimesdrSignalSource** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_stream, unsigned int out\_stream, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override



## Public Member Functions inherited from [SignalSourceBase](#)

- `std::string` [role](#) () final
- `std::string` [implementation](#) () final
- `size_t` [getRfChannels](#) () const override
- `gr::basic_block_sptr` [get\\_left\\_block](#) () override

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual `gr::basic_block_sptr` [get\\_left\\_block](#) (int RF\_channel)
- virtual `gr::basic_block_sptr` [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

## Additional Inherited Members

## Protected Member Functions inherited from [SignalSourceBase](#)

- [SignalSourceBase](#) ([ConfigurationInterface](#) const \*configuration, `std::string` role, `std::string` impl)  
*Constructor.*
- `size_t` [decode\\_item\\_type](#) (`std::string` const &item\_type, `bool` \*is\_interleaved=nullptr, `bool` throw\_on\_↔ error=false)  
*utility for decoding passed ".item\_type" values*

### 10.269.1 Detailed Description

This class instantiates the LimeSDR gnuradio signal source. It has support also for a customized LimeSDR firmware and signal source to support PPS samplestamp reading.  
Definition at line 42 of file [limesdr\\_signal\\_source.h](#).

### 10.269.2 Member Function Documentation

#### 10.269.2.1 connect()

```
void LimesdrSignalSource::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.269.2.2 disconnect()

```
void LimesdrSignalSource::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.269.2.3 get\_left\_block()

```
gr::basic_block_sptr LimesdrSignalSource::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.269.2.4 get\_right\_block()

```
gr::basic_block_sptr LimesdrSignalSource::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.269.2.5 item\_size()

size\_t LimesdrSignalSource::item\_size () [inline], [override], [virtual]

Implements [GNSSBlockInterface](#).

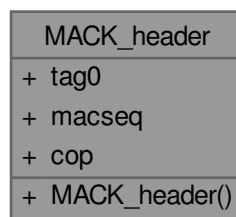
Definition at line 51 of file [limesdr\\_signal\\_source.h](#).

The documentation for this class was generated from the following file:

- [limesdr\\_signal\\_source.h](#)

## 10.270 MACK\_header Class Reference

Collaboration diagram for MACK\_header:



### Public Attributes

- uint64\_t [tag0](#) {}
- uint16\_t [macseq](#) {}
- uint8\_t [cop](#) {}

### 10.270.1 Detailed Description

Definition at line 53 of file [osnma\\_data.h](#).

### 10.270.2 Member Data Documentation

#### 10.270.2.1 cop

uint8\_t MACK\_header::cop {}

Definition at line 59 of file [osnma\\_data.h](#).

#### 10.270.2.2 macseq

uint16\_t MACK\_header::macseq {}

Definition at line 58 of file [osnma\\_data.h](#).

#### 10.270.2.3 tag0

uint64\_t MACK\_header::tag0 {}

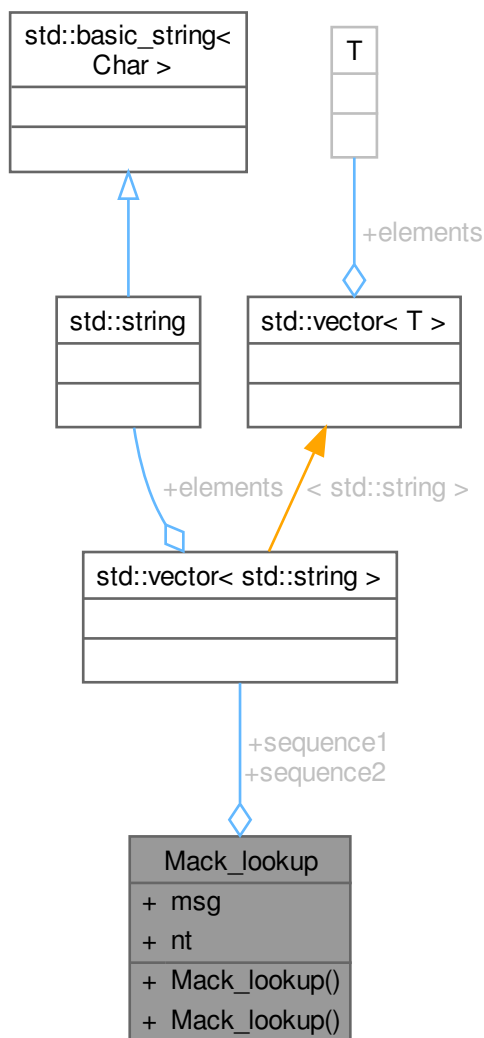
Definition at line 57 of file [osnma\\_data.h](#).

The documentation for this class was generated from the following file:

- [osnma\\_data.h](#)

## 10.271 Mack\_lookup Class Reference

Collaboration diagram for Mack\_lookup:



### Public Member Functions

- [Mack\\_lookup](#) (uint8\_t msg\_, uint8\_t nt\_, const std::vector< std::string > &s1\_, const std::vector< std::string > &s2\_)

### Public Attributes

- uint8\_t [msg](#) {}
- uint8\_t [nt](#) {}
- std::vector< std::string > [sequence1](#)
- std::vector< std::string > [sequence2](#)

### 10.271.1 Detailed Description

Definition at line 166 of file [Galileo\\_OSNMA.h](#).

### 10.271.2 Constructor & Destructor Documentation

#### 10.271.2.1 Mack\_lookup()

```
Mack_lookup::Mack_lookup (
    uint8_t msg_,
    uint8_t nt_,
    const std::vector< std::string > & s1_,
    const std::vector< std::string > & s2_) [inline]
```

Definition at line 170 of file [Galileo\\_OSNMA.h](#).

### 10.271.3 Member Data Documentation

#### 10.271.3.1 msg

```
uint8_t Mack_lookup::msg {}
```

Definition at line 177 of file [Galileo\\_OSNMA.h](#).

#### 10.271.3.2 nt

```
uint8_t Mack_lookup::nt {}
```

Definition at line 178 of file [Galileo\\_OSNMA.h](#).

#### 10.271.3.3 sequence1

```
std::vector<std::string> Mack_lookup::sequence1
```

Definition at line 179 of file [Galileo\\_OSNMA.h](#).

#### 10.271.3.4 sequence2

```
std::vector<std::string> Mack_lookup::sequence2
```

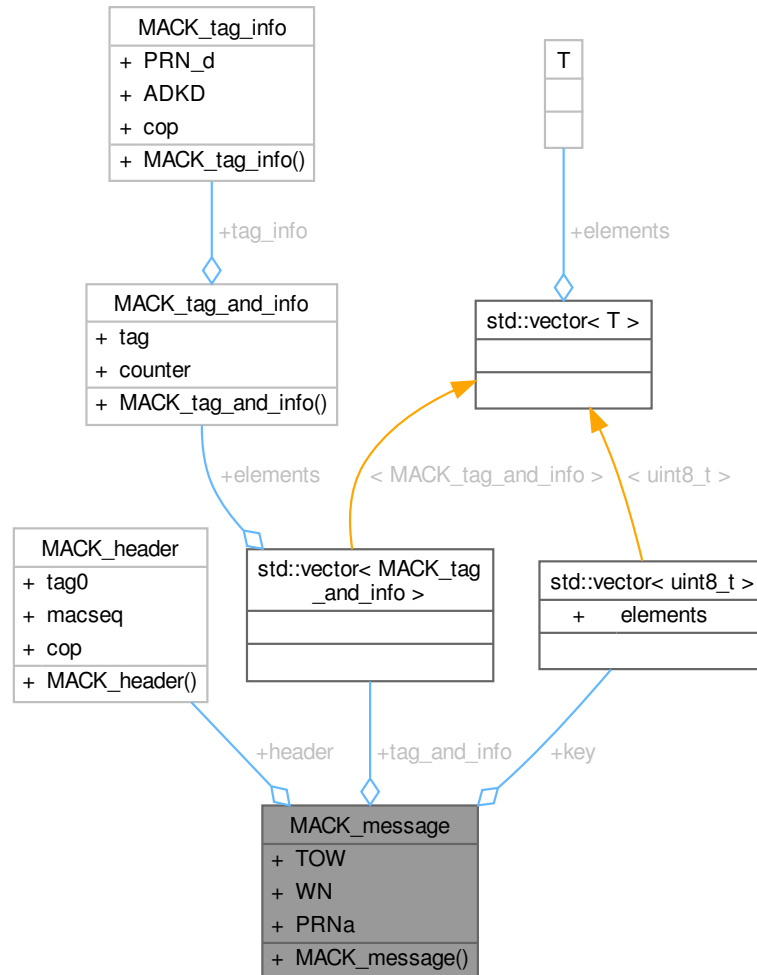
Definition at line 180 of file [Galileo\\_OSNMA.h](#).

The documentation for this class was generated from the following file:

- [Galileo\\_OSNMA.h](#)

## 10.272 MACK\_message Class Reference

Collaboration diagram for MACK\_message:



### Public Attributes

- [MACK\\_header](#) header
- `std::vector< MACK\_tag\_and\_info >` tag\_and\_info
- `std::vector< uint8_t >` key
- `uint32_t` TOW
- `uint32_t` WN
- `uint32_t` PRNa

### 10.272.1 Detailed Description

Definition at line 123 of file `osnma_data.h`.

## 10.272.2 Member Data Documentation

### 10.272.2.1 header

[MACK\\_header](#) MACK\_message::header

Definition at line 127 of file [osnma\\_data.h](#).

### 10.272.2.2 key

`std::vector<uint8_t>` MACK\_message::key

Definition at line 129 of file [osnma\\_data.h](#).

### 10.272.2.3 PRNa

`uint32_t` MACK\_message::PRNa

Definition at line 132 of file [osnma\\_data.h](#).

### 10.272.2.4 tag\_and\_info

`std::vector<MACK\_tag\_and\_info>` MACK\_message::tag\_and\_info

Definition at line 128 of file [osnma\\_data.h](#).

### 10.272.2.5 TOW

`uint32_t` MACK\_message::TOW

Definition at line 130 of file [osnma\\_data.h](#).

### 10.272.2.6 WN

`uint32_t` MACK\_message::WN

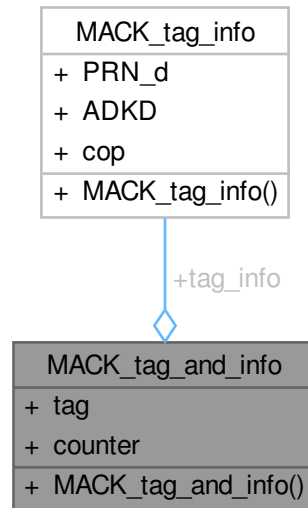
Definition at line 131 of file [osnma\\_data.h](#).

The documentation for this class was generated from the following file:

- [osnma\\_data.h](#)

## 10.273 MACK\_tag\_and\_info Class Reference

Collaboration diagram for MACK\_tag\_and\_info:



### Public Attributes

- `uint64_t tag`
- `MACK_tag_info tag_info`
- `uint32_t counter`

### 10.273.1 Detailed Description

Definition at line 73 of file [osnma\\_data.h](#).

### 10.273.2 Member Data Documentation

#### 10.273.2.1 counter

`uint32_t MACK_tag_and_info::counter`

Definition at line 79 of file [osnma\\_data.h](#).

#### 10.273.2.2 tag

`uint64_t MACK_tag_and_info::tag`

Definition at line 77 of file [osnma\\_data.h](#).

#### 10.273.2.3 tag\_info

`MACK_tag_info MACK_tag_and_info::tag_info`

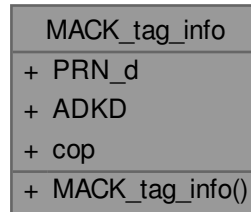
Definition at line 78 of file [osnma\\_data.h](#).

The documentation for this class was generated from the following file:

- [osnma\\_data.h](#)

## 10.274 MACK\_tag\_info Class Reference

Collaboration diagram for MACK\_tag\_info:



### Public Attributes

- uint8\_t [PRN\\_d](#) {}
- uint8\_t [ADKD](#) {}
- uint8\_t [cop](#) {}

### 10.274.1 Detailed Description

Definition at line 63 of file [osnma\\_data.h](#).

### 10.274.2 Member Data Documentation

#### 10.274.2.1 ADKD

```
uint8_t MACK_tag_info::ADKD {}
```

Definition at line 68 of file [osnma\\_data.h](#).

#### 10.274.2.2 cop

```
uint8_t MACK_tag_info::cop {}
```

Definition at line 69 of file [osnma\\_data.h](#).

#### 10.274.2.3 PRN\_d

```
uint8_t MACK_tag_info::PRN_d {}
```

Definition at line 67 of file [osnma\\_data.h](#).

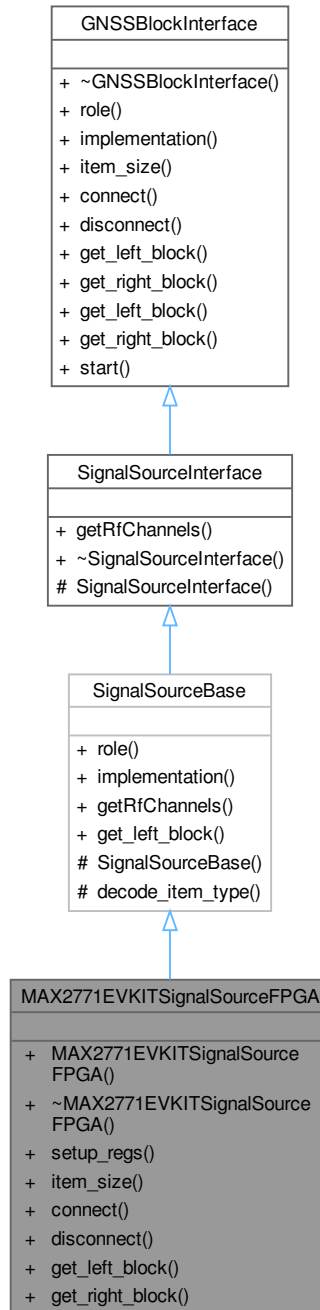
The documentation for this class was generated from the following file:

- [osnma\\_data.h](#)

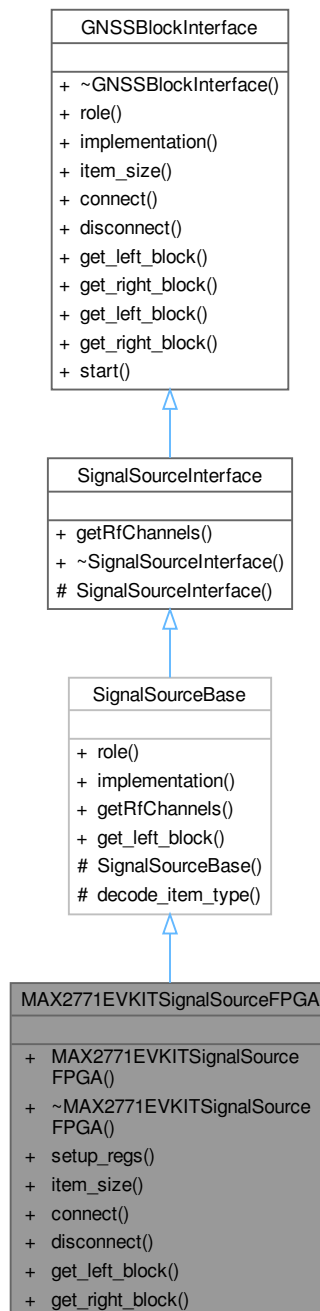


## 10.275 MAX2771EVKITSignalSourceFPGA Class Reference

Inheritance diagram for MAX2771EVKITSignalSourceFPGA:



Collaboration diagram for MAX2771EVKITSignalSourceFPGA:



## Public Member Functions

- **MAX2771EVKITSignalSourceFPGA** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_stream, unsigned int out\_stream, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)
- std::vector< uint32\_t > **setup\_regs** (void)
- size\_t **item\_size** () override
- void **connect** (gr::top\_block\_sptr top\_block) override
- void **disconnect** (gr::top\_block\_sptr top\_block) override

- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override

### Public Member Functions inherited from [SignalSourceBase](#)

- std::string [role](#) () final
- std::string [implementation](#) () final
- size\_t [getRfChannels](#) () const override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### Additional Inherited Members

### Protected Member Functions inherited from [SignalSourceBase](#)

- [SignalSourceBase](#) ([ConfigurationInterface](#) const \*configuration, std::string role, std::string impl)  
*Constructor.*
- size\_t [decode\\_item\\_type](#) (std::string const &item\_type, bool \*is\_interleaved=nullptr, bool throw\_on\_error=false)  
*utility for decoding passed ".item\_type" values*

## 10.275.1 Detailed Description

Definition at line 45 of file [max2771\\_evkit\\_signal\\_source\\_fpga.h](#).

## 10.275.2 Member Function Documentation

### 10.275.2.1 connect()

```
void MAX2771EVKITSignalSourceFPGA::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.275.2.2 disconnect()

```
void MAX2771EVKITSignalSourceFPGA::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.275.2.3 get\_left\_block()

```
gr::basic_block_sptr MAX2771EVKITSignalSourceFPGA::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.275.2.4 get\_right\_block()

```
gr::basic_block_sptr MAX2771EVKITSignalSourceFPGA::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.275.2.5 item\_size()

size\_t MAX2771EVKITSignalSourceFPGA::item\_size () [inline], [override], [virtual]

Implements [GNSSBlockInterface](#).

Definition at line 56 of file [max2771\\_evkit\\_signal\\_source\\_fpga.h](#).

The documentation for this class was generated from the following file:

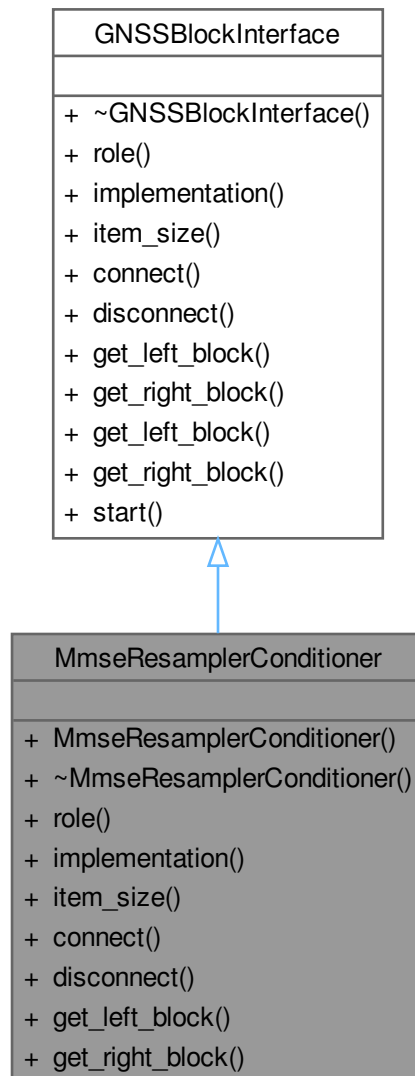
- [max2771\\_evkit\\_signal\\_source\\_fpga.h](#)

## 10.276 MmseResamplerConditioner Class Reference

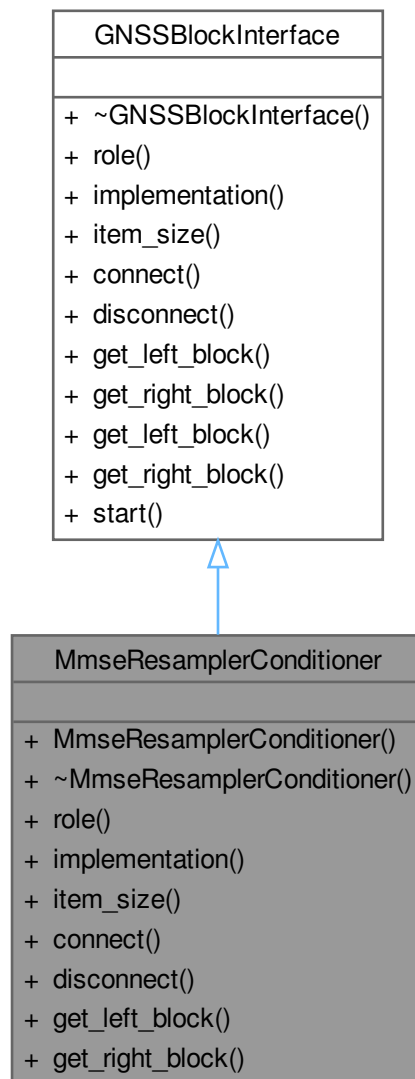
Interface of a MMSE resampler block adapter to a SignalConditionerInterface.

#include <mmse\_resampler\_conditioner.h>

Inheritance diagram for MmseResamplerConditioner:



Collaboration diagram for MmseResamplerConditioner:



## Public Member Functions

- **MmseResamplerConditioner** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_stream, unsigned int out\_stream)
- std::string [role](#) () override
- std::string [implementation](#) () override
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual `gr::basic_block_sptr` [get\\_left\\_block](#) (int RF\_channel)
- virtual `gr::basic_block_sptr` [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.276.1 Detailed Description

Interface of a MMSE resampler block adapter to a `SignalConditionerInterface`.  
Definition at line 48 of file [mmse\\_resampler\\_conditioner.h](#).

### 10.276.2 Member Function Documentation

#### 10.276.2.1 `connect()`

```
void MmseResamplerConditioner::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.276.2.2 `disconnect()`

```
void MmseResamplerConditioner::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.276.2.3 `get_left_block()`

```
gr::basic_block_sptr MmseResamplerConditioner::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.276.2.4 `get_right_block()`

```
gr::basic_block_sptr MmseResamplerConditioner::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.276.2.5 `implementation()`

```
std::string MmseResamplerConditioner::implementation () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 62 of file [mmse\\_resampler\\_conditioner.h](#).

#### 10.276.2.6 `item_size()`

```
size_t MmseResamplerConditioner::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 67 of file [mmse\\_resampler\\_conditioner.h](#).

#### 10.276.2.7 `role()`

```
std::string MmseResamplerConditioner::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

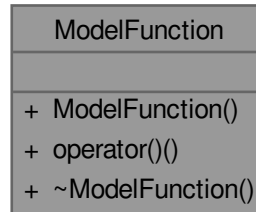
Definition at line 57 of file [mmse\\_resampler\\_conditioner.h](#).

The documentation for this class was generated from the following file:

- [mmse\\_resampler\\_conditioner.h](#)

## 10.277 ModelFunction Class Reference

Collaboration diagram for ModelFunction:



### Public Member Functions

- virtual arma::vec **operator()** (const arma::vec &input)=0

### 10.277.1 Detailed Description

Definition at line 46 of file [nonlinear\\_tracking.h](#).

### 10.277.2 Constructor & Destructor Documentation

#### 10.277.2.1 ModelFunction()

```
ModelFunction::ModelFunction () [inline]
```

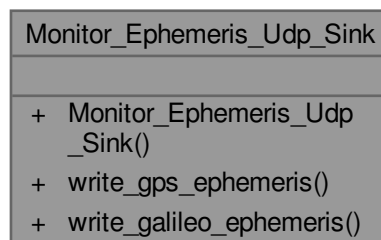
Definition at line 49 of file [nonlinear\\_tracking.h](#).

The documentation for this class was generated from the following file:

- [nonlinear\\_tracking.h](#)

## 10.278 Monitor\_Ephemeris\_Udp\_Sink Class Reference

Collaboration diagram for Monitor\_Ephemeris\_Udp\_Sink:



### Public Member Functions

- **Monitor\_Ephemeris\_Udp\_Sink** (const std::vector< std::string > &addresses, const uint16\_t &port, bool protobuf\_enabled)
- bool **write\_gps\_ephemeris** (const std::shared\_ptr< [Gps\\_Ephemeris](#) > &monitor\_gps\_eph)
- bool **write\_galileo\_ephemeris** (const std::shared\_ptr< [Galileo\\_Ephemeris](#) > &monitor\_gal\_eph)

### 10.278.1 Detailed Description

Definition at line 42 of file [monitor\\_ephemeris\\_udp\\_sink.h](#).

The documentation for this class was generated from the following file:

- [monitor\\_ephemeris\\_udp\\_sink.h](#)

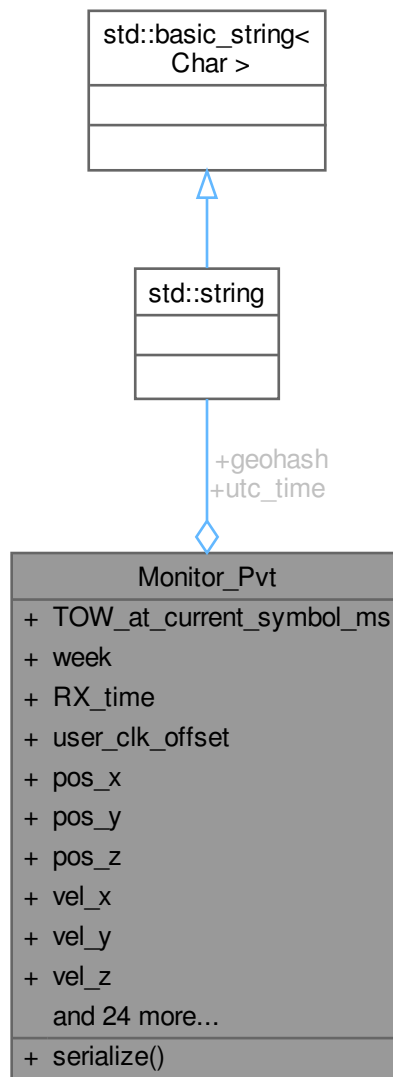
## 10.279 Monitor\_Pvt Class Reference

This class contains parameters and outputs of the PVT block.

```
#include <monitor_pvt.h>
```



Collaboration diagram for Monitor\_Pvt:



## Public Member Functions

- `template<class Archive>`  
void `serialize` (Archive &ar, const unsigned int version)  
*This member function serializes and restores [Monitor\\_Pvt](#) objects from a byte stream.*

## Public Attributes

- `uint32_t` `TOW_at_current_symbol_ms`
- `uint32_t` `week`
- `double` `RX_time`
- `double` `user_clk_offset`
- `double` `pos_x`
- `double` `pos_y`

- double [pos\\_z](#)
- double [vel\\_x](#)
- double [vel\\_y](#)
- double [vel\\_z](#)
- double [cov\\_xx](#)
- double [cov\\_yy](#)
- double [cov\\_zz](#)
- double [cov\\_xy](#)
- double [cov\\_yz](#)
- double [cov\\_zx](#)
- double [latitude](#)
- double [longitude](#)
- double [height](#)
- double [vel\\_e](#)
- double [vel\\_n](#)
- double [vel\\_u](#)
- double [cog](#)
- uint32\_t [galhas\\_status](#)
- uint8\_t [valid\\_sats](#)
- uint8\_t [solution\\_status](#)
- uint8\_t [solution\\_type](#)
- float [AR\\_ratio\\_factor](#)
- float [AR\\_ratio\\_threshold](#)
- double [gdop](#)
- double [pdop](#)
- double [hdop](#)
- double [vdop](#)
- double [user\\_clk\\_drift\\_ppm](#)
- std::string [utc\\_time](#)
- std::string [geohash](#)

### 10.279.1 Detailed Description

This class contains parameters and outputs of the PVT block.

Definition at line 33 of file [monitor\\_pvt.h](#).

### 10.279.2 Member Function Documentation

#### 10.279.2.1 [serialize\(\)](#)

```
template<class Archive>
void Monitor_Pvt::serialize (
    Archive & ar,
    const unsigned int version) [inline]
```

This member function serializes and restores [Monitor\\_Pvt](#) objects from a byte stream.

Definition at line 109 of file [monitor\\_pvt.h](#).

### 10.279.3 Member Data Documentation

#### 10.279.3.1 [AR\\_ratio\\_factor](#)

```
float Monitor_Pvt::AR_ratio_factor
```

Definition at line 85 of file [monitor\\_pvt.h](#).

#### 10.279.3.2 [AR\\_ratio\\_threshold](#)

```
float Monitor_Pvt::AR_ratio_threshold
```

Definition at line 87 of file [monitor\\_pvt.h](#).

**10.279.3.3 cog**

```
double Monitor_Pvt::cog
```

Definition at line 73 of file [monitor\\_pvt.h](#).

**10.279.3.4 cov\_xx**

```
double Monitor_Pvt::cov_xx
```

Definition at line 54 of file [monitor\\_pvt.h](#).

**10.279.3.5 cov\_xy**

```
double Monitor_Pvt::cov_xy
```

Definition at line 57 of file [monitor\\_pvt.h](#).

**10.279.3.6 cov\_yy**

```
double Monitor_Pvt::cov_yy
```

Definition at line 55 of file [monitor\\_pvt.h](#).

**10.279.3.7 cov\_yz**

```
double Monitor_Pvt::cov_yz
```

Definition at line 58 of file [monitor\\_pvt.h](#).

**10.279.3.8 cov\_zx**

```
double Monitor_Pvt::cov_zx
```

Definition at line 59 of file [monitor\\_pvt.h](#).

**10.279.3.9 cov\_zz**

```
double Monitor_Pvt::cov_zz
```

Definition at line 56 of file [monitor\\_pvt.h](#).

**10.279.3.10 galhas\_status**

```
uint32_t Monitor_Pvt::galhas_status
```

Definition at line 76 of file [monitor\\_pvt.h](#).

**10.279.3.11 gdop**

```
double Monitor_Pvt::gdop
```

Definition at line 90 of file [monitor\\_pvt.h](#).

**10.279.3.12 geohash**

```
std::string Monitor_Pvt::geohash
```

Definition at line 101 of file [monitor\\_pvt.h](#).

**10.279.3.13 hdop**

```
double Monitor_Pvt::hdop
```

Definition at line 92 of file [monitor\\_pvt.h](#).

**10.279.3.14 height**

```
double Monitor_Pvt::height
```

Definition at line 66 of file [monitor\\_pvt.h](#).

**10.279.3.15 latitude**

double Monitor\_Pvt::latitude

Definition at line 62 of file [monitor\\_pvt.h](#).

**10.279.3.16 longitude**

double Monitor\_Pvt::longitude

Definition at line 64 of file [monitor\\_pvt.h](#).

**10.279.3.17 pdop**

double Monitor\_Pvt::pdop

Definition at line 91 of file [monitor\\_pvt.h](#).

**10.279.3.18 pos\_x**

double Monitor\_Pvt::pos\_x

Definition at line 46 of file [monitor\\_pvt.h](#).

**10.279.3.19 pos\_y**

double Monitor\_Pvt::pos\_y

Definition at line 47 of file [monitor\\_pvt.h](#).

**10.279.3.20 pos\_z**

double Monitor\_Pvt::pos\_z

Definition at line 48 of file [monitor\\_pvt.h](#).

**10.279.3.21 RX\_time**

double Monitor\_Pvt::RX\_time

Definition at line 41 of file [monitor\\_pvt.h](#).

**10.279.3.22 solution\_status**

uint8\_t Monitor\_Pvt::solution\_status

Definition at line 81 of file [monitor\\_pvt.h](#).

**10.279.3.23 solution\_type**

uint8\_t Monitor\_Pvt::solution\_type

Definition at line 83 of file [monitor\\_pvt.h](#).

**10.279.3.24 TOW\_at\_current\_symbol\_ms**

uint32\_t Monitor\_Pvt::TOW\_at\_current\_symbol\_ms

Definition at line 37 of file [monitor\\_pvt.h](#).

**10.279.3.25 user\_clk\_drift\_ppm**

double Monitor\_Pvt::user\_clk\_drift\_ppm

Definition at line 96 of file [monitor\\_pvt.h](#).

**10.279.3.26 user\_clk\_offset**

double Monitor\_Pvt::user\_clk\_offset

Definition at line 43 of file [monitor\\_pvt.h](#).

**10.279.3.27 utc\_time**

std::string Monitor\_Pvt::utc\_time  
Definition at line 99 of file [monitor\\_pvt.h](#).

**10.279.3.28 valid\_sats**

uint8\_t Monitor\_Pvt::valid\_sats  
Definition at line 79 of file [monitor\\_pvt.h](#).

**10.279.3.29 vdop**

double Monitor\_Pvt::vdop  
Definition at line 93 of file [monitor\\_pvt.h](#).

**10.279.3.30 vel\_e**

double Monitor\_Pvt::vel\_e  
Definition at line 68 of file [monitor\\_pvt.h](#).

**10.279.3.31 vel\_n**

double Monitor\_Pvt::vel\_n  
Definition at line 69 of file [monitor\\_pvt.h](#).

**10.279.3.32 vel\_u**

double Monitor\_Pvt::vel\_u  
Definition at line 70 of file [monitor\\_pvt.h](#).

**10.279.3.33 vel\_x**

double Monitor\_Pvt::vel\_x  
Definition at line 49 of file [monitor\\_pvt.h](#).

**10.279.3.34 vel\_y**

double Monitor\_Pvt::vel\_y  
Definition at line 50 of file [monitor\\_pvt.h](#).

**10.279.3.35 vel\_z**

double Monitor\_Pvt::vel\_z  
Definition at line 51 of file [monitor\\_pvt.h](#).

**10.279.3.36 week**

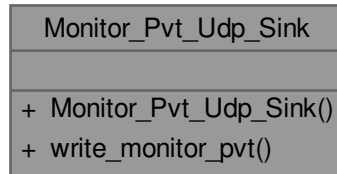
uint32\_t Monitor\_Pvt::week  
Definition at line 39 of file [monitor\\_pvt.h](#).

The documentation for this class was generated from the following file:

- [monitor\\_pvt.h](#)

## 10.280 Monitor\_Pvt\_Udp\_Sink Class Reference

Collaboration diagram for Monitor\_Pvt\_Udp\_Sink:



### Public Member Functions

- **Monitor\_Pvt\_Udp\_Sink** (const std::vector< std::string > &addresses, const std::vector< std::string > &ports, bool protobuf\_enabled)
- bool **write\_monitor\_pvt** (const [Monitor\\_Pvt](#) \*const monitor\_pvt)

### 10.280.1 Detailed Description

Definition at line 40 of file [monitor\\_pvt\\_udp\\_sink.h](#).

The documentation for this class was generated from the following file:

- [monitor\\_pvt\\_udp\\_sink.h](#)

## 10.281 msm\_h\_t Struct Reference

Collaboration diagram for msm\_h\_t:

msm_h_t
+ iod
+ time_s
+ clk_str
+ clk_ext
+ smooth
+ tint_s
+ nsat
+ nsig
+ sats
+ sigs
+ cellmask

### Public Attributes

- unsigned char [iod](#)
- unsigned char [time\\_s](#)
- unsigned char [clk\\_str](#)
- unsigned char [clk\\_ext](#)
- unsigned char [smooth](#)
- unsigned char [tint\\_s](#)
- unsigned char [nsat](#)
- unsigned char [nsig](#)
- unsigned char [sats](#) [64]
- unsigned char [sigs](#) [32]
- unsigned char [cellmask](#) [64]

### 10.281.1 Detailed Description

Definition at line [1276](#) of file [rtklib.h](#).

### 10.281.2 Member Data Documentation

#### 10.281.2.1 cellmask

unsigned char msm\_h\_t::cellmask[64]

Definition at line [1287](#) of file [rtklib.h](#).

#### 10.281.2.2 clk\_ext

unsigned char msm\_h\_t::clk\_ext

Definition at line [1281](#) of file [rtklib.h](#).

### 10.281.2.3 clk\_str

unsigned char msm\_h\_t::clk\_str  
Definition at line 1280 of file [rtklib.h](#).

### 10.281.2.4 iod

unsigned char msm\_h\_t::iod  
Definition at line 1278 of file [rtklib.h](#).

### 10.281.2.5 nsat

unsigned char msm\_h\_t::nsat  
Definition at line 1284 of file [rtklib.h](#).

### 10.281.2.6 nsig

unsigned char msm\_h\_t::nsig  
Definition at line 1284 of file [rtklib.h](#).

### 10.281.2.7 sats

unsigned char msm\_h\_t::sats[64]  
Definition at line 1285 of file [rtklib.h](#).

### 10.281.2.8 sigs

unsigned char msm\_h\_t::sigs[32]  
Definition at line 1286 of file [rtklib.h](#).

### 10.281.2.9 smooth

unsigned char msm\_h\_t::smooth  
Definition at line 1282 of file [rtklib.h](#).

### 10.281.2.10 time\_s

unsigned char msm\_h\_t::time\_s  
Definition at line 1279 of file [rtklib.h](#).

### 10.281.2.11 tint\_s

unsigned char msm\_h\_t::tint\_s  
Definition at line 1283 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)



## 10.282 mt1\_header Struct Reference

Collaboration diagram for mt1\_header:

mt1_header
<ul style="list-style-type: none"><li>+ toh</li><li>+ mask_id</li><li>+ iod_set_id</li><li>+ reserved</li><li>+ mask_flag</li><li>+ orbit_correction_flag</li><li>+ clock_fullset_flag</li><li>+ clock_subset_flag</li><li>+ code_bias_flag</li><li>+ phase_bias_flag</li></ul>

### Public Attributes

- uint16\_t [toh](#)
- uint8\_t [mask\\_id](#)
- uint8\_t [iod\\_set\\_id](#)
- uint8\_t [reserved](#)
- bool [mask\\_flag](#)
- bool [orbit\\_correction\\_flag](#)
- bool [clock\\_fullset\\_flag](#)
- bool [clock\\_subset\\_flag](#)
- bool [code\\_bias\\_flag](#)
- bool [phase\\_bias\\_flag](#)

### 10.282.1 Detailed Description

Definition at line 30 of file [galileo\\_has\\_data.h](#).

### 10.282.2 Member Data Documentation

#### 10.282.2.1 clock\_fullset\_flag

`bool mt1_header::clock_fullset_flag`

Definition at line 38 of file [galileo\\_has\\_data.h](#).

#### 10.282.2.2 clock\_subset\_flag

`bool mt1_header::clock_subset_flag`

Definition at line 39 of file [galileo\\_has\\_data.h](#).

#### 10.282.2.3 code\_bias\_flag

`bool mtl_header::code_bias_flag`

Definition at line 40 of file [galileo\\_has\\_data.h](#).

#### 10.282.2.4 iod\_set\_id

`uint8_t mtl_header::iod_set_id`

Definition at line 34 of file [galileo\\_has\\_data.h](#).

#### 10.282.2.5 mask\_flag

`bool mtl_header::mask_flag`

Definition at line 36 of file [galileo\\_has\\_data.h](#).

#### 10.282.2.6 mask\_id

`uint8_t mtl_header::mask_id`

Definition at line 33 of file [galileo\\_has\\_data.h](#).

#### 10.282.2.7 orbit\_correction\_flag

`bool mtl_header::orbit_correction_flag`

Definition at line 37 of file [galileo\\_has\\_data.h](#).

#### 10.282.2.8 phase\_bias\_flag

`bool mtl_header::phase_bias_flag`

Definition at line 41 of file [galileo\\_has\\_data.h](#).

#### 10.282.2.9 reserved

`uint8_t mtl_header::reserved`

Definition at line 35 of file [galileo\\_has\\_data.h](#).

#### 10.282.2.10 toh

`uint16_t mtl_header::toh`

Definition at line 32 of file [galileo\\_has\\_data.h](#).

The documentation for this struct was generated from the following file:

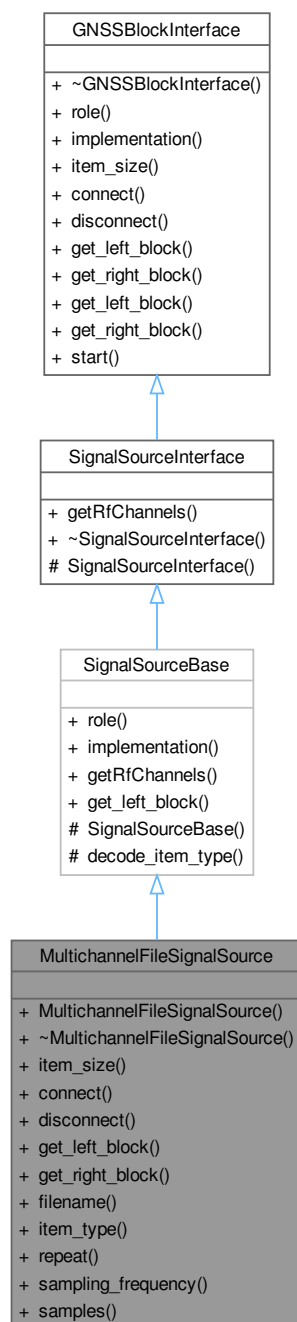
- [galileo\\_has\\_data.h](#)

## 10.283 MultichannelFileSignalSource Class Reference

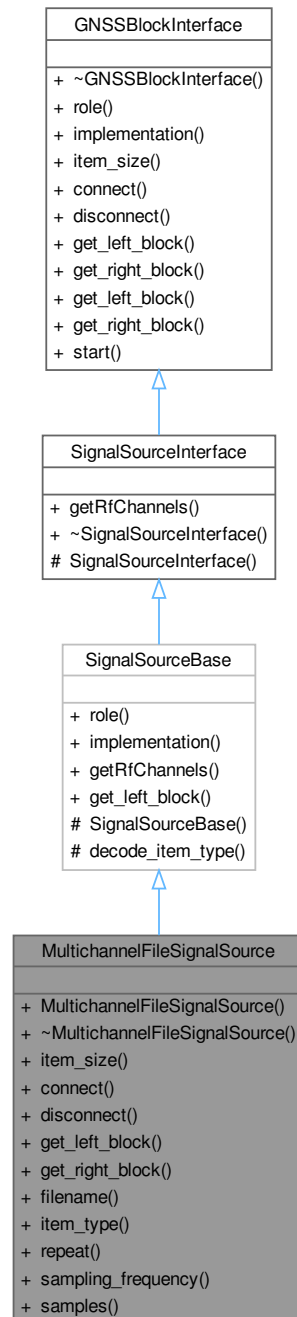
Class that reads signals samples from files at different frequency bands and adapts it to a [SignalSourceInterface](#).

```
#include <multichannel_file_signal_source.h>
```

Inheritance diagram for MultichannelFileSignalSource:



Collaboration diagram for MultichannelFileSignalSource:



## Public Member Functions

- **MultichannelFileSignalSource** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)
- `size_t item_size ()` override
- void `connect` (gr::top\_block\_sptr top\_block) override
- void `disconnect` (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr `get_left_block ()` override

- `gr::basic_block_sptr` [get\\_right\\_block](#) () override
- `std::string` [filename](#) () const
- `std::string` [item\\_type](#) () const
- `bool` [repeat](#) () const
- `int64_t` [sampling\\_frequency](#) () const
- `uint64_t` [samples](#) () const

### Public Member Functions inherited from [SignalSourceBase](#)

- `std::string` [role](#) () final
- `std::string` [implementation](#) () final
- `size_t` [getRfChannels](#) () const override
- `gr::basic_block_sptr` [get\\_left\\_block](#) () override

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual `gr::basic_block_sptr` [get\\_left\\_block](#) (int RF\_channel)
- virtual `gr::basic_block_sptr` [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### Additional Inherited Members

### Protected Member Functions inherited from [SignalSourceBase](#)

- [SignalSourceBase](#) ([ConfigurationInterface](#) const \*configuration, `std::string` role, `std::string` impl)  
*Constructor.*
- `size_t` [decode\\_item\\_type](#) (`std::string` const &item\_type, `bool` \*is\_interleaved=nullptr, `bool` throw\_on\_↵  
error=false)  
*utility for decoding passed ".item\_type" values*

## 10.283.1 Detailed Description

Class that reads signals samples from files at different frequency bands and adapts it to a [SignalSourceInterface](#).  
Definition at line 49 of file [multichannel\\_file\\_signal\\_source.h](#).

## 10.283.2 Member Function Documentation

### 10.283.2.1 connect()

```
void MultichannelFileSignalSource::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.283.2.2 disconnect()

```
void MultichannelFileSignalSource::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.283.2.3 filename()

```
std::string MultichannelFileSignalSource::filename () const [inline]
```

Definition at line 68 of file [multichannel\\_file\\_signal\\_source.h](#).

### 10.283.2.4 get\_left\_block()

```
gr::basic_block_sptr MultichannelFileSignalSource::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

**10.283.2.5 get\_right\_block()**

`gr::basic_block_sptr MultichannelFileSignalSource::get_right_block () [override], [virtual]`  
 Implements [GNSSBlockInterface](#).

**10.283.2.6 item\_size()**

`size_t MultichannelFileSignalSource::item_size () [inline], [override], [virtual]`  
 Implements [GNSSBlockInterface](#).  
 Definition at line 58 of file [multichannel\\_file\\_signal\\_source.h](#).

**10.283.2.7 item\_type()**

`std::string MultichannelFileSignalSource::item_type () const [inline]`  
 Definition at line 73 of file [multichannel\\_file\\_signal\\_source.h](#).

**10.283.2.8 repeat()**

`bool MultichannelFileSignalSource::repeat () const [inline]`  
 Definition at line 78 of file [multichannel\\_file\\_signal\\_source.h](#).

**10.283.2.9 samples()**

`uint64_t MultichannelFileSignalSource::samples () const [inline]`  
 Definition at line 88 of file [multichannel\\_file\\_signal\\_source.h](#).

**10.283.2.10 sampling\_frequency()**

`int64_t MultichannelFileSignalSource::sampling_frequency () const [inline]`  
 Definition at line 83 of file [multichannel\\_file\\_signal\\_source.h](#).

The documentation for this class was generated from the following file:

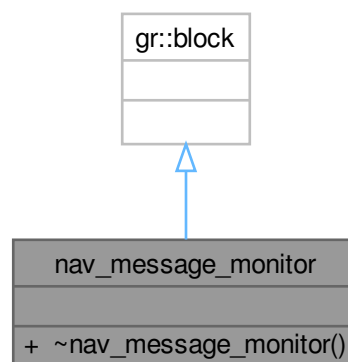
- [multichannel\\_file\\_signal\\_source.h](#)

**10.284 nav\_message\_monitor Class Reference**

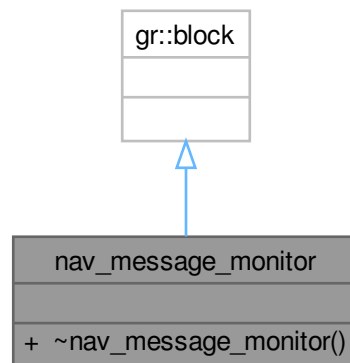
GNU Radio block that receives asynchronous [Nav\\_Message\\_Packet](#) objects from the telemetry blocks and sends them via UDP.

`#include <nav_message_monitor.h>`

Inheritance diagram for `nav_message_monitor`:



Collaboration diagram for nav\_message\_monitor:



### Public Member Functions

- `~nav_message_monitor()`=default  
*Default destructor.*

### Friends

- `nav_message_monitor_sptr nav_message_monitor_make` (const std::vector< std::string > &addresses, uint16\_t port)

## 10.284.1 Detailed Description

GNU Radio block that receives asynchronous [Nav\\_Message\\_Packet](#) objects from the telemetry blocks and sends them via UDP.

Definition at line 45 of file [nav\\_message\\_monitor.h](#).

## 10.284.2 Constructor & Destructor Documentation

### 10.284.2.1 ~nav\_message\_monitor()

```
nav_message_monitor::~nav_message_monitor () [default]
```

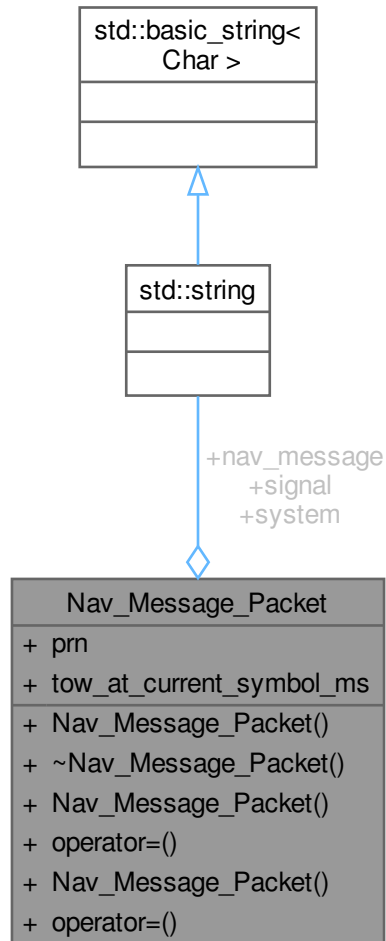
Default destructor.

The documentation for this class was generated from the following file:

- [nav\\_message\\_monitor.h](#)

## 10.285 Nav\_Message\_Packet Class Reference

Collaboration diagram for Nav\_Message\_Packet:



### Public Member Functions

- [Nav\\_Message\\_Packet](#) ()=default  
*Default constructor.*
- [~Nav\\_Message\\_Packet](#) ()=default  
*Default destructor.*
- [Nav\\_Message\\_Packet](#) (const [Nav\\_Message\\_Packet](#) &other) noexcept  
*Copy constructor.*
- [Nav\\_Message\\_Packet](#) & operator= (const [Nav\\_Message\\_Packet](#) &rhs) noexcept  
*Copy assignment operator.*
- [Nav\\_Message\\_Packet](#) ([Nav\\_Message\\_Packet](#) &&other) noexcept  
*Move constructor.*
- [Nav\\_Message\\_Packet](#) & operator= ([Nav\\_Message\\_Packet](#) &&other) noexcept  
*Move assignment operator.*



## Public Attributes

- `std::string` [system](#)  
GNSS constellation: "G" for GPS, "R" for Glonass, "S" for SBAS, "E" for Galileo and "C" for Beidou.
- `std::string` [signal](#)  
GNSS signal: "1C" for GPS L1 C/A, "1B" for Galileo E1b/c, "1G" for Glonass L1 C/A, "2S" for GPS L2 L2C(M), "2G" for Glonass L2 C/A, "L5" for GPS L5 and "5X" for Galileo E5a.
- `int32_t` [prn](#)  
SV ID.
- `int32_t` [tow\\_at\\_current\\_symbol\\_ms](#)  
Time of week of the current symbol, in ms.
- `std::string` [nav\\_message](#)  
Content of the navigation page.

## 10.285.1 Detailed Description

Definition at line 29 of file [nav\\_message\\_packet.h](#).

## 10.285.2 Constructor & Destructor Documentation

### 10.285.2.1 Nav\_Message\_Packet() [1/3]

```
Nav_Message_Packet::Nav_Message_Packet () [default]
```

Default constructor.

Referenced by [Nav\\_Message\\_Packet\(\)](#), [Nav\\_Message\\_Packet\(\)](#), [operator=\(\)](#), and [operator=\(\)](#).

### 10.285.2.2 ~Nav\_Message\_Packet()

```
Nav_Message_Packet::~Nav_Message_Packet () [default]
```

Default destructor.

### 10.285.2.3 Nav\_Message\_Packet() [2/3]

```
Nav_Message_Packet::Nav_Message_Packet (
    const Nav_Message_Packet & other) [inline], [noexcept]
```

Copy constructor.

Definition at line 43 of file [nav\\_message\\_packet.h](#).

References [Nav\\_Message\\_Packet\(\)](#).

Here is the call graph for this function:



### 10.285.2.4 Nav\_Message\_Packet() [3/3]

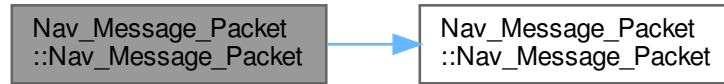
```
Nav_Message_Packet::Nav_Message_Packet (
    Nav_Message_Packet && other) [inline], [noexcept]
```

Move constructor.

Definition at line 64 of file [nav\\_message\\_packet.h](#).

References [Nav\\_Message\\_Packet\(\)](#).

Here is the call graph for this function:



## 10.285.3 Member Function Documentation

### 10.285.3.1 operator=() [1/2]

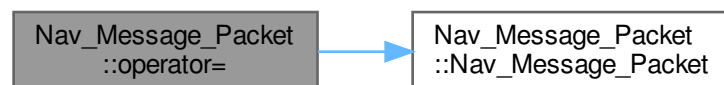
```
Nav_Message_Packet & Nav_Message_Packet::operator= (
    const Nav_Message_Packet & rhs) [inline], [noexcept]
```

Copy assignment operator.

Definition at line 49 of file [nav\\_message\\_packet.h](#).

References [Nav\\_Message\\_Packet\(\)](#).

Here is the call graph for this function:



### 10.285.3.2 operator=() [2/2]

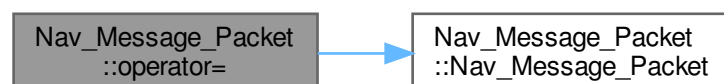
```
Nav_Message_Packet & Nav_Message_Packet::operator= (
    Nav_Message_Packet && other) [inline], [noexcept]
```

Move assignment operator.

Definition at line 70 of file [nav\\_message\\_packet.h](#).

References [Nav\\_Message\\_Packet\(\)](#).

Here is the call graph for this function:



## 10.285.4 Member Data Documentation

### 10.285.4.1 nav\_message

`std::string Nav_Message_Packet::nav_message`

Content of the navigation page.

Definition at line 40 of file [nav\\_message\\_packet.h](#).

Referenced by [Serdes\\_Nav\\_Message::readProtobuffer\(\)](#).

### 10.285.4.2 prn

`int32_t Nav_Message_Packet::prn`

SV ID.

Definition at line 38 of file [nav\\_message\\_packet.h](#).

Referenced by [Serdes\\_Nav\\_Message::readProtobuffer\(\)](#).

### 10.285.4.3 signal

`std::string Nav_Message_Packet::signal`

GNSS signal: "1C" for GPS L1 C/A, "1B" for Galileo E1b/c, "1G" for Glonass L1 C/A, "2S" for GPS L2 L2C(M), "2G" for Glonass L2 C/A, "L5" for GPS L5 and "5X" for Galileo E5a.

Definition at line 37 of file [nav\\_message\\_packet.h](#).

Referenced by [Serdes\\_Nav\\_Message::readProtobuffer\(\)](#).

### 10.285.4.4 system

`std::string Nav_Message_Packet::system`

GNSS constellation: "G" for GPS, "R" for Glonass, "S" for SBAS, "E" for Galileo and "C" for Beidou.

Definition at line 36 of file [nav\\_message\\_packet.h](#).

Referenced by [Serdes\\_Nav\\_Message::readProtobuffer\(\)](#).

### 10.285.4.5 tow\_at\_current\_symbol\_ms

`int32_t Nav_Message_Packet::tow_at_current_symbol_ms`

Time of week of the current symbol, in ms.

Definition at line 39 of file [nav\\_message\\_packet.h](#).

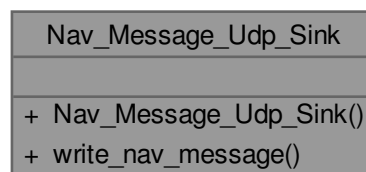
Referenced by [Serdes\\_Nav\\_Message::readProtobuffer\(\)](#).

The documentation for this class was generated from the following file:

- [nav\\_message\\_packet.h](#)

## 10.286 Nav\_Message\_Udp\_Sink Class Reference

Collaboration diagram for Nav\_Message\_Udp\_Sink:





- int [nmax](#)
- int [ng](#)
- int [ngmax](#)
- int [ns](#)
- int [nsmax](#)
- int [ne](#)
- int [nemax](#)
- int [nc](#)
- int [ncmax](#)
- int [na](#)
- int [namax](#)
- int [nt](#)
- int [ntmax](#)
- int [nf](#)
- int [nfmax](#)
- [eph\\_t](#) \* [eph](#)
- [geph\\_t](#) \* [geph](#)
- [seph\\_t](#) \* [seph](#)
- [peph\\_t](#) \* [peph](#)
- [pclt\\_t](#) \* [pclt](#)
- [alm\\_t](#) \* [alm](#)
- [tec\\_t](#) \* [tec](#)
- [fcbd\\_t](#) \* [fcb](#)
- [erp\\_t](#) [erp](#)
- double [utc\\_gps](#) [4]
- double [utc\\_glo](#) [4]
- double [utc\\_gal](#) [4]
- double [utc\\_qzs](#) [4]
- double [utc\\_cmp](#) [4]
- double [utc\\_irn](#) [4]
- double [utc\\_sbs](#) [4]
- double [ion\\_gps](#) [8]
- double [ion\\_gal](#) [4]
- double [ion\\_qzs](#) [8]
- double [ion\\_cmp](#) [8]
- double [ion\\_irn](#) [8]
- int [leaps](#)
- double [lam](#) [MAXSAT][NFREQ]
- double [cbias](#) [MAXSAT][3]
- double [rbias](#) [MAXRCV][2][3]
- double [wlbias](#) [MAXSAT]
- double [glo\\_cpbias](#) [4]
- char [glo\\_fcn](#) [MAXPRNGLO+1]
- [pcv\\_t](#) [pcvs](#) [MAXSAT]
- [sbssat\\_t](#) [sbssat](#)
- [sbsion\\_t](#) [sbsion](#) [MAXBAND+1]
- [dgps\\_t](#) [dgps](#) [MAXSAT]
- [ssr\\_t](#) [ssr](#) [MAXSAT]
- [lexeph\\_t](#) [lexeph](#) [MAXSAT]
- [lexion\\_t](#) [lexion](#)
- [pppcorr\\_t](#) [pppcorr](#)

### 10.287.1 Detailed Description

Definition at line 752 of file [rtklib.h](#).

## 10.287.2 Member Data Documentation

### 10.287.2.1 alm

[alm\\_t\\*](#) `nav_t::alm`

Definition at line 767 of file [rtklib.h](#).

### 10.287.2.2 cbias

`double nav_t::cbias[MAXSAT][3]`

Definition at line 785 of file [rtklib.h](#).

### 10.287.2.3 dgps

[dgps\\_t](#) `nav_t::dgps[MAXSAT]`

Definition at line 793 of file [rtklib.h](#).

### 10.287.2.4 eph

[eph\\_t\\*](#) `nav_t::eph`

Definition at line 762 of file [rtklib.h](#).

### 10.287.2.5 erp

[erp\\_t](#) `nav_t::erp`

Definition at line 770 of file [rtklib.h](#).

### 10.287.2.6 fcb

[fcbd\\_t\\*](#) `nav_t::fcb`

Definition at line 769 of file [rtklib.h](#).

### 10.287.2.7 geph

[geph\\_t\\*](#) `nav_t::geph`

Definition at line 763 of file [rtklib.h](#).

### 10.287.2.8 glo\_cpbias

`double nav_t::glo_cpbias[4]`

Definition at line 788 of file [rtklib.h](#).

### 10.287.2.9 glo\_fcn

`char nav_t::glo_fcn[MAXPRNGLO+1]`

Definition at line 789 of file [rtklib.h](#).

### 10.287.2.10 ion\_cmp

`double nav_t::ion_cmp[8]`

Definition at line 781 of file [rtklib.h](#).

### 10.287.2.11 ion\_gal

`double nav_t::ion_gal[4]`

Definition at line 779 of file [rtklib.h](#).

### 10.287.2.12 ion\_gps

`double nav_t::ion_gps[8]`

Definition at line 778 of file [rtklib.h](#).

**10.287.2.13 ion\_irn**

```
double nav_t::ion_irn[8]
```

Definition at line 782 of file [rtklib.h](#).

**10.287.2.14 ion\_qzs**

```
double nav_t::ion_qzs[8]
```

Definition at line 780 of file [rtklib.h](#).

**10.287.2.15 lam**

```
double nav_t::lam[MAXSAT][NFREQ]
```

Definition at line 784 of file [rtklib.h](#).

**10.287.2.16 leaps**

```
int nav_t::leaps
```

Definition at line 783 of file [rtklib.h](#).

**10.287.2.17 lexeph**

```
lexeph_t nav_t::lexeph[MAXSAT]
```

Definition at line 795 of file [rtklib.h](#).

**10.287.2.18 lexion**

```
lexion_t nav_t::lexion
```

Definition at line 796 of file [rtklib.h](#).

**10.287.2.19 n**

```
int nav_t::n
```

Definition at line 754 of file [rtklib.h](#).

**10.287.2.20 na**

```
int nav_t::na
```

Definition at line 759 of file [rtklib.h](#).

**10.287.2.21 namax**

```
int nav_t::namax
```

Definition at line 759 of file [rtklib.h](#).

**10.287.2.22 nc**

```
int nav_t::nc
```

Definition at line 758 of file [rtklib.h](#).

**10.287.2.23 ncmax**

```
int nav_t::ncmax
```

Definition at line 758 of file [rtklib.h](#).

**10.287.2.24 ne**

```
int nav_t::ne
```

Definition at line 757 of file [rtklib.h](#).

**10.287.2.25 nemax**

```
int nav_t::nemax
```

Definition at line 757 of file [rtklib.h](#).

**10.287.2.26 nf**

```
int nav_t::nf
```

Definition at line 761 of file [rtklib.h](#).

**10.287.2.27 nfmax**

```
int nav_t::nfmax
```

Definition at line 761 of file [rtklib.h](#).

**10.287.2.28 ng**

```
int nav_t::ng
```

Definition at line 755 of file [rtklib.h](#).

**10.287.2.29 ngmax**

```
int nav_t::ngmax
```

Definition at line 755 of file [rtklib.h](#).

**10.287.2.30 nmax**

```
int nav_t::nmax
```

Definition at line 754 of file [rtklib.h](#).

**10.287.2.31 ns**

```
int nav_t::ns
```

Definition at line 756 of file [rtklib.h](#).

**10.287.2.32 nsmax**

```
int nav_t::nsmax
```

Definition at line 756 of file [rtklib.h](#).

**10.287.2.33 nt**

```
int nav_t::nt
```

Definition at line 760 of file [rtklib.h](#).

**10.287.2.34 ntmax**

```
int nav_t::ntmax
```

Definition at line 760 of file [rtklib.h](#).

**10.287.2.35 pclk**

```
pclk_t* nav_t::pclk
```

Definition at line 766 of file [rtklib.h](#).

**10.287.2.36 pcvs**

```
pcv_t nav_t::pcvs[MAXSAT]
```

Definition at line 790 of file [rtklib.h](#).



**10.287.2.37 peph**

`peph_t* nav_t::peph`

Definition at line 765 of file [rtklib.h](#).

**10.287.2.38 pppcorr**

`pppcorr_t nav_t::pppcorr`

Definition at line 797 of file [rtklib.h](#).

**10.287.2.39 rbias**

`double nav_t::rbias[MAXRCV][2][3]`

Definition at line 786 of file [rtklib.h](#).

**10.287.2.40 sbsion**

`sbsion_t nav_t::sbsion[MAXBAND+1]`

Definition at line 792 of file [rtklib.h](#).

**10.287.2.41 sbssat**

`sbssat_t nav_t::sbssat`

Definition at line 791 of file [rtklib.h](#).

**10.287.2.42 seph**

`seph_t* nav_t::seph`

Definition at line 764 of file [rtklib.h](#).

**10.287.2.43 ssr**

`ssr_t nav_t::ssr[MAXSAT]`

Definition at line 794 of file [rtklib.h](#).

**10.287.2.44 tec**

`tec_t* nav_t::tec`

Definition at line 768 of file [rtklib.h](#).

**10.287.2.45 utc\_cmp**

`double nav_t::utc_cmp[4]`

Definition at line 775 of file [rtklib.h](#).

**10.287.2.46 utc\_gal**

`double nav_t::utc_gal[4]`

Definition at line 773 of file [rtklib.h](#).

**10.287.2.47 utc\_glo**

`double nav_t::utc_glo[4]`

Definition at line 772 of file [rtklib.h](#).

**10.287.2.48 utc\_gps**

`double nav_t::utc_gps[4]`

Definition at line 771 of file [rtklib.h](#).

**10.287.2.49 utc\_irn**

```
double nav_t::utc_irn[4]
```

Definition at line 776 of file [rtklib.h](#).

**10.287.2.50 utc\_qzs**

```
double nav_t::utc_qzs[4]
```

Definition at line 774 of file [rtklib.h](#).

**10.287.2.51 utc\_sbs**

```
double nav_t::utc_sbs[4]
```

Definition at line 777 of file [rtklib.h](#).

**10.287.2.52 wlbias**

```
double nav_t::wlbias[MAXSAT]
```

Definition at line 787 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

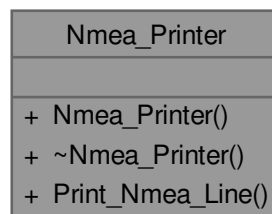
- [rtklib.h](#)

**10.288 Nmea\_Printer Class Reference**

This class provides a implementation of a subset of the NMEA-0183 standard for interfacing marine electronic devices as defined by the National Marine Electronics Association (NMEA).

```
#include <nmea_printer.h>
```

Collaboration diagram for Nmea\_Printer:

**Public Member Functions**

- [Nmea\\_Printer](#) (const std::string &filename, bool flag\_nmea\_output\_file, bool flag\_nmea\_tty\_port, std::string nmea\_dump\_devname, const std::string &base\_path=".")  
*Default constructor.*
- [~Nmea\\_Printer](#) ()  
*Default destructor.*
- bool [Print\\_Nmea\\_Line](#) (const [Rtklib\\_Solver](#) \*const pvt\_data)  
*Print NMEA PVT and satellite info to the initialized device.*

### 10.288.1 Detailed Description

This class provides a implementation of a subset of the NMEA-0183 standard for interfacing marine electronic devices as defined by the National Marine Electronics Association (NMEA).

See [https://en.wikipedia.org/wiki/NMEA\\_0183](https://en.wikipedia.org/wiki/NMEA_0183)

Definition at line 44 of file [nmea\\_printer.h](#).

### 10.288.2 Constructor & Destructor Documentation

#### 10.288.2.1 Nmea\_Printer()

```
Nmea_Printer::Nmea_Printer (  
    const std::string & filename,  
    bool flag_nmea_output_file,  
    bool flag_nmea_tty_port,  
    std::string nmea_dump_devname,  
    const std::string & base_path = ".")
```

Default constructor.

#### 10.288.2.2 ~Nmea\_Printer()

```
Nmea_Printer::~Nmea_Printer ()
```

Default destructor.

### 10.288.3 Member Function Documentation

#### 10.288.3.1 Print\_Nmea\_Line()

```
bool Nmea_Printer::Print_Nmea_Line (  
    const Rtklib\_Solver *const pvt_data)
```

Print NMEA PVT and satellite info to the initialized device.

The documentation for this class was generated from the following file:

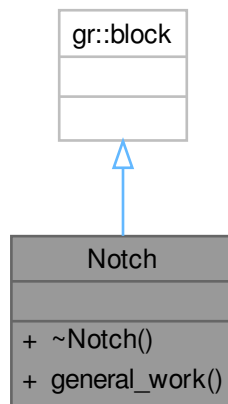
- [nmea\\_printer.h](#)

## 10.289 Notch Class Reference

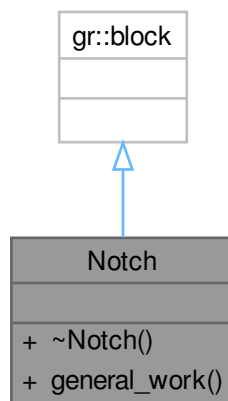
This class implements a real-time software-defined multi state notch filter.

```
#include <notch_cc.h>
```

Inheritance diagram for Notch:



Collaboration diagram for Notch:



### Public Member Functions

- int **general\_work** (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)

### Friends

- notch\_sptr **make\_notch\_filter** (float pfa, float p\_c\_factor, int32\_t length, int32\_t n\_segments\_est, int32\_t n\_segments\_reset)

## 10.289.1 Detailed Description

This class implements a real-time software-defined multi state notch filter.

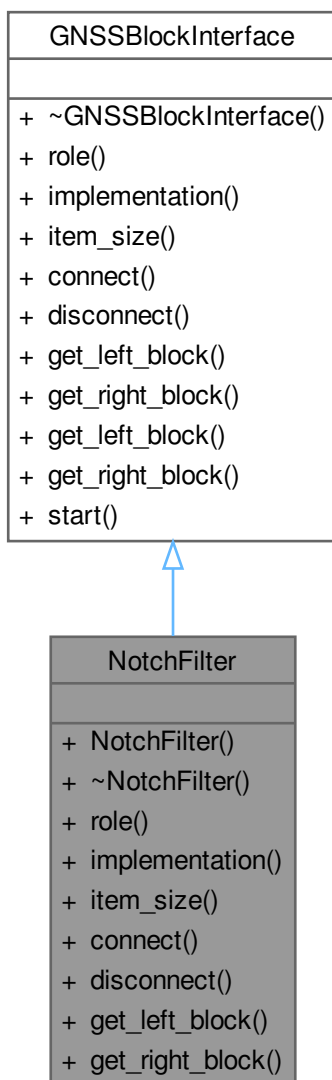
Definition at line 48 of file [notch\\_cc.h](#).

The documentation for this class was generated from the following file:

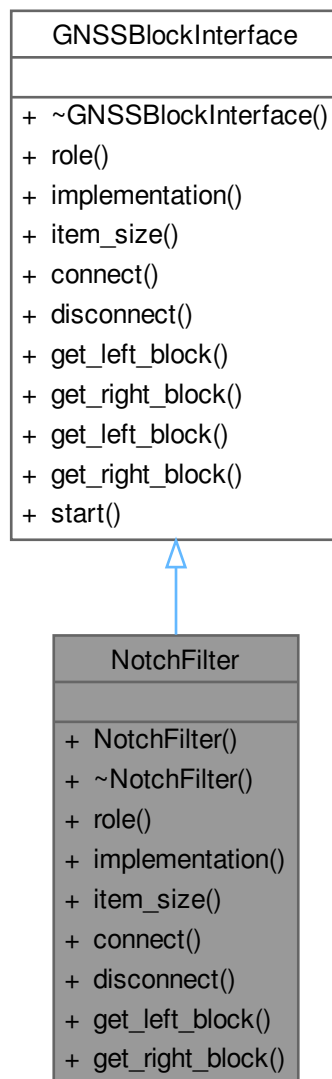
- [notch\\_cc.h](#)

## 10.290 NotchFilter Class Reference

Inheritance diagram for NotchFilter:



Collaboration diagram for NotchFilter:



### Public Member Functions

- **NotchFilter** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) ()
- std::string [implementation](#) ()  
Returns "Notch\_Filter".
- size\_t [item\\_size](#) ()
- void [connect](#) (gr::top\_block\_sptr top\_block)
- void [disconnect](#) (gr::top\_block\_sptr top\_block)
- gr::basic\_block\_sptr [get\\_left\\_block](#) ()
- gr::basic\_block\_sptr [get\\_right\\_block](#) ()

## Public Member Functions inherited from GNSSBlockInterface

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.290.1 Detailed Description

Definition at line 35 of file [notch\\_filter.h](#).

### 10.290.2 Member Function Documentation

#### 10.290.2.1 connect()

```
void NotchFilter::connect (
    gr::top_block_sptr top_block) [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.290.2.2 disconnect()

```
void NotchFilter::disconnect (
    gr::top_block_sptr top_block) [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.290.2.3 get\_left\_block()

```
gr::basic_block_sptr NotchFilter::get_left_block () [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.290.2.4 get\_right\_block()

```
gr::basic_block_sptr NotchFilter::get_right_block () [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.290.2.5 implementation()

```
std::string NotchFilter::implementation () [inline], [virtual]
```

Returns "Notch\_Filter".

Implements [GNSSBlockInterface](#).

Definition at line 50 of file [notch\\_filter.h](#).

#### 10.290.2.6 item\_size()

```
size_t NotchFilter::item_size () [inline], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 55 of file [notch\\_filter.h](#).

#### 10.290.2.7 role()

```
std::string NotchFilter::role () [inline], [virtual]
```

Implements [GNSSBlockInterface](#).

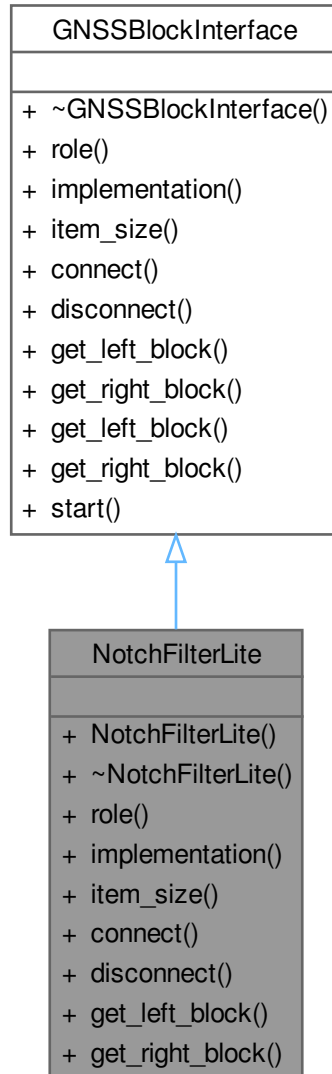
Definition at line 44 of file [notch\\_filter.h](#).

The documentation for this class was generated from the following file:

- [notch\\_filter.h](#)

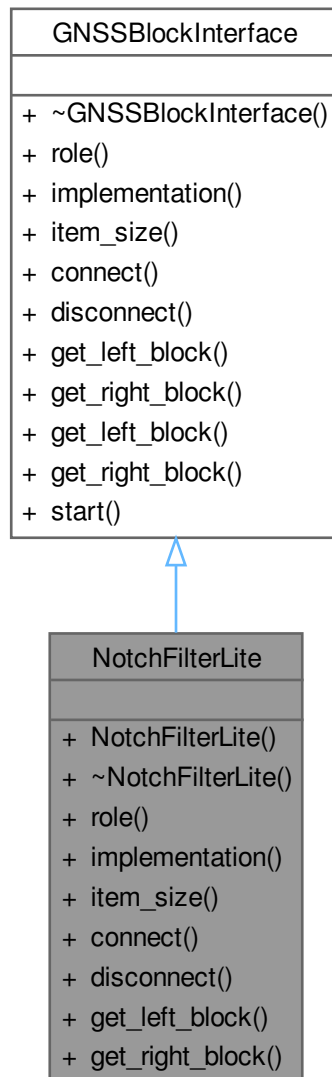
## 10.291 NotchFilterLite Class Reference

Inheritance diagram for NotchFilterLite:





Collaboration diagram for NotchFilterLite:



## Public Member Functions

- **NotchFilterLite** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) ()
- std::string [implementation](#) ()  
Returns "Notch\_Filter\_Lite".
- size\_t [item\\_size](#) ()
- void [connect](#) (gr::top\_block\_sptr top\_block)
- void [disconnect](#) (gr::top\_block\_sptr top\_block)
- gr::basic\_block\_sptr [get\\_left\\_block](#) ()
- gr::basic\_block\_sptr [get\\_right\\_block](#) ()

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual `gr::basic_block_sptr` [get\\_left\\_block](#) (int RF\_channel)
- virtual `gr::basic_block_sptr` [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.291.1 Detailed Description

Definition at line 35 of file [notch\\_filter\\_lite.h](#).

### 10.291.2 Member Function Documentation

#### 10.291.2.1 connect()

```
void NotchFilterLite::connect (
    gr::top_block_sptr top_block) [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.291.2.2 disconnect()

```
void NotchFilterLite::disconnect (
    gr::top_block_sptr top_block) [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.291.2.3 get\_left\_block()

```
gr::basic_block_sptr NotchFilterLite::get_left_block () [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.291.2.4 get\_right\_block()

```
gr::basic_block_sptr NotchFilterLite::get_right_block () [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.291.2.5 implementation()

```
std::string NotchFilterLite::implementation () [inline], [virtual]
```

Returns "Notch\_Filter\_Lite".

Implements [GNSSBlockInterface](#).

Definition at line 50 of file [notch\\_filter\\_lite.h](#).

#### 10.291.2.6 item\_size()

```
size_t NotchFilterLite::item_size () [inline], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 55 of file [notch\\_filter\\_lite.h](#).

#### 10.291.2.7 role()

```
std::string NotchFilterLite::role () [inline], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 44 of file [notch\\_filter\\_lite.h](#).

The documentation for this class was generated from the following file:

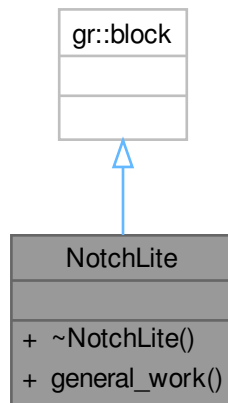
- [notch\\_filter\\_lite.h](#)

## 10.292 NotchLite Class Reference

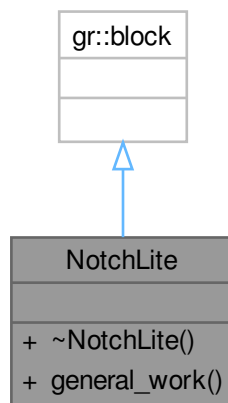
This class implements a real-time software-defined multi state notch filter light version.

```
#include <notch_lite_cc.h>
```

Inheritance diagram for NotchLite:



Collaboration diagram for NotchLite:



### Public Member Functions

- int **general\_work** (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)

### Friends

- notch\_lite\_sptr **make\_notch\_filter\_lite** (float p\_c\_factor, float pfa, int32\_t length, int32\_t n\_segments\_est, int32\_t n\_segments\_reset, int32\_t n\_segments\_coeff)

### 10.292.1 Detailed Description

This class implements a real-time software-defined multi state notch filter light version.

Definition at line 49 of file [notch\\_lite\\_cc.h](#).

The documentation for this class was generated from the following file:

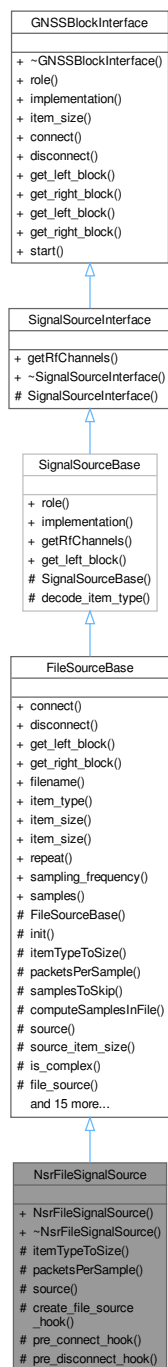
- [notch\\_lite\\_cc.h](#)

## 10.293 NsrFileSignalSource Class Reference

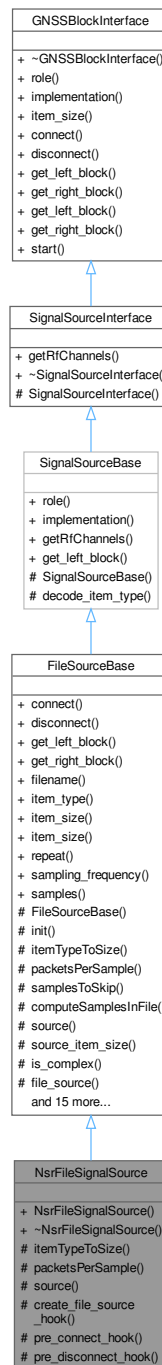
Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).

```
#include <nsr_file_signal_source.h>
```

Inheritance diagram for NsrFileSignalSource:



Collaboration diagram for NsrFileSignalSource:



## Public Member Functions

- **NsrFileSignalSource** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)

## Public Member Functions inherited from [FileSourceBase](#)

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override

- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- std::string [filename](#) () const  
*The file to read.*
- std::string [item\\_type](#) () const  
*The item type.*
- size\_t [item\\_size](#) () override  
*The configured size of each item.*
- virtual size\_t [item\\_size](#) () const
- bool [repeat](#) () const  
*Whether to repeat reading after end-of-file.*
- int64\_t [sampling\\_frequency](#) () const  
*The sampling frequency of the source file.*
- uint64\_t [samples](#) () const  
*The number of samples in the file.*

### Public Member Functions inherited from [SignalSourceBase](#)

- std::string [role](#) () final
- std::string [implementation](#) () final
- size\_t [getRfChannels](#) () const override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()  
*Start the flow of samples if needed.*

### Protected Member Functions

- std::tuple< size\_t, bool > [itemTypeToSize](#) () override  
*Compute the item size, from the [item\\_type](#)(). Subclasses may constrain types that don't make.*
- double [packetsPerSample](#) () const override  
*The number of (possibly unpacked) samples in a (raw) file sample (default=1)*
- gnss\_shared\_ptr< gr::block > [source](#) () const override  
*Abstracted front-end source. Sub-classes may override if they create specialized chains to decode source files into a usable format.*
- void [create\\_file\\_source\\_hook](#) () override
- void [pre\\_connect\\_hook](#) (gr::top\_block\_sptr top\_block) override
- void [pre\\_disconnect\\_hook](#) (gr::top\_block\_sptr top\_block) override

### Protected Member Functions inherited from [FileSourceBase](#)

- [FileSourceBase](#) ([ConfigurationInterface](#) const \*configuration, std::string const &role, std::string impl, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue, std::string default\_item\_type="short")  
*Constructor.*
- void [init](#) ()  
*Perform post-construction initialization.*
- virtual size\_t [samplesToSkip](#) () const  
*Compute the number of samples to skip.*
- size\_t [computeSamplesInFile](#) () const  
*Compute the number of samples in the file.*

- virtual size\_t [source\\_item\\_size](#) () const

*For complex source chains, the size of the file item may not be the same as the size of the.*

- bool **is\_complex** () const
- gnss\_shared\_ptr< gr::block > **file\_source** () const
- gnss\_shared\_ptr< gr::block > **valve** () const
- gnss\_shared\_ptr< gr::block > **throttle** () const
- gnss\_shared\_ptr< gr::block > **sink** () const
- gr::blocks::file\_source::sptr **create\_file\_source** ()
- gr::blocks::throttle::sptr **create\_throttle** ()
- gnss\_shared\_ptr< gr::block > **create\_valve** ()
- gr::blocks::file\_sink::sptr **create\_sink** ()
- virtual void **create\_throttle\_hook** ()
- virtual void **create\_valve\_hook** ()
- virtual void **create\_sink\_hook** ()
- virtual void **post\_connect\_hook** (gr::top\_block\_sptr top\_block)
- virtual void **post\_disconnect\_hook** (gr::top\_block\_sptr top\_block)

## Protected Member Functions inherited from [SignalSourceBase](#)

- [SignalSourceBase](#) ([ConfigurationInterface](#) const \*configuration, std::string role, std::string impl)  
*Constructor.*
- size\_t [decode\\_item\\_type](#) (std::string const &item\_type, bool \*is\_interleaved=nullptr, bool throw\_on\_↔ error=false)  
*utility for decoding passed ".item\_type" values*

### 10.293.1 Detailed Description

Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).

Definition at line 41 of file [nsr\\_file\\_signal\\_source.h](#).

### 10.293.2 Member Function Documentation

#### 10.293.2.1 [create\\_file\\_source\\_hook\(\)](#)

void NsrFileSignalSource::create\_file\_source\_hook () [override], [protected], [virtual]  
Reimplemented from [FileSourceBase](#).

#### 10.293.2.2 [itemTypeToSize\(\)](#)

std::tuple< size\_t, bool > NsrFileSignalSource::itemTypeToSize () [override], [protected], [virtual]

Compute the item size, from the [item\\_type\(\)](#). Subclasses may constrain types that don't make.

Reimplemented from [FileSourceBase](#).

#### 10.293.2.3 [packetsPerSample\(\)](#)

double NsrFileSignalSource::packetsPerSample () const [override], [protected], [virtual]

The number of (possibly unpacked) samples in a (raw) file sample (default=1)

Reimplemented from [FileSourceBase](#).

#### 10.293.2.4 [pre\\_connect\\_hook\(\)](#)

void NsrFileSignalSource::pre\_connect\_hook (gr::top\_block\_sptr top\_block) [override], [protected], [virtual]

Reimplemented from [FileSourceBase](#).



### 10.293.2.5 pre\_disconnect\_hook()

```
void NsrFileSignalSource::pre_disconnect_hook (  
    gr::top_block_sptr top_block) [override], [protected], [virtual]
```

Reimplemented from [FileSourceBase](#).

### 10.293.2.6 source()

```
gnss_shared_ptr< gr::block > NsrFileSignalSource::source () const [override], [protected],  
[virtual]
```

Abstracted front-end source. Sub-classes may override if they create specialized chains to decode source files into a usable format.

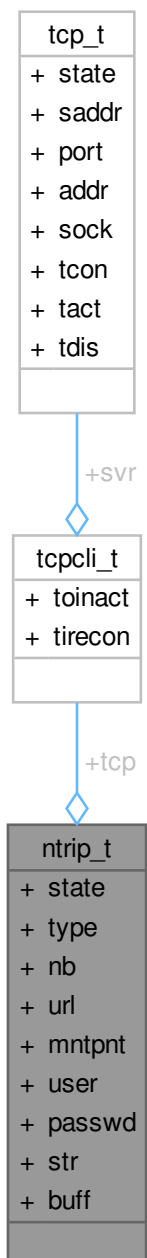
Reimplemented from [FileSourceBase](#).

The documentation for this class was generated from the following file:

- [nsr\\_file\\_signal\\_source.h](#)

## 10.294 ntrip\_t Struct Reference

Collaboration diagram for ntrip\_t:



### Public Attributes

- int `state`
- int `type`
- int `nb`
- char `url` [256]
- char `mntpnt` [256]

- char [user](#) [256]
- char [passwd](#) [256]
- char [str](#) [NTRIP\_MAXSTR]
- unsigned char [buff](#) [NTRIP\_MAXRSP]
- [tcpcli\\_t](#) \* [tcp](#)

### 10.294.1 Detailed Description

Definition at line [1169](#) of file [rtklib.h](#).

### 10.294.2 Member Data Documentation

#### 10.294.2.1 buff

```
unsigned char ntrip_t::buff[NTRIP_MAXRSP]
```

Definition at line [1179](#) of file [rtklib.h](#).

#### 10.294.2.2 mntpnt

```
char ntrip_t::mntpnt[256]
```

Definition at line [1175](#) of file [rtklib.h](#).

#### 10.294.2.3 nb

```
int ntrip_t::nb
```

Definition at line [1173](#) of file [rtklib.h](#).

#### 10.294.2.4 passwd

```
char ntrip_t::passwd[256]
```

Definition at line [1177](#) of file [rtklib.h](#).

#### 10.294.2.5 state

```
int ntrip_t::state
```

Definition at line [1171](#) of file [rtklib.h](#).

#### 10.294.2.6 str

```
char ntrip_t::str[NTRIP_MAXSTR]
```

Definition at line [1178](#) of file [rtklib.h](#).

#### 10.294.2.7 tcp

```
tcpcli\_t* ntrip_t::tcp
```

Definition at line [1180](#) of file [rtklib.h](#).

#### 10.294.2.8 type

```
int ntrip_t::type
```

Definition at line [1172](#) of file [rtklib.h](#).

#### 10.294.2.9 url

```
char ntrip_t::url[256]
```

Definition at line [1174](#) of file [rtklib.h](#).

### 10.294.2.10 user

```
char ntrip_t::user[256]
```

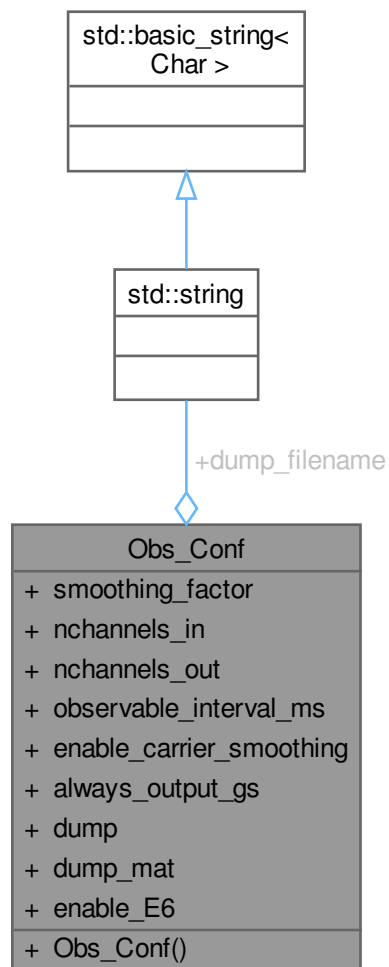
Definition at line 1176 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.295 Obs\_Conf Class Reference

Collaboration diagram for Obs\_Conf:



### Public Attributes

- `std::string` `dump_filename` {"obs\_dump.dat"}
- `int32_t` `smoothing_factor` {0}
- `uint32_t` `nchannels_in` {0U}
- `uint32_t` `nchannels_out` {0U}
- `uint32_t` `observable_interval_ms` {20U}

- bool [enable\\_carrier\\_smoothing](#) {false}
- bool [always\\_output\\_gs](#) {false}
- bool [dump](#) {false}
- bool [dump\\_mat](#) {false}
- bool [enable\\_E6](#) {false}

### 10.295.1 Detailed Description

Definition at line 30 of file [obs\\_conf.h](#).

### 10.295.2 Member Data Documentation

#### 10.295.2.1 always\_output\_gs

```
bool Obs_Conf::always_output_gs {false}
```

Definition at line 41 of file [obs\\_conf.h](#).

#### 10.295.2.2 dump

```
bool Obs_Conf::dump {false}
```

Definition at line 42 of file [obs\\_conf.h](#).

#### 10.295.2.3 dump\_filename

```
std::string Obs_Conf::dump_filename {"obs_dump.dat"}
```

Definition at line 35 of file [obs\\_conf.h](#).

#### 10.295.2.4 dump\_mat

```
bool Obs_Conf::dump_mat {false}
```

Definition at line 43 of file [obs\\_conf.h](#).

#### 10.295.2.5 enable\_carrier\_smoothing

```
bool Obs_Conf::enable_carrier_smoothing {false}
```

Definition at line 40 of file [obs\\_conf.h](#).

#### 10.295.2.6 enable\_E6

```
bool Obs_Conf::enable_E6 {false}
```

Definition at line 44 of file [obs\\_conf.h](#).

#### 10.295.2.7 nchannels\_in

```
uint32_t Obs_Conf::nchannels_in {0U}
```

Definition at line 37 of file [obs\\_conf.h](#).

#### 10.295.2.8 nchannels\_out

```
uint32_t Obs_Conf::nchannels_out {0U}
```

Definition at line 38 of file [obs\\_conf.h](#).

#### 10.295.2.9 observable\_interval\_ms

```
uint32_t Obs_Conf::observable_interval_ms {20U}
```

Definition at line 39 of file [obs\\_conf.h](#).

### 10.295.2.10 smoothing\_factor

```
int32_t Obs_Conf::smoothing_factor {0}
```

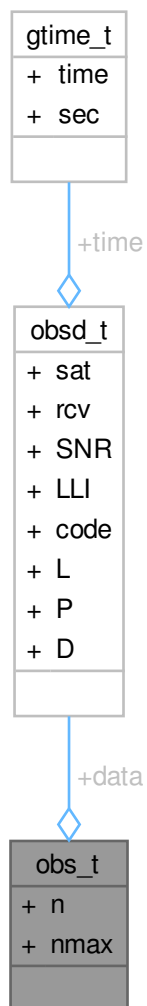
Definition at line 36 of file [obs\\_conf.h](#).

The documentation for this class was generated from the following file:

- [obs\\_conf.h](#)

## 10.296 obs\_t Struct Reference

Collaboration diagram for obs\_t:



### Public Attributes

- `int n`
- `int nmax`
- `obsd_t * data`

### 10.296.1 Detailed Description

Definition at line 375 of file [rtklib.h](#).

### 10.296.2 Member Data Documentation

#### 10.296.2.1 data

```
obsd_t* obs_t::data
```

Definition at line 378 of file [rtklib.h](#).

#### 10.296.2.2 n

```
int obs_t::n
```

Definition at line 377 of file [rtklib.h](#).

#### 10.296.2.3 nmax

```
int obs_t::nmax
```

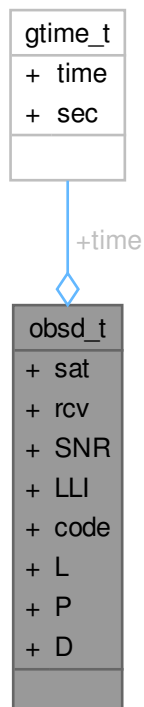
Definition at line 377 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.297 obsd\_t Struct Reference

Collaboration diagram for obsd\_t:



## Public Attributes

- [gtime\\_t](#) `time`
- unsigned char `sat`
- unsigned char `rcv`
- unsigned char `SNR` [`NFREQ+NEXOBS`]
- unsigned char `LLI` [`NFREQ+NEXOBS`]
- unsigned char `code` [`NFREQ+NEXOBS`]
- double `L` [`NFREQ+NEXOBS`]
- double `P` [`NFREQ+NEXOBS`]
- float `D` [`NFREQ+NEXOBS`]

### 10.297.1 Detailed Description

Definition at line 362 of file [rtklib.h](#).

### 10.297.2 Member Data Documentation

#### 10.297.2.1 `code`

```
unsigned char obsd_t::code[NFREQ+NEXOBS]
```

Definition at line 368 of file [rtklib.h](#).

#### 10.297.2.2 `D`

```
float obsd_t::D[NFREQ+NEXOBS]
```

Definition at line 371 of file [rtklib.h](#).

#### 10.297.2.3 `L`

```
double obsd_t::L[NFREQ+NEXOBS]
```

Definition at line 369 of file [rtklib.h](#).

#### 10.297.2.4 `LLI`

```
unsigned char obsd_t::LLI[NFREQ+NEXOBS]
```

Definition at line 367 of file [rtklib.h](#).

#### 10.297.2.5 `P`

```
double obsd_t::P[NFREQ+NEXOBS]
```

Definition at line 370 of file [rtklib.h](#).

#### 10.297.2.6 `rcv`

```
unsigned char obsd_t::rcv
```

Definition at line 365 of file [rtklib.h](#).

#### 10.297.2.7 `sat`

```
unsigned char obsd_t::sat
```

Definition at line 365 of file [rtklib.h](#).

#### 10.297.2.8 `SNR`

```
unsigned char obsd_t::SNR[NFREQ+NEXOBS]
```

Definition at line 366 of file [rtklib.h](#).



### 10.297.2.9 time

`ptime_t` `obsd_t::time`

Definition at line 364 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

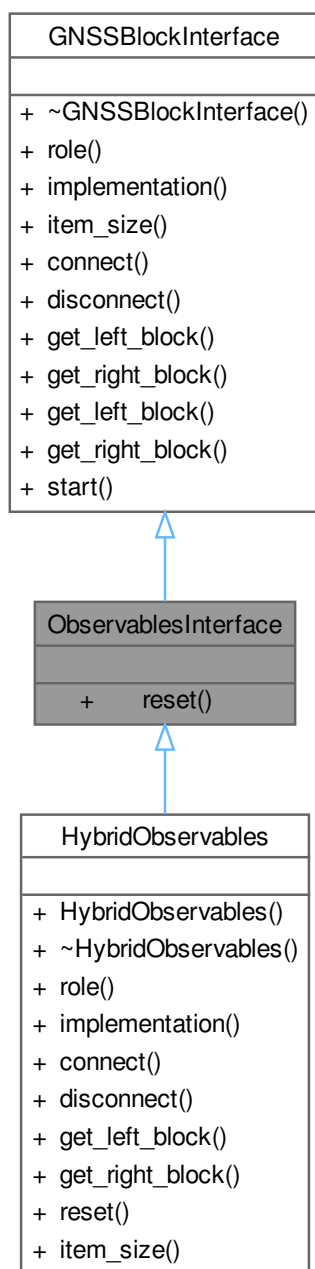
- [rtklib.h](#)

## 10.298 ObservablesInterface Class Reference

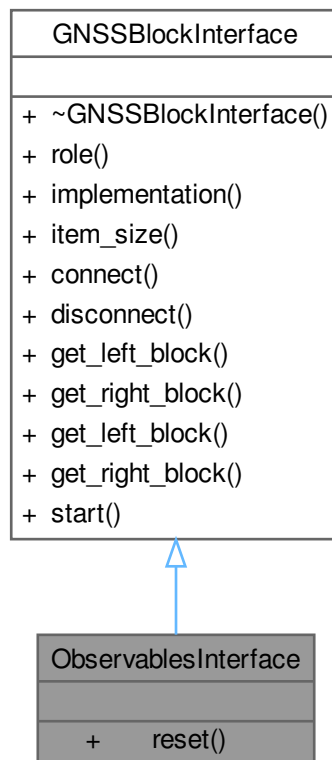
This abstract class represents an interface to an observables block.

```
#include <observables_interface.h>
```

Inheritance diagram for ObservablesInterface:



Collaboration diagram for ObservablesInterface:



### Public Member Functions

- virtual void **reset** ()=0

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual std::string **role** ()=0
- virtual std::string **implementation** ()=0
- virtual size\_t **item\_size** ()=0
- virtual void **connect** (gr::top\_block\_sptr top\_block)=0
- virtual void **disconnect** (gr::top\_block\_sptr top\_block)=0
- virtual gr::basic\_block\_sptr **get\_left\_block** ()=0
- virtual gr::basic\_block\_sptr **get\_right\_block** ()=0
- virtual gr::basic\_block\_sptr **get\_left\_block** (int RF\_channel)
- virtual gr::basic\_block\_sptr **get\_right\_block** (int RF\_channel)
- virtual void **start** ()

*Start the flow of samples if needed.*

#### 10.298.1 Detailed Description

This abstract class represents an interface to an observables block.

Abstract class for pseudorange\_interfaces, derived from [GNSSBlockInterface](#). Since all its methods are virtual, this class cannot be instantiated directly, and a subclass can only be instantiated directly if all inherited pure virtual methods have been implemented by that class or a parent class.

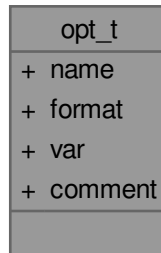
Definition at line 43 of file [observables\\_interface.h](#).

The documentation for this class was generated from the following file:

- [observables\\_interface.h](#)

## 10.299 opt\_t Struct Reference

Collaboration diagram for opt\_t:



### Public Attributes

- const char \* [name](#)
- int [format](#)
- void \* [var](#)
- const char \* [comment](#)

### 10.299.1 Detailed Description

Definition at line 916 of file [rtklib.h](#).

### 10.299.2 Member Data Documentation

#### 10.299.2.1 comment

```
const char* opt_t::comment
```

Definition at line 921 of file [rtklib.h](#).

#### 10.299.2.2 format

```
int opt_t::format
```

Definition at line 919 of file [rtklib.h](#).

#### 10.299.2.3 name

```
const char* opt_t::name
```

Definition at line 918 of file [rtklib.h](#).

#### 10.299.2.4 var

```
void* opt_t::var
```

Definition at line 920 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

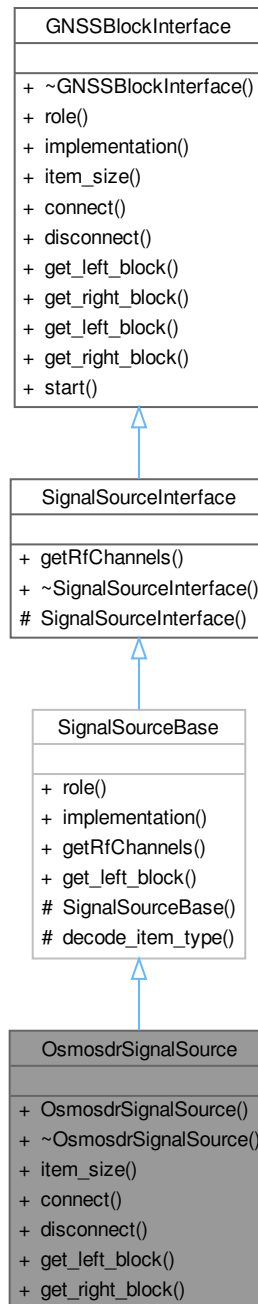
- [rtklib.h](#)

## 10.300 OsmosdrSignalSource Class Reference

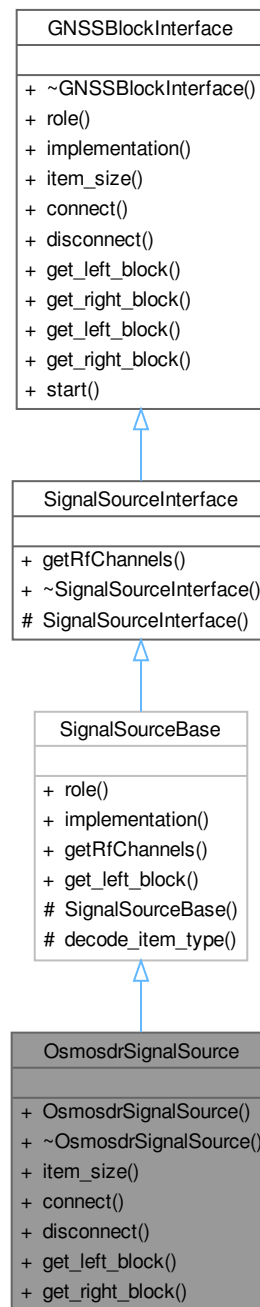
This class reads samples OmoSDR-compatible front-ends, such as HackRF or Realtek's RTL2832U-based USB dongle DVB-T receivers (see <https://osmocom.org/projects/rtl-sdr/wiki>)

```
#include <osmosdr_signal_source.h>
```

Inheritance diagram for OsmosdrSignalSource:



Collaboration diagram for OsmosdrSignalSource:



## Public Member Functions

- **OsmosdrSignalSource** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_stream, unsigned int out\_stream, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override

## Public Member Functions inherited from [SignalSourceBase](#)

- `std::string` [role](#) () final
- `std::string` [implementation](#) () final
- `size_t` [getRfChannels](#) () const override
- `gr::basic_block_sptr` [get\\_left\\_block](#) () override

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual `gr::basic_block_sptr` [get\\_left\\_block](#) (int RF\_channel)
- virtual `gr::basic_block_sptr` [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

## Additional Inherited Members

## Protected Member Functions inherited from [SignalSourceBase](#)

- [SignalSourceBase](#) ([ConfigurationInterface](#) const \*configuration, `std::string` role, `std::string` impl)  
*Constructor.*
- `size_t` [decode\\_item\\_type](#) (`std::string` const &item\_type, `bool` \*is\_interleaved=nullptr, `bool` throw\_on\_↔ error=false)  
*utility for decoding passed ".item\_type" values*

### 10.300.1 Detailed Description

This class reads samples OsmoSDR-compatible front-ends, such as HackRF or Realtek's RTL2832U-based USB dongle DVB-T receivers (see <https://osmocom.org/projects/rtl-sdr/wiki>)  
Definition at line 45 of file [osmosdr\\_signal\\_source.h](#).

### 10.300.2 Member Function Documentation

#### 10.300.2.1 connect()

```
void OsmosdrSignalSource::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.300.2.2 disconnect()

```
void OsmosdrSignalSource::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.300.2.3 get\_left\_block()

```
gr::basic_block_sptr OsmosdrSignalSource::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.300.2.4 get\_right\_block()

```
gr::basic_block_sptr OsmosdrSignalSource::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).





### 10.301.1 Detailed Description

This class handles OSNMA data See [https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo\\_OSNMA\\_User\\_ICD\\_for\\_Test\\_Phase\\_v1.0.pdf](https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OSNMA_User_ICD_for_Test_Phase_v1.0.pdf).

Definition at line 178 of file [osnma\\_data.h](#).

### 10.301.2 Member Data Documentation

#### 10.301.2.1 d\_dsm\_header

[DSM\\_dsm\\_header](#) OSNMA\_data::d\_dsm\_header

Definition at line 183 of file [osnma\\_data.h](#).

#### 10.301.2.2 d\_dsm\_kroot\_message

[DSM\\_KROOT\\_message](#) OSNMA\_data::d\_dsm\_kroot\_message

Definition at line 185 of file [osnma\\_data.h](#).

#### 10.301.2.3 d\_dsm\_kroot\_new\_message

[DSM\\_KROOT\\_message](#) OSNMA\_data::d\_dsm\_kroot\_new\_message

Definition at line 186 of file [osnma\\_data.h](#).

#### 10.301.2.4 d\_dsm\_pkr\_message

[DSM\\_PKR\\_message](#) OSNMA\_data::d\_dsm\_pkr\_message

Definition at line 184 of file [osnma\\_data.h](#).

#### 10.301.2.5 d\_mack\_message

[MACK\\_message](#) OSNMA\_data::d\_mack\_message

Definition at line 187 of file [osnma\\_data.h](#).

#### 10.301.2.6 d\_nav\_data

[OSNMA\\_NavData](#) OSNMA\_data::d\_nav\_data

Definition at line 188 of file [osnma\\_data.h](#).

#### 10.301.2.7 d\_nma\_header

[DSM\\_nma\\_header](#) OSNMA\_data::d\_nma\_header

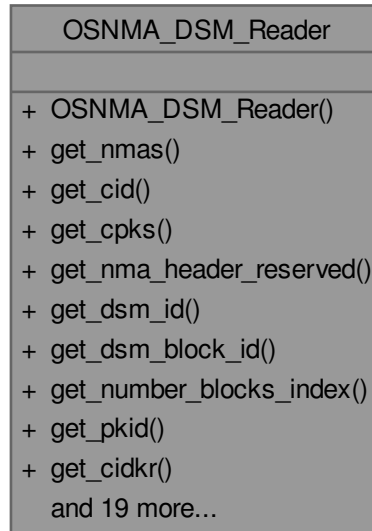
Definition at line 182 of file [osnma\\_data.h](#).

The documentation for this class was generated from the following file:

- [osnma\\_data.h](#)

## 10.302 OSNMA\_DSM\_Reader Class Reference

Collaboration diagram for OSNMA\_DSM\_Reader:



### Public Member Functions

- uint8\_t **get\_nmas** (uint8\_t nma\_header) const
- uint8\_t **get\_cid** (uint8\_t nma\_header) const
- uint8\_t **get\_cpks** (uint8\_t nma\_header) const
- bool **get\_nma\_header\_reserved** (uint8\_t nma\_header) const
- uint8\_t **get\_dsm\_id** (uint8\_t dsm\_header) const
- uint8\_t **get\_dsm\_block\_id** (uint8\_t dsm\_header) const
- uint8\_t **get\_number\_blocks\_index** (uint8\_t dsm\_msg\_0) const
- uint8\_t **get\_pkid** (const std::vector< uint8\_t > &dsm\_msg) const
- uint8\_t **get\_cidkr** (const std::vector< uint8\_t > &dsm\_msg) const
- uint8\_t **get\_dsm\_reserved1** (const std::vector< uint8\_t > &dsm\_msg) const
- uint8\_t **get\_hf** (const std::vector< uint8\_t > &dsm\_msg) const
- uint8\_t **get\_mf** (const std::vector< uint8\_t > &dsm\_msg) const
- uint8\_t **get\_ks** (const std::vector< uint8\_t > &dsm\_msg) const
- uint8\_t **get\_ts** (const std::vector< uint8\_t > &dsm\_msg) const
- uint8\_t **get\_macIt** (const std::vector< uint8\_t > &dsm\_msg) const
- uint8\_t **get\_dsm\_reserved** (const std::vector< uint8\_t > &dsm\_msg) const
- uint16\_t **get\_wn\_k** (const std::vector< uint8\_t > &dsm\_msg) const
- uint8\_t **get\_towh\_k** (const std::vector< uint8\_t > &dsm\_msg) const
- uint64\_t **get\_alpha** (const std::vector< uint8\_t > &dsm\_msg) const
- uint16\_t **get\_l\_dk\_bits** (uint8\_t nb\_dk) const
- uint16\_t **get\_lk\_bits** (uint8\_t ks) const
- std::vector< uint8\_t > **get\_kroot** (const std::vector< uint8\_t > &dsm\_msg, uint16\_t bytes\_lk) const
- std::string **get\_hash\_function** (uint8\_t hf) const
- std::string **get\_nmas\_status** (uint8\_t nmas) const
- std::string **get\_cpks\_status** (uint8\_t cpks) const

- `uint8_t get_mid` (const std::vector< uint8\_t > &dsm\_msg) const
- `uint8_t get_npkt` (const std::vector< uint8\_t > &dsm\_msg) const
- `uint8_t get_npktid` (const std::vector< uint8\_t > &dsm\_msg) const

### 10.302.1 Detailed Description

Definition at line 29 of file [osnma\\_dsm\\_reader.h](#).

The documentation for this class was generated from the following file:

- [osnma\\_dsm\\_reader.h](#)

## 10.303 Osnma\_Helper Class Reference

Collaboration diagram for Osnma\_Helper:

Osnma_Helper
+ GST_START_EPOCH
+ Osnma_Helper()
+ ~Osnma_Helper()
+ compute_gst()
+ compute_gst()
+ compute_gst_now()
+ get_WN()
+ get_TOW()
+ gst_to_uint8()
+ bytes()
+ verification_status_str()
+ convert_to_hex_string()
+ convert_from_hex_string()

### Public Member Functions

- `uint32_t compute_gst` (uint32\_t WN, uint32\_t TOW) const
- `uint32_t compute_gst` (std::tm &input)
- `uint32_t compute_gst_now` ()
- `uint32_t get_WN` (uint32\_t GST) const
- `uint32_t get_TOW` (uint32\_t GST) const
- `std::vector< uint8_t > gst_to_uint8` (uint32\_t GST) const
- `std::vector< uint8_t > bytes` (const std::string &binaryString) const
- `std::string verification_status_str` (int status) const
- `std::string convert_to_hex_string` (const std::vector< uint8\_t > &vector) const
- `std::vector< uint8_t > convert_from_hex_string` (const std::string &hex\_string) const

### Public Attributes

- std::tm [GST\\_START\\_EPOCH](#) {}

### 10.303.1 Detailed Description

Definition at line 31 of file [osnma\\_helper.h](#).

### 10.303.2 Member Data Documentation

#### 10.303.2.1 GST\_START\_EPOCH

```
std::tm Osnma_Helper::GST_START_EPOCH {}
```

Definition at line 46 of file [osnma\\_helper.h](#).

The documentation for this class was generated from the following file:

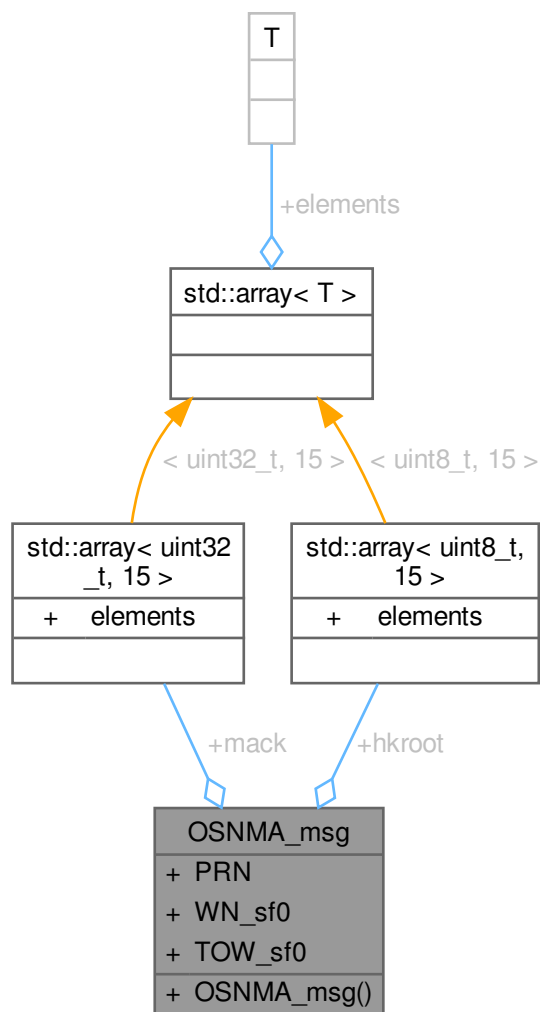
- [osnma\\_helper.h](#)

## 10.304 OSNMA\_msg Class Reference

This class fills the [OSNMA\\_msg](#) structure with the data received from the telemetry blocks.

```
#include <galileo_inav_message.h>
```

Collaboration diagram for OSNMA\_msg:



**Public Attributes**

- `std::array< uint32_t, 15 > mack {}`
- `std::array< uint8_t, 15 > hkroot {}`
- `uint32_t PRN {}`
- `uint32_t WN_sf0 {}`
- `uint32_t TOW_sf0 {}`

**10.304.1 Detailed Description**

This class fills the [OSNMA\\_msg](#) structure with the data received from the telemetry blocks.  
Definition at line 46 of file [galileo\\_inav\\_message.h](#).

**10.304.2 Member Data Documentation****10.304.2.1 hkroot**

```
std::array<uint8_t, 15> OSNMA_msg::hkroot {}
```

Definition at line 51 of file [galileo\\_inav\\_message.h](#).

**10.304.2.2 mack**

```
std::array<uint32_t, 15> OSNMA_msg::mack {}
```

Definition at line 50 of file [galileo\\_inav\\_message.h](#).

**10.304.2.3 PRN**

```
uint32_t OSNMA_msg::PRN {}
```

Definition at line 52 of file [galileo\\_inav\\_message.h](#).

**10.304.2.4 TOW\_sf0**

```
uint32_t OSNMA_msg::TOW_sf0 {}
```

Definition at line 54 of file [galileo\\_inav\\_message.h](#).

**10.304.2.5 WN\_sf0**

```
uint32_t OSNMA_msg::WN_sf0 {}
```

Definition at line 53 of file [galileo\\_inav\\_message.h](#).

The documentation for this class was generated from the following file:

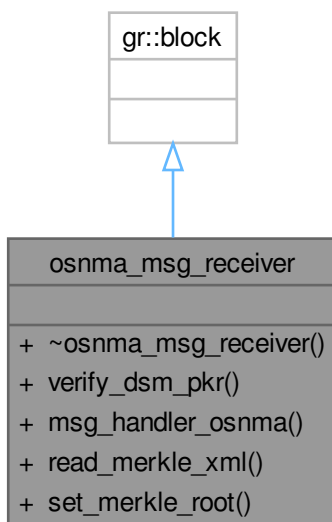
- [galileo\\_inav\\_message.h](#)

**10.305 osnma\_msg\_receiver Class Reference**

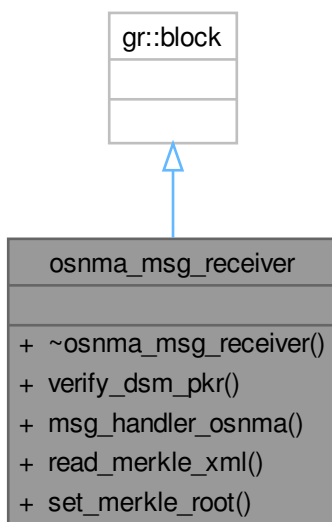
GNU Radio block that receives asynchronous OSNMA messages from the telemetry blocks, stores them in memory, and decodes OSNMA info when enough data have been received. The decoded OSNMA data is sent to the PVT block.

```
#include <osnma_msg_receiver.h>
```

Inheritance diagram for `osnma_msg_receiver`:



Collaboration diagram for `osnma_msg_receiver`:



## Public Member Functions

- `~osnma_msg_receiver()`=default  
*Default destructor.*
- `bool verify_dsm_pkr (const DSM_PKR_message &message) const`

*Public for benchmarking purposes.*

- void [msg\\_handler\\_osnma](#) (const pmt::pmt\_t &msg)

*For testing purposes.*

- void [read\\_merkle\\_xml](#) (const std::string &merklepath)

*Public for testing purposes.*

- void [set\\_merkle\\_root](#) (const std::vector< uint8\_t > &v)

*Public for benchmarking purposes.*

## Friends

- osnma\_msg\_receiver\_sptr [osnma\\_msg\\_receiver\\_make](#) (const std::string &pemFilePath, const std::string &merkleFilePath, bool strict\_mode=false)

## 10.305.1 Detailed Description

GNU Radio block that receives asynchronous OSNMA messages from the telemetry blocks, stores them in memory, and decodes OSNMA info when enough data have been received. The decoded OSNMA data is sent to the PVT block.

Definition at line 61 of file [osnma\\_msg\\_receiver.h](#).

## 10.305.2 Constructor & Destructor Documentation

### 10.305.2.1 ~osnma\_msg\_receiver()

```
osnma_msg_receiver::~osnma_msg_receiver () [default]
```

Default destructor.

## 10.305.3 Member Function Documentation

### 10.305.3.1 msg\_handler\_osnma()

```
void osnma_msg_receiver::msg_handler_osnma (
    const pmt::pmt_t & msg)
```

For testing purposes.

### 10.305.3.2 read\_merkle\_xml()

```
void osnma_msg_receiver::read_merkle_xml (
    const std::string & merklepath)
```

Public for testing purposes.

### 10.305.3.3 set\_merkle\_root()

```
void osnma_msg_receiver::set_merkle_root (
    const std::vector< uint8_t > & v)
```

Public for benchmarking purposes.

### 10.305.3.4 verify\_dsm\_pkr()

```
bool osnma_msg_receiver::verify_dsm_pkr (
    const DSM\_PKR\_message & message) const
```

Public for benchmarking purposes.

The documentation for this class was generated from the following file:

- [osnma\\_msg\\_receiver.h](#)

## 10.306 OSNMA\_NavData Class Reference

Collaboration diagram for OSNMA\_NavData:

OSNMA_NavData
+ nav_data_id
+ OSNMA_NavData()
+ get_utc_data()
+ get_ephemeris_data()
+ get_verified_bits()
+ get_prn_d()
+ get_IOD_nav()
+ get_last_received_TOW()
+ get_tow_sf0()
+ have_this_bits()
+ get_verified_status()
and 10 more...

### Public Member Functions

- std::string **get\_utc\_data** () const
- std::string **get\_ephemeris\_data** () const
- uint32\_t **get\_verified\_bits** () const
- uint32\_t **get\_prn\_d** () const
- uint32\_t **get\_IOD\_nav** () const
- uint32\_t **get\_last\_received\_TOW** () const
- uint32\_t **get\_tow\_sf0** () const
- bool **have\_this\_bits** (std::string nav\_data)
- bool **get\_verified\_status** () const
- bool **add\_nav\_data** (const std::string &nav\_data)
- void **set\_tow\_sf0** (int value)
- void **set\_ephemeris\_data** (std::string value)
- void **set\_utc\_data** (std::string value)
- void **update\_last\_received\_timestamp** (uint32\_t TOW)
- void **set\_prn\_d** (uint32\_t value)
- void **set\_last\_received\_TOW** (uint32\_t TOW)
- void **set\_update\_verified\_bits** (uint32\_t morebits)
- void **set\_verified\_status** (bool value)
- void **set\_IOD\_nav** (uint32\_t value)

### Public Attributes

- const uint32\_t **nav\_data\_id**

### 10.306.1 Detailed Description

Definition at line 136 of file [osnma\\_data.h](#).



## 10.306.2 Constructor & Destructor Documentation

### 10.306.2.1 OSNMA\_NavData()

OSNMA\_NavData::OSNMA\_NavData () [inline]

Definition at line 139 of file [osnma\\_data.h](#).

## 10.306.3 Member Function Documentation

### 10.306.3.1 get\_IOD\_nav()

uint32\_t OSNMA\_NavData::get\_IOD\_nav () const [inline]

Definition at line 145 of file [osnma\\_data.h](#).

### 10.306.3.2 get\_last\_received\_TOW()

uint32\_t OSNMA\_NavData::get\_last\_received\_TOW () const [inline]

Definition at line 146 of file [osnma\\_data.h](#).

### 10.306.3.3 get\_prn\_d()

uint32\_t OSNMA\_NavData::get\_prn\_d () const [inline]

Definition at line 144 of file [osnma\\_data.h](#).

### 10.306.3.4 get\_tow\_sf0()

uint32\_t OSNMA\_NavData::get\_tow\_sf0 () const [inline]

Definition at line 147 of file [osnma\\_data.h](#).

### 10.306.3.5 get\_verified\_bits()

uint32\_t OSNMA\_NavData::get\_verified\_bits () const [inline]

Definition at line 143 of file [osnma\\_data.h](#).

### 10.306.3.6 get\_verified\_status()

bool OSNMA\_NavData::get\_verified\_status () const [inline]

Definition at line 149 of file [osnma\\_data.h](#).

### 10.306.3.7 set\_ephemeris\_data()

void OSNMA\_NavData::set\_ephemeris\_data (  
std::string value) [inline]

Definition at line 152 of file [osnma\\_data.h](#).

### 10.306.3.8 set\_IOD\_nav()

void OSNMA\_NavData::set\_IOD\_nav (  
uint32\_t value) [inline]

Definition at line 159 of file [osnma\\_data.h](#).

### 10.306.3.9 set\_last\_received\_TOW()

void OSNMA\_NavData::set\_last\_received\_TOW (  
uint32\_t TOW) [inline]

Definition at line 156 of file [osnma\\_data.h](#).

#### 10.306.3.10 set\_prn\_d()

```
void OSNMA_NavData::set_prn_d (
    uint32_t value) [inline]
```

Definition at line 155 of file [osnma\\_data.h](#).

#### 10.306.3.11 set\_tow\_sf0()

```
void OSNMA_NavData::set_tow_sf0 (
    int value) [inline]
```

Definition at line 151 of file [osnma\\_data.h](#).

#### 10.306.3.12 set\_update\_verified\_bits()

```
void OSNMA_NavData::set_update_verified_bits (
    uint32_t morebits) [inline]
```

Definition at line 157 of file [osnma\\_data.h](#).

#### 10.306.3.13 set\_utc\_data()

```
void OSNMA_NavData::set_utc_data (
    std::string value) [inline]
```

Definition at line 153 of file [osnma\\_data.h](#).

#### 10.306.3.14 set\_verified\_status()

```
void OSNMA_NavData::set_verified_status (
    bool value) [inline]
```

Definition at line 158 of file [osnma\\_data.h](#).

### 10.306.4 Member Data Documentation

#### 10.306.4.1 nav\_data\_id

```
const uint32_t OSNMA_NavData::nav_data_id
```

Definition at line 140 of file [osnma\\_data.h](#).

The documentation for this class was generated from the following file:

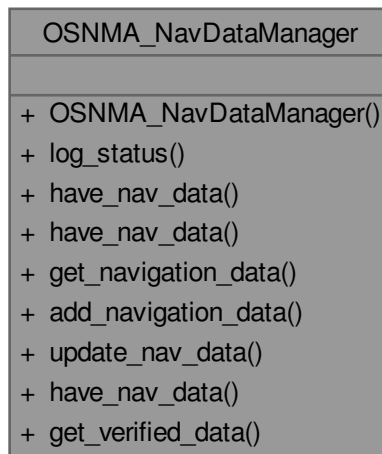
- [osnma\\_data.h](#)

## 10.307 OSNMA\_NavDataManager Class Reference

Class for managing OSNMA navigation data.

```
#include <osnma_nav_data_manager.h>
```

Collaboration diagram for OSNMA\_NavDataManager:



### Public Member Functions

- void **log\_status** () const
- bool **have\_nav\_data** (const [Tag](#) &t) const
- bool **have\_nav\_data** (uint32\_t PRNd, uint32\_t TOW, uint8\_t ADKD) const
- std::string **get\_navigation\_data** (const [Tag](#) &t) const
- void **add\_navigation\_data** (const std::string &nav\_bits, uint32\_t PRNd, uint32\_t TOW)
- void **update\_nav\_data** (const std::multimap< uint32\_t, [Tag](#) > &tags\_verified, uint8\_t tag\_size)
- bool **have\_nav\_data** (const std::string &nav\_bits, uint32\_t PRNd, uint32\_t TOW)
- std::vector< [OSNMA\\_NavData](#) > **get\_verified\_data** ()

### 10.307.1 Detailed Description

Class for managing OSNMA navigation data.

Definition at line 35 of file [osnma\\_nav\\_data\\_manager.h](#).

The documentation for this class was generated from the following file:

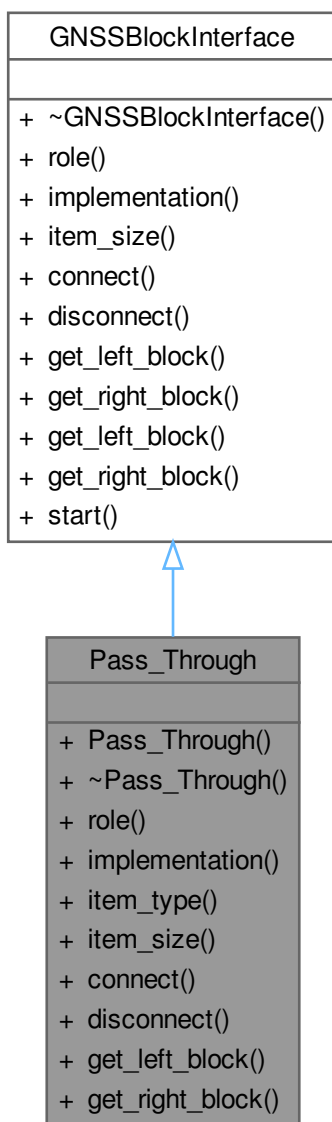
- [osnma\\_nav\\_data\\_manager.h](#)

## 10.308 Pass\_Through Class Reference

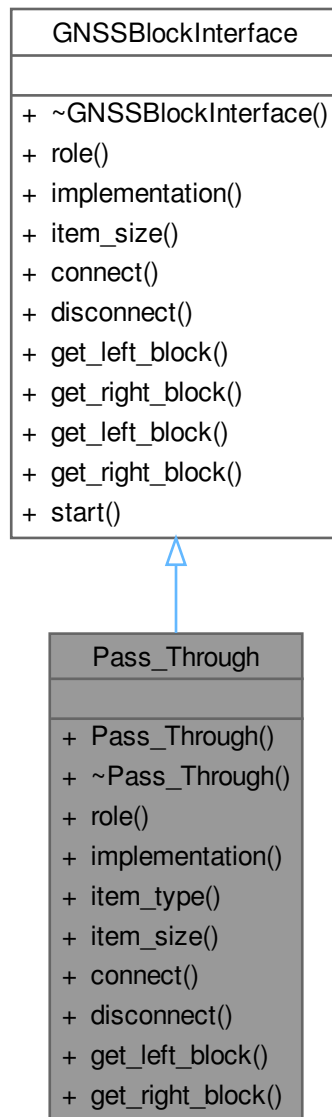
This class implements a block that connects input and output (does nothing)

```
#include <pass_through.h>
```

Inheritance diagram for Pass\_Through:



Collaboration diagram for Pass\_Through:



### Public Member Functions

- **Pass\_Through** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_stream, unsigned int out\_stream)
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "Pass\_Through".
- std::string [item\\_type](#) () const
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual `gr::basic_block_sptr` [get\\_left\\_block](#) (int RF\_channel)
- virtual `gr::basic_block_sptr` [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.308.1 Detailed Description

This class implements a block that connects input and output (does nothing)  
Definition at line 42 of file [pass\\_through.h](#).

### 10.308.2 Member Function Documentation

#### 10.308.2.1 connect()

```
void Pass_Through::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.308.2.2 disconnect()

```
void Pass_Through::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.308.2.3 get\_left\_block()

```
gr::basic_block_sptr Pass_Through::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.308.2.4 get\_right\_block()

```
gr::basic_block_sptr Pass_Through::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.308.2.5 implementation()

```
std::string Pass_Through::implementation () [inline], [override], [virtual]
```

Returns "Pass\_Through".  
Implements [GNSSBlockInterface](#).  
Definition at line 58 of file [pass\\_through.h](#).

#### 10.308.2.6 item\_size()

```
size_t Pass_Through::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).  
Definition at line 68 of file [pass\\_through.h](#).

#### 10.308.2.7 item\_type()

```
std::string Pass_Through::item_type () const [inline]
```

Definition at line 63 of file [pass\\_through.h](#).

#### 10.308.2.8 role()

```
std::string Pass_Through::role () [inline], [override], [virtual]
```

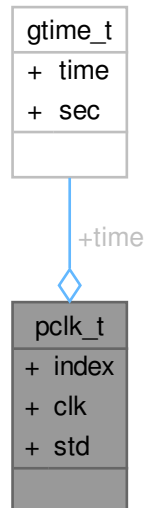
Implements [GNSSBlockInterface](#).  
Definition at line 52 of file [pass\\_through.h](#).

The documentation for this class was generated from the following file:

- [pass\\_through.h](#)

## 10.309 pclk\_t Struct Reference

Collaboration diagram for pclk\_t:



### Public Attributes

- [gtime\\_t](#) `time`
- `int` `index`
- `double` `clk` [`MAXSAT`][1]
- `float` `std` [`MAXSAT`][1]

### 10.309.1 Detailed Description

Definition at line 491 of file [rtklib.h](#).

### 10.309.2 Member Data Documentation

#### 10.309.2.1 clk

```
double pclk_t::clk[MAXSAT][1]
```

Definition at line 495 of file [rtklib.h](#).

#### 10.309.2.2 index

```
int pclk_t::index
```

Definition at line 494 of file [rtklib.h](#).

#### 10.309.2.3 std

```
float pclk_t::std[MAXSAT][1]
```

Definition at line 496 of file [rtklib.h](#).

#### 10.309.2.4 time

`ptime_t pclk_t::time`

Definition at line 493 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

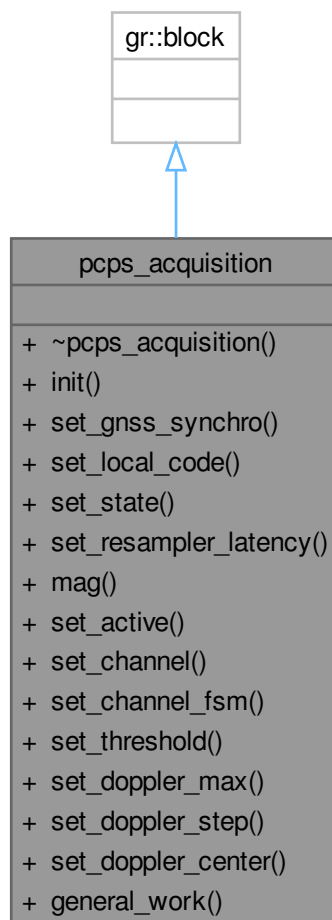
- [rtklib.h](#)

### 10.310 pcps\_acquisition Class Reference

This class implements a Parallel Code Phase Search Acquisition.

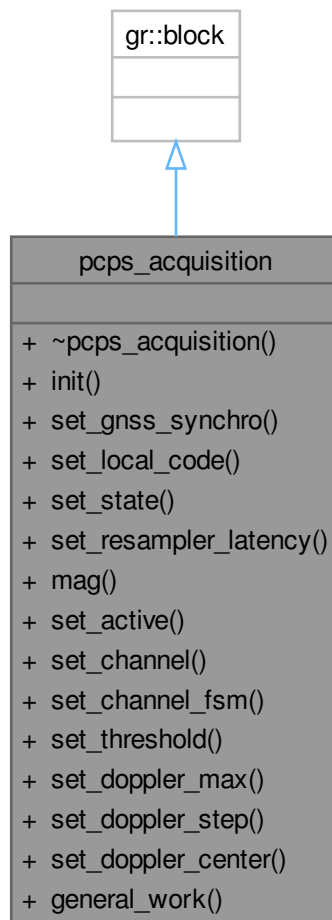
`#include <pcps_acquisition.h>`

Inheritance diagram for `pcps_acquisition`:





Collaboration diagram for pcps\_acquisition:



## Public Member Functions

- void `init` ()  
*Initializes acquisition algorithm and reserves memory.*
- void `set_gnss_synchro` (`Gnss_Synchro` \*p\_gnss\_synchro)  
*Set acquisition/tracking common `Gnss_Synchro` object pointer to exchange synchronization data between acquisition and tracking blocks.*
- void `set_local_code` (`std::complex< float >` \*code)  
*Sets local code for PCPS acquisition algorithm.*
- void `set_state` (`int32_t` state)  
*If set to 1, ensures that acquisition starts at the first available sample.*
- void `set_resampler_latency` (`uint32_t` latency\_samples)
- `uint32_t` `mag` () const  
*Returns the maximum peak of grid search.*
- void `set_active` (bool active)  
*Starts acquisition algorithm, turning from standby mode to active mode.*
- void `set_channel` (`uint32_t` channel)

- Set acquisition channel unique ID.*
- void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm)  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold)  
*Set statistics threshold of PCPS algorithm.*
- void [set\\_doppler\\_max](#) (uint32\_t doppler\_max)  
*Set maximum Doppler grid search.*
- void [set\\_doppler\\_step](#) (uint32\_t doppler\_step)  
*Set Doppler steps for the grid search.*
- void [set\\_doppler\\_center](#) (int32\_t doppler\_center)  
*Set Doppler center frequency for the grid search. It will refresh the Doppler grid.*
- int [general\\_work](#) (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items) override  
*Parallel Code Phase Search Acquisition signal processing.*

## Friends

- pcps\_acquisition\_sptr [pcps\\_make\\_acquisition](#) (const [Acq\\_Conf](#) &conf\_)

## 10.310.1 Detailed Description

This class implements a Parallel Code Phase Search Acquisition.

Check [An Open Source Galileo E1 Software Receiver](#), Algorithm 1, for a pseudocode description of this implementation.

Definition at line 92 of file [pcps\\_acquisition.h](#).

## 10.310.2 Member Function Documentation

### 10.310.2.1 [general\\_work\(\)](#)

```
int pcps_acquisition::general_work (
    int noutput_items,
    gr_vector_int & ninput_items,
    gr_vector_const_void_star & input_items,
    gr_vector_void_star & output_items) [override]
```

Parallel Code Phase Search Acquisition signal processing.

### 10.310.2.2 [init\(\)](#)

```
void pcps_acquisition::init ()
```

Initializes acquisition algorithm and reserves memory.

### 10.310.2.3 [mag\(\)](#)

```
uint32_t pcps_acquisition::mag () const [inline]
```

Returns the maximum peak of grid search.

Definition at line 131 of file [pcps\\_acquisition.h](#).

### 10.310.2.4 [set\\_active\(\)](#)

```
void pcps_acquisition::set_active (
    bool active) [inline]
```

Starts acquisition algorithm, turning from standby mode to active mode.

## Parameters

<i>active</i>	- bool that activates/deactivates the block.
---------------	--

Definition at line 141 of file [pcps\\_acquisition.h](#).

**10.310.2.5 set\_channel()**

```
void pcps_acquisition::set_channel (
    uint32_t channel) [inline]
```

Set acquisition channel unique ID.

**Parameters**

<i>channel</i>	- receiver channel.
----------------	---------------------

Definition at line 151 of file [pcps\\_acquisition.h](#).

**10.310.2.6 set\_channel\_fsm()**

```
void pcps_acquisition::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline]
```

Set channel fsm associated to this acquisition instance.

Definition at line 159 of file [pcps\\_acquisition.h](#).

**10.310.2.7 set\_doppler\_center()**

```
void pcps_acquisition::set_doppler_center (
    int32_t doppler_center)
```

Set Doppler center frequency for the grid search. It will refresh the Doppler grid.

**Parameters**

<i>doppler_center</i>	- Frequency center of the search grid [Hz].
-----------------------	---

**10.310.2.8 set\_doppler\_max()**

```
void pcps_acquisition::set_doppler_max (
    uint32_t doppler_max) [inline]
```

Set maximum Doppler grid search.

**Parameters**

<i>doppler_max</i>	- Maximum Doppler shift considered in the grid search [Hz].
--------------------	---

Definition at line 179 of file [pcps\\_acquisition.h](#).

**10.310.2.9 set\_doppler\_step()**

```
void pcps_acquisition::set_doppler_step (
    uint32_t doppler_step) [inline]
```

Set Doppler steps for the grid search.

**Parameters**

<i>doppler_step</i>	- Frequency bin of the search grid [Hz].
---------------------	--

Definition at line 189 of file [pcps\\_acquisition.h](#).

**10.310.2.10 set\_gnss\_synchro()**

```
void pcps_acquisition::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [inline]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to exchange synchronization data between acquisition and tracking blocks.

## Parameters

<i>p_gnss_synchro</i>	Satellite information shared by the processing blocks.
-----------------------	--

Definition at line 107 of file [pcps\\_acquisition.h](#).

**10.310.2.11 set\_local\_code()**

```
void pcps_acquisition::set_local_code (
    std::complex< float > * code)
```

Sets local code for PCPS acquisition algorithm.

## Parameters

<i>code</i>	- Pointer to the PRN code.
-------------	----------------------------

**10.310.2.12 set\_state()**

```
void pcps_acquisition::set_state (
    int32_t state)
```

If set to 1, ensures that acquisition starts at the first available sample.

## Parameters

<i>state</i>	- int=1 forces start of acquisition
--------------	-------------------------------------

**10.310.2.13 set\_threshold()**

```
void pcps_acquisition::set_threshold (
    float threshold) [inline]
```

Set statistics threshold of PCPS algorithm.

## Parameters

<i>threshold</i>	- Threshold for signal detection (check <a href="#">Navitec2012</a> , Algorithm 1, for a definition of this threshold).
------------------	---

Definition at line 169 of file [pcps\\_acquisition.h](#).

The documentation for this class was generated from the following file:

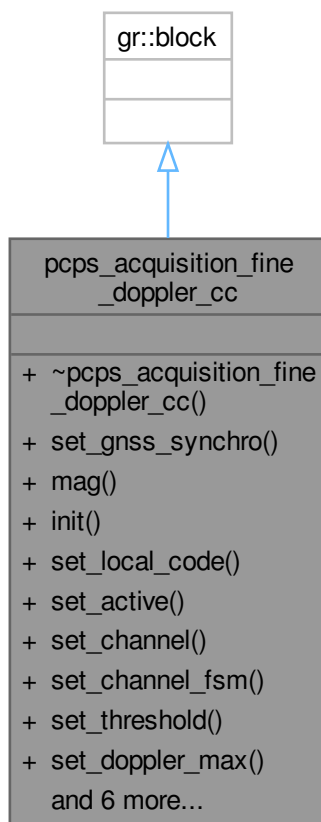
- [pcps\\_acquisition.h](#)

**10.311 pcps\_acquisition\_fine\_doppler\_cc Class Reference**

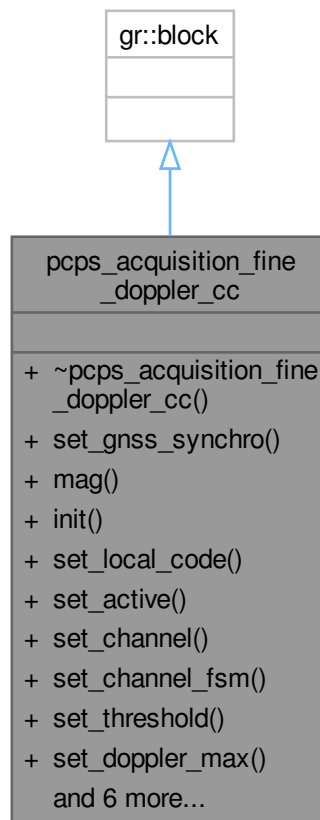
This class implements a Parallel Code Phase Search Acquisition.

```
#include <pcps_acquisition_fine_doppler_cc.h>
```

Inheritance diagram for pcps\_acquisition\_fine\_doppler\_cc:



Collaboration diagram for pcps\_acquisition\_fine\_doppler\_cc:



## Public Member Functions

- `~pcps_acquisition_fine_doppler_cc ()`=default  
*Default destructor.*
- void `set_gnss_synchro` (`Gnss_Synchro *p_gnss_synchro`)  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to exchange synchronization data between acquisition and tracking blocks.*
- unsigned int `mag ()` const  
*Returns the maximum peak of grid search.*
- void `init ()`  
*Initializes acquisition algorithm.*
- void `set_local_code` (`std::complex< float > *code`)  
*Sets local code for PCPS acquisition algorithm.*
- void `set_active` (bool active)  
*Starts acquisition algorithm, turning from standby mode to active mode.*
- void `set_channel` (unsigned int channel)  
*Set acquisition channel unique ID.*
- void `set_channel_fsm` (`std::weak_ptr< ChannelFsm > channel_fsm`)  
*Set channel fsm associated to this acquisition instance.*
- void `set_threshold` (float threshold)

- Set statistics threshold of PCPS algorithm.*
- void [set\\_doppler\\_max](#) (unsigned int doppler\_max)
- Set maximum Doppler grid search.*
- void [set\\_doppler\\_step](#) (unsigned int doppler\_step)
- Set Doppler steps for the grid search.*
- void [set\\_state](#) (int state)
- If set to 1, ensures that acquisition starts at the first available sample.*
- unsigned int [nextPowerOf2](#) (unsigned int n)
- Obtains the next power of 2 greater or equal to the input parameter.*
- void **dump\_results** (int effective\_fft\_size)
- void **forecast** (int noutput\_items, gr\_vector\_int &ninput\_items\_required)
- int [general\\_work](#) (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)
- Parallel Code Phase Search Acquisition signal processing.*

## Friends

- pcps\_acquisition\_fine\_doppler\_cc\_sptr **pcps\_make\_acquisition\_fine\_doppler\_cc** (const [Acq\\_Conf](#) &conf\_)

## 10.311.1 Detailed Description

This class implements a Parallel Code Phase Search Acquisition.

Definition at line 73 of file [pcps\\_acquisition\\_fine\\_doppler\\_cc.h](#).

## 10.311.2 Constructor & Destructor Documentation

### 10.311.2.1 ~pcps\_acquisition\_fine\_doppler\_cc()

```
pcps_acquisition_fine_doppler_cc::~~pcps_acquisition_fine_doppler_cc () [default]
```

Default destructor.

## 10.311.3 Member Function Documentation

### 10.311.3.1 general\_work()

```
int pcps_acquisition_fine_doppler_cc::general_work (
    int noutput_items,
    gr_vector_int & ninput_items,
    gr_vector_const_void_star & input_items,
    gr_vector_void_star & output_items)
```

Parallel Code Phase Search Acquisition signal processing.

### 10.311.3.2 init()

```
void pcps_acquisition_fine_doppler_cc::init ()
```

Initializes acquisition algorithm.

### 10.311.3.3 mag()

```
unsigned int pcps_acquisition_fine_doppler_cc::mag () const [inline]
```

Returns the maximum peak of grid search.

Definition at line 94 of file [pcps\\_acquisition\\_fine\\_doppler\\_cc.h](#).

**10.311.3.4 nextPowerOf2()**

```
unsigned int pcps_acquisition_fine_doppler_cc::nextPowerOf2 (  
    unsigned int n)
```

Obtains the next power of 2 greater or equal to the input parameter.



## Parameters

<i>n</i>	- Integer value to obtain the next power of 2.
----------	--

**10.311.3.5 set\_active()**

```
void pcps_acquisition_fine_doppler_cc::set_active (
    bool active) [inline]
```

Starts acquisition algorithm, turning from standby mode to active mode.

## Parameters

<i>active</i>	- bool that activates/deactivates the block.
---------------	--

Definition at line 115 of file [pcps\\_acquisition\\_fine\\_doppler\\_cc.h](#).

**10.311.3.6 set\_channel()**

```
void pcps_acquisition_fine_doppler_cc::set_channel (
    unsigned int channel) [inline]
```

Set acquisition channel unique ID.

## Parameters

<i>channel</i>	- receiver channel.
----------------	---------------------

Definition at line 124 of file [pcps\\_acquisition\\_fine\\_doppler\\_cc.h](#).

**10.311.3.7 set\_channel\_fsm()**

```
void pcps_acquisition_fine_doppler_cc::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline]
```

Set channel fsm associated to this acquisition instance.

Definition at line 133 of file [pcps\\_acquisition\\_fine\\_doppler\\_cc.h](#).

**10.311.3.8 set\_doppler\_max()**

```
void pcps_acquisition_fine_doppler_cc::set_doppler_max (
    unsigned int doppler_max) [inline]
```

Set maximum Doppler grid search.

## Parameters

<i>doppler_max</i>	- Maximum Doppler shift considered in the grid search [Hz].
--------------------	---

Definition at line 152 of file [pcps\\_acquisition\\_fine\\_doppler\\_cc.h](#).

**10.311.3.9 set\_doppler\_step()**

```
void pcps_acquisition_fine_doppler_cc::set_doppler_step (
    unsigned int doppler_step)
```

Set Doppler steps for the grid search.

## Parameters

<i>doppler_step</i>	- Frequency bin of the search grid [Hz].
---------------------	--

**10.311.3.10 set\_gnss\_synchro()**

```
void pcps_acquisition_fine_doppler_cc::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [inline]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to exchange synchronization data between acquisition and tracking blocks.

**Parameters**

<i>p_gnss_synchro</i>	Satellite information shared by the processing blocks.
-----------------------	--

Definition at line 86 of file [pcps\\_acquisition\\_fine\\_doppler\\_cc.h](#).

**10.311.3.11 set\_local\_code()**

```
void pcps_acquisition_fine_doppler_cc::set_local_code (
    std::complex< float > * code)
```

Sets local code for PCPS acquisition algorithm.

**Parameters**

<i>code</i>	- Pointer to the PRN code.
-------------	----------------------------

**10.311.3.12 set\_state()**

```
void pcps_acquisition_fine_doppler_cc::set_state (
    int state)
```

If set to 1, ensures that acquisition starts at the first available sample.

**Parameters**

<i>state</i>	- int=1 forces start of acquisition
--------------	-------------------------------------

**10.311.3.13 set\_threshold()**

```
void pcps_acquisition_fine_doppler_cc::set_threshold (
    float threshold) [inline]
```

Set statistics threshold of PCPS algorithm.

**Parameters**

<i>threshold</i>	- Threshold for signal detection (check <a href="#">Navitec2012</a> , Algorithm 1, for a definition of this threshold).
------------------	---

Definition at line 143 of file [pcps\\_acquisition\\_fine\\_doppler\\_cc.h](#).

The documentation for this class was generated from the following file:

- [pcps\\_acquisition\\_fine\\_doppler\\_cc.h](#)

**10.312 pcps\_acquisition\_fpga Class Reference**

This class implements a Parallel Code Phase Search Acquisition that uses the FPGA.

```
#include <pcps_acquisition_fpga.h>
```

Collaboration diagram for pcps\_acquisition\_fpga:

pcps_acquisition_fpga
<ul style="list-style-type: none"> <li>+ ~pcps_acquisition_fpga()</li> <li>+ set_gnss_synchro()</li> <li>+ mag()</li> <li>+ init()</li> <li>+ set_local_code()</li> <li>+ set_state()</li> <li>+ set_active()</li> <li>+ set_channel()</li> <li>+ set_channel_fsm()</li> <li>+ set_threshold()</li> <li>+ set_doppler_max()</li> <li>+ set_doppler_step()</li> <li>+ set_doppler_center()</li> <li>+ reset_acquisition()</li> <li>+ stop_acquisition()</li> </ul>

## Public Member Functions

- [~pcps\\_acquisition\\_fpga](#) ()=default  
*Destructor.*
- void [set\\_gnss\\_synchro](#) ([Gnss\\_Synchro](#) \*p\_gnss\_synchro)  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to exchange synchronization data between acquisition and tracking blocks.*
- uint32\_t [mag](#) () const  
*Returns the maximum peak of grid search.*
- void [init](#) ()  
*Initializes acquisition algorithm.*
- void [set\\_local\\_code](#) ()  
*Sets local code for PCPS acquisition algorithm.*
- void [set\\_state](#) (int32\_t state)  
*If set to 1, ensures that acquisition starts at the first available sample.*
- void [set\\_active](#) (bool active)  
*Starts acquisition algorithm, turning from standby mode to active mode.*
- void [set\\_channel](#) (uint32\_t channel)  
*Set acquisition channel unique ID.*
- void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm)  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold)  
*Set statistics threshold of PCPS algorithm.*

- void `set_doppler_max` (uint32\_t doppler\_max)  
*Set maximum Doppler grid search.*
- void `set_doppler_step` (uint32\_t doppler\_step)  
*Set Doppler steps for the grid search.*
- void `set_doppler_center` (int32\_t doppler\_center)  
*Set Doppler center frequency for the grid search. It will refresh the Doppler grid.*
- void `reset_acquisition` ()  
*This function triggers a HW reset of the FPGA PL.*
- void `stop_acquisition` ()  
*stop the acquisition and the other FPGA modules.*

## Friends

- pcps\_acquisition\_fpga\_sptr `pcps_make_acquisition_fpga` (`Acq_Conf_Fpga` \*conf, uint32\_t acq\_buff\_num, std::vector< std::pair< uint32\_t, uint32\_t > > &downsampling\_filter\_specs, uint32\_t &max\_FFT\_size)

## 10.312.1 Detailed Description

This class implements a Parallel Code Phase Search Acquisition that uses the FPGA.

Check [An Open Source Galileo E1 Software Receiver](#), Algorithm 1, for a pseudocode description of this implementation.

Definition at line 58 of file `pcps_acquisition_fpga.h`.

## 10.312.2 Constructor & Destructor Documentation

### 10.312.2.1 ~pcps\_acquisition\_fpga()

```
pcps_acquisition_fpga::~pcps_acquisition_fpga () [default]
```

Destructor.

## 10.312.3 Member Function Documentation

### 10.312.3.1 init()

```
void pcps_acquisition_fpga::init ()
```

Initializes acquisition algorithm.

### 10.312.3.2 mag()

```
uint32_t pcps_acquisition_fpga::mag () const [inline]
```

Returns the maximum peak of grid search.  
Definition at line 79 of file `pcps_acquisition_fpga.h`.

### 10.312.3.3 reset\_acquisition()

```
void pcps_acquisition_fpga::reset_acquisition ()
```

This function triggers a HW reset of the FPGA PL.

### 10.312.3.4 set\_active()

```
void pcps_acquisition_fpga::set_active (
    bool active)
```

Starts acquisition algorithm, turning from standby mode to active mode.

## Parameters

<i>active</i>	- bool that activates/deactivates the block.
---------------	--

**10.312.3.5 set\_channel()**

```
void pcps_acquisition_fpga::set_channel (
    uint32_t channel) [inline]
```

Set acquisition channel unique ID.

**Parameters**

<i>channel</i>	- receiver channel.
----------------	---------------------

Definition at line 112 of file [pcps\\_acquisition\\_fpga.h](#).

**10.312.3.6 set\_channel\_fsm()**

```
void pcps_acquisition_fpga::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline]
```

Set channel fsm associated to this acquisition instance.

Definition at line 120 of file [pcps\\_acquisition\\_fpga.h](#).

**10.312.3.7 set\_doppler\_center()**

```
void pcps_acquisition_fpga::set_doppler_center (
    int32_t doppler_center)
```

Set Doppler center frequency for the grid search. It will refresh the Doppler grid.

**Parameters**

<i>doppler_center</i>	- Frequency center of the search grid [Hz].
-----------------------	---

**10.312.3.8 set\_doppler\_max()**

```
void pcps_acquisition_fpga::set_doppler_max (
    uint32_t doppler_max) [inline]
```

Set maximum Doppler grid search.

**Parameters**

<i>doppler_max</i>	- Maximum Doppler shift considered in the grid search [Hz].
--------------------	---

Definition at line 139 of file [pcps\\_acquisition\\_fpga.h](#).

**10.312.3.9 set\_doppler\_step()**

```
void pcps_acquisition_fpga::set_doppler_step (
    uint32_t doppler_step) [inline]
```

Set Doppler steps for the grid search.

**Parameters**

<i>doppler_step</i>	- Frequency bin of the search grid [Hz].
---------------------	--

Definition at line 149 of file [pcps\\_acquisition\\_fpga.h](#).

**10.312.3.10 set\_gnss\_synchro()**

```
void pcps_acquisition_fpga::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [inline]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to exchange synchronization data between acquisition and tracking blocks.

## Parameters

<i>p_gnss_synchro</i>	Satellite information shared by the processing blocks.
-----------------------	--

Definition at line 71 of file [pcps\\_acquisition\\_fpga.h](#).

**10.312.3.11 set\_local\_code()**

```
void pcps_acquisition_fpga::set_local_code ()
```

Sets local code for PCPS acquisition algorithm.

**10.312.3.12 set\_state()**

```
void pcps_acquisition_fpga::set_state (
    int32_t state)
```

If set to 1, ensures that acquisition starts at the first available sample.

## Parameters

<i>state</i>	- int=1 forces start of acquisition
--------------	-------------------------------------

**10.312.3.13 set\_threshold()**

```
void pcps_acquisition_fpga::set_threshold (
    float threshold) [inline]
```

Set statistics threshold of PCPS algorithm.

## Parameters

<i>threshold</i>	- Threshold for signal detection (check <a href="#">Navitec2012</a> , Algorithm 1, for a definition of this threshold).
------------------	---

Definition at line 130 of file [pcps\\_acquisition\\_fpga.h](#).

**10.312.3.14 stop\_acquisition()**

```
void pcps_acquisition_fpga::stop_acquisition ()
```

stop the acquisition and the other FPGA modules.

The documentation for this class was generated from the following file:

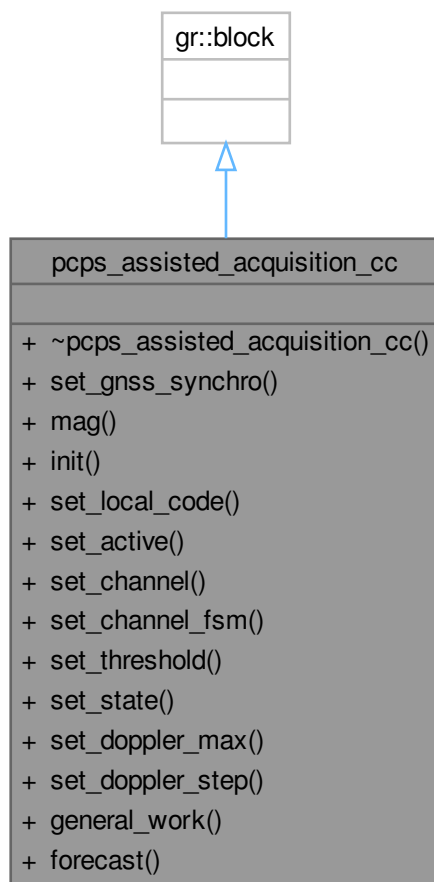
- [pcps\\_acquisition\\_fpga.h](#)

**10.313 pcps\_assisted\_acquisition\_cc Class Reference**

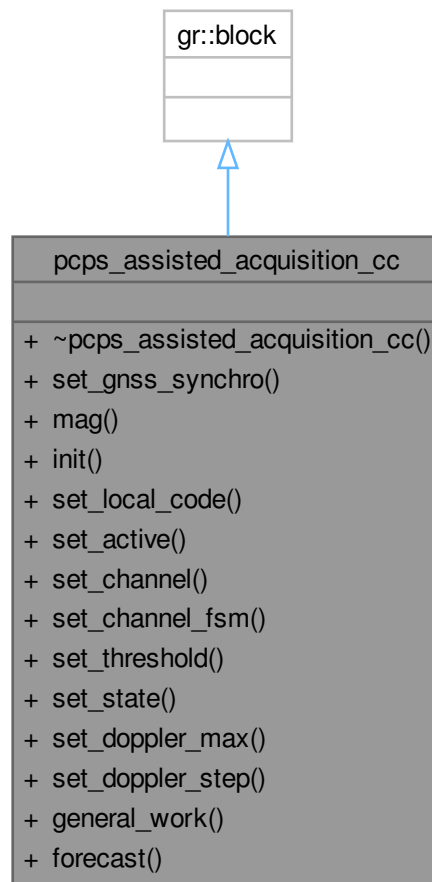
This class implements a Parallel Code Phase Search Acquisition.

```
#include <pcps_assisted_acquisition_cc.h>
```

Inheritance diagram for pcps\_assisted\_acquisition\_cc:



Collaboration diagram for pcps\_assisted\_acquisition\_cc:



## Public Member Functions

- `~pcps_assisted_acquisition_cc ()`  
*Default destructor.*
- void `set_gnss_synchro (Gnss_Synchro *p_gnss_synchro)`  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to exchange synchronization data between acquisition and tracking blocks.*
- `uint32_t mag () const`  
*Returns the maximum peak of grid search.*
- void `init ()`  
*Initializes acquisition algorithm.*
- void `set_local_code (std::complex< float > *code)`  
*Sets local code for PCPS acquisition algorithm.*
- void `set_active (bool active)`  
*Starts acquisition algorithm, turning from standby mode to active mode.*
- void `set_channel (uint32_t channel)`  
*Set acquisition channel unique ID.*
- void `set_channel_fsm (std::weak_ptr< ChannelFsm > channel_fsm)`



*Set channel fsm associated to this acquisition instance.*

- void [set\\_threshold](#) (float threshold)

*Set statistics threshold of PCPS algorithm.*

- void [set\\_state](#) (int32\_t state)
- void [set\\_doppler\\_max](#) (uint32\_t doppler\_max)

*Set maximum Doppler grid search.*

- void [set\\_doppler\\_step](#) (uint32\_t doppler\_step)

*Set Doppler steps for the grid search.*

- int [general\\_work](#) (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)

*Parallel Code Phase Search Acquisition signal processing.*

- void **forecast** (int noutput\_items, gr\_vector\_int &ninput\_items\_required)

## Friends

- pcps\_assisted\_acquisition\_cc\_sptr **pcps\_make\_assisted\_acquisition\_cc** (int32\_t max\_dwells, uint32\_t sampled\_ms, int32\_t doppler\_max, int32\_t doppler\_min, int64\_t fs\_in, int32\_t samples\_per\_ms, bool dump, const std::string &dump\_filename, bool enable\_monitor\_output)

## 10.313.1 Detailed Description

This class implements a Parallel Code Phase Search Acquisition.

Check [An Open Source Galileo E1 Software Receiver](#), Algorithm 1, for a pseudocode description of this implementation.

Definition at line 75 of file [pcps\\_assisted\\_acquisition\\_cc.h](#).

## 10.313.2 Constructor & Destructor Documentation

### 10.313.2.1 ~pcps\_assisted\_acquisition\_cc()

```
pcps_assisted_acquisition_cc::~pcps_assisted_acquisition_cc ()
```

Default destructor.

## 10.313.3 Member Function Documentation

### 10.313.3.1 general\_work()

```
int pcps_assisted_acquisition_cc::general_work (
    int noutput_items,
    gr_vector_int & ninput_items,
    gr_vector_const_void_star & input_items,
    gr_vector_void_star & output_items)
```

Parallel Code Phase Search Acquisition signal processing.

### 10.313.3.2 init()

```
void pcps_assisted_acquisition_cc::init ()
```

Initializes acquisition algorithm.

### 10.313.3.3 mag()

```
uint32_t pcps_assisted_acquisition_cc::mag () const [inline]
```

Returns the maximum peak of grid search.

Definition at line 96 of file [pcps\\_assisted\\_acquisition\\_cc.h](#).

**10.313.3.4 set\_active()**

```
void pcps_assisted_acquisition_cc::set_active (
    bool active) [inline]
```

Starts acquisition algorithm, turning from standby mode to active mode.

**Parameters**

<i>active</i>	- bool that activates/deactivates the block.
---------------	--

Definition at line 117 of file [pcps\\_assisted\\_acquisition\\_cc.h](#).

**10.313.3.5 set\_channel()**

```
void pcps_assisted_acquisition_cc::set_channel (
    uint32_t channel) [inline]
```

Set acquisition channel unique ID.

**Parameters**

<i>channel</i>	- receiver channel.
----------------	---------------------

Definition at line 126 of file [pcps\\_assisted\\_acquisition\\_cc.h](#).

**10.313.3.6 set\_channel\_fsm()**

```
void pcps_assisted_acquisition_cc::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline]
```

Set channel fsm associated to this acquisition instance.

Definition at line 134 of file [pcps\\_assisted\\_acquisition\\_cc.h](#).

**10.313.3.7 set\_doppler\_max()**

```
void pcps_assisted_acquisition_cc::set_doppler_max (
    uint32_t doppler_max) [inline]
```

Set maximum Doppler grid search.

**Parameters**

<i>doppler_max</i>	- Maximum Doppler shift considered in the grid search [Hz].
--------------------	---

Definition at line 158 of file [pcps\\_assisted\\_acquisition\\_cc.h](#).

**10.313.3.8 set\_doppler\_step()**

```
void pcps_assisted_acquisition_cc::set_doppler_step (
    uint32_t doppler_step)
```

Set Doppler steps for the grid search.

**Parameters**

<i>doppler_step</i>	- Frequency bin of the search grid [Hz].
---------------------	--

**10.313.3.9 set\_gnss\_synchro()**

```
void pcps_assisted_acquisition_cc::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [inline]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to exchange synchronization data between acquisition and tracking blocks.

## Parameters

<i>p_gnss_synchro</i>	Satellite information shared by the processing blocks.
-----------------------	--

Definition at line 88 of file [pcps\\_assisted\\_acquisition\\_cc.h](#).

**10.313.3.10 set\_local\_code()**

```
void pcps_assisted_acquisition_cc::set_local_code (
    std::complex< float > * code)
```

Sets local code for PCPS acquisition algorithm.

## Parameters

<i>code</i>	- Pointer to the PRN code.
-------------	----------------------------

**10.313.3.11 set\_state()**

```
void pcps_assisted_acquisition_cc::set_state (
    int32_t state) [inline]
```

Definition at line 149 of file [pcps\\_assisted\\_acquisition\\_cc.h](#).

**10.313.3.12 set\_threshold()**

```
void pcps_assisted_acquisition_cc::set_threshold (
    float threshold) [inline]
```

Set statistics threshold of PCPS algorithm.

## Parameters

<i>threshold</i>	- Threshold for signal detection (check <a href="#">Navitec2012</a> , Algorithm 1, for a definition of this threshold).
------------------	---

Definition at line 144 of file [pcps\\_assisted\\_acquisition\\_cc.h](#).

The documentation for this class was generated from the following file:

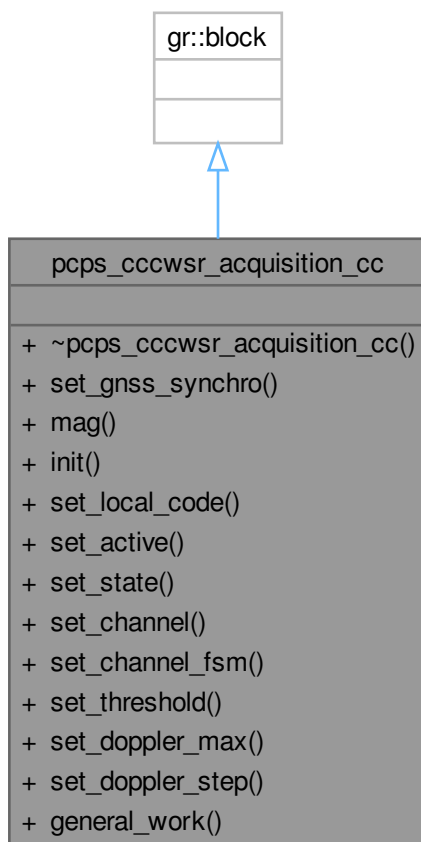
- [pcps\\_assisted\\_acquisition\\_cc.h](#)

**10.314 pcps\_cccwsr\_acquisition\_cc Class Reference**

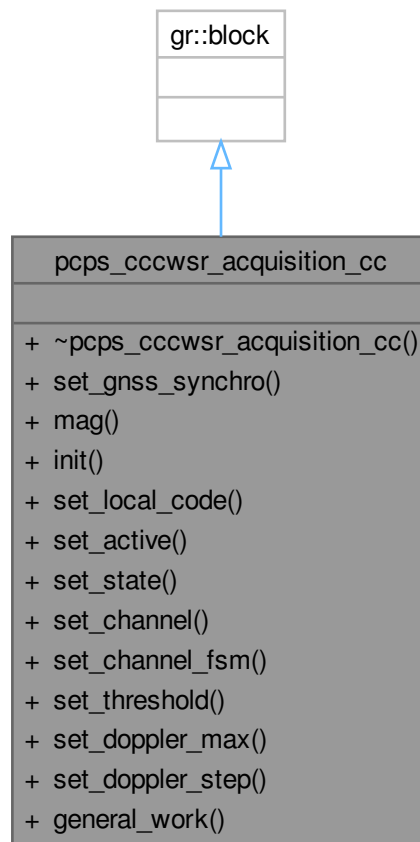
This class implements a Parallel Code Phase Search Acquisition with Coherent [Channel](#) Combining With Sign Recovery scheme.

```
#include <pcps_cccwsr_acquisition_cc.h>
```

Inheritance diagram for pcps\_cccwsr\_acquisition\_cc:



Collaboration diagram for pcps\_cccwsr\_acquisition\_cc:



## Public Member Functions

- `~pcps_cccwsr_acquisition_cc ()`  
*Default destructor.*
- `void set_gnss_synchro (Gnss_Synchro *p_gnss_synchro)`  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to exchange synchronization data between acquisition and tracking blocks.*
- `uint32_t mag () const`  
*Returns the maximum peak of grid search.*
- `void init ()`  
*Initializes acquisition algorithm.*
- `void set_local_code (std::complex< float > *code_data, std::complex< float > *code_pilot)`  
*Sets local code for CCCWSR acquisition algorithm.*
- `void set_active (bool active)`  
*Starts acquisition algorithm, turning from standby mode to active mode.*
- `void set_state (int32_t state)`  
*If set to 1, ensures that acquisition starts at the first available sample.*
- `void set_channel (uint32_t channel)`  
*Set acquisition channel unique ID.*

- void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm)  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold)  
*Set statistics threshold of CCCWSR algorithm.*
- void [set\\_doppler\\_max](#) (uint32\_t doppler\_max)  
*Set maximum Doppler grid search.*
- void [set\\_doppler\\_step](#) (uint32\_t doppler\_step)  
*Set Doppler steps for the grid search.*
- int [general\\_work](#) (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)  
*Coherent [Channel](#) Combining With Sign Recovery Acquisition signal processing.*

## Friends

- pcps\_cccwsr\_acquisition\_cc\_sptr [pcps\\_cccwsr\\_make\\_acquisition\\_cc](#) (uint32\_t sampled\_ms, uint32\_t max\_dwells, uint32\_t doppler\_max, int64\_t fs\_in, int32\_t samples\_per\_ms, int32\_t samples\_per\_code, bool dump, const std::string &dump\_filename, bool enable\_monitor\_output)

## 10.314.1 Detailed Description

This class implements a Parallel Code Phase Search Acquisition with Coherent [Channel](#) Combining With Sign Recovery scheme.

Definition at line 62 of file [pcps\\_cccwsr\\_acquisition\\_cc.h](#).

## 10.314.2 Constructor & Destructor Documentation

### 10.314.2.1 ~pcps\_cccwsr\_acquisition\_cc()

```
pcps_cccwsr_acquisition_cc::~pcps_cccwsr_acquisition_cc ()
```

Default destructor.

## 10.314.3 Member Function Documentation

### 10.314.3.1 general\_work()

```
int pcps_cccwsr_acquisition_cc::general_work (
    int noutput_items,
    gr_vector_int & ninput_items,
    gr_vector_const_void_star & input_items,
    gr_vector_void_star & output_items)
```

Coherent [Channel](#) Combining With Sign Recovery Acquisition signal processing.

### 10.314.3.2 init()

```
void pcps_cccwsr_acquisition_cc::init ()
```

Initializes acquisition algorithm.

### 10.314.3.3 mag()

```
uint32_t pcps_cccwsr_acquisition_cc::mag () const [inline]
```

Returns the maximum peak of grid search.  
Definition at line 83 of file [pcps\\_cccwsr\\_acquisition\\_cc.h](#).

### 10.314.3.4 set\_active()

```
void pcps_cccwsr_acquisition_cc::set_active (
    bool active) [inline]
```

Starts acquisition algorithm, turning from standby mode to active mode.

## Parameters

<i>active</i>	- bool that activates/deactivates the block.
---------------	--

Definition at line 105 of file [pcps\\_cccwsr\\_acquisition\\_cc.h](#).

**10.314.3.5 set\_channel()**

```
void pcps_cccwsr_acquisition_cc::set_channel (
    uint32_t channel) [inline]
```

Set acquisition channel unique ID.

## Parameters

<i>channel</i>	- receiver channel.
----------------	---------------------

Definition at line 121 of file [pcps\\_cccwsr\\_acquisition\\_cc.h](#).

**10.314.3.6 set\_channel\_fsm()**

```
void pcps_cccwsr_acquisition_cc::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline]
```

Set channel fsm associated to this acquisition instance.

Definition at line 129 of file [pcps\\_cccwsr\\_acquisition\\_cc.h](#).

**10.314.3.7 set\_doppler\_max()**

```
void pcps_cccwsr_acquisition_cc::set_doppler_max (
    uint32_t doppler_max) [inline]
```

Set maximum Doppler grid search.

## Parameters

<i>doppler_max</i>	- Maximum Doppler shift considered in the grid search [Hz].
--------------------	---

Definition at line 148 of file [pcps\\_cccwsr\\_acquisition\\_cc.h](#).

**10.314.3.8 set\_doppler\_step()**

```
void pcps_cccwsr_acquisition_cc::set_doppler_step (
    uint32_t doppler_step) [inline]
```

Set Doppler steps for the grid search.

## Parameters

<i>doppler_step</i>	- Frequency bin of the search grid [Hz].
---------------------	--

Definition at line 157 of file [pcps\\_cccwsr\\_acquisition\\_cc.h](#).

**10.314.3.9 set\_gnss\_synchro()**

```
void pcps_cccwsr_acquisition_cc::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [inline]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to exchange synchronization data between acquisition and tracking blocks.

## Parameters

<i>p_gnss_synchro</i>	Satellite information shared by the processing blocks.
-----------------------	--

Definition at line 75 of file [pcps\\_cccwsr\\_acquisition\\_cc.h](#).

**10.314.3.10 set\_local\_code()**

```
void pcps_cccwsr_acquisition_cc::set_local_code (
    std::complex< float > * code_data,
    std::complex< float > * code_pilot)
```

Sets local code for CCCWSR acquisition algorithm.

**Parameters**

<i>data_code</i>	- Pointer to the data PRN code.
<i>pilot_code</i>	- Pointer to the pilot PRN code.

**10.314.3.11 set\_state()**

```
void pcps_cccwsr_acquisition_cc::set_state (
    int32_t state)
```

If set to 1, ensures that acquisition starts at the first available sample.

**Parameters**

<i>state</i>	- int=1 forces start of acquisition
--------------	-------------------------------------

**10.314.3.12 set\_threshold()**

```
void pcps_cccwsr_acquisition_cc::set_threshold (
    float threshold) [inline]
```

Set statistics threshold of CCCWSR algorithm.

**Parameters**

<i>threshold</i>	- Threshold for signal detection (check <a href="#">Navitec2012</a> , Algorithm 1, for a definition of this threshold).
------------------	---

Definition at line 139 of file [pcps\\_cccwsr\\_acquisition\\_cc.h](#).

The documentation for this class was generated from the following file:

- [pcps\\_cccwsr\\_acquisition\\_cc.h](#)

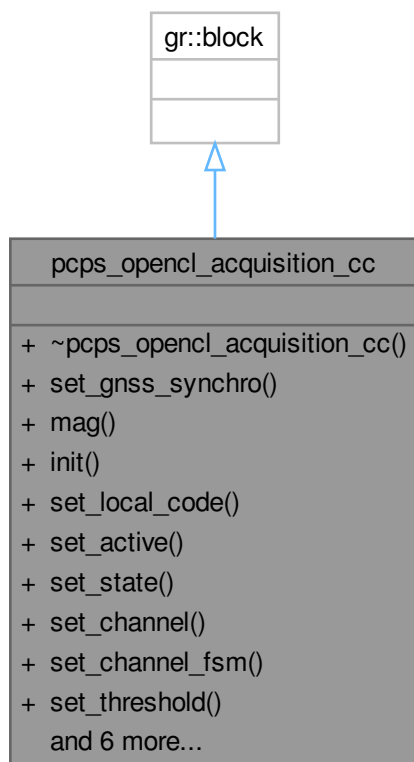
**10.315 pcps\_openc1\_acquisition\_cc Class Reference**

This class implements a Parallel Code Phase Search Acquisition.

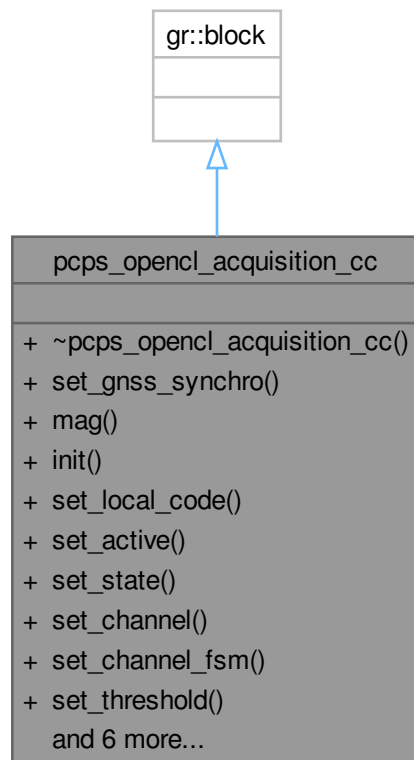
```
#include <pcps_openc1_acquisition_cc.h>
```



Inheritance diagram for pcps\_openc1\_acquisition\_cc:



Collaboration diagram for `pcps_openc1_acquisition_cc`:



## Public Member Functions

- `~pcps_openc1_acquisition_cc ()`  
*Default destructor.*
- void `set_gnss_synchro (Gnss_Synchro *p_gnss_synchro)`  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to exchange synchronization data between acquisition and tracking blocks.*
- `uint32_t mag () const`  
*Returns the maximum peak of grid search.*
- void `init ()`  
*Initializes acquisition algorithm.*
- void `set_local_code (std::complex< float > *code)`  
*Sets local code for PCPS acquisition algorithm.*
- void `set_active (bool active)`  
*Starts acquisition algorithm, turning from standby mode to active mode.*
- void `set_state (int state)`  
*If set to 1, ensures that acquisition starts at the first available sample.*
- void `set_channel (uint32_t channel)`  
*Set acquisition channel unique ID.*
- void `set_channel_fsm (std::weak_ptr< ChannelFsm > channel_fsm)`  
*Set channel fsm associated to this acquisition instance.*

- void [set\\_threshold](#) (float threshold)  
*Set statistics threshold of PCPS algorithm.*
- void [set\\_doppler\\_max](#) (uint32\_t doppler\_max)  
*Set maximum Doppler grid search.*
- void [set\\_doppler\\_step](#) (uint32\_t doppler\_step)  
*Set Doppler steps for the grid search.*
- bool [openc1\\_ready](#) () const
- void [acquisition\\_core\\_volk](#) ()
- void [acquisition\\_core\\_openc1](#) ()
- int [general\\_work](#) (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)  
*Parallel Code Phase Search Acquisition signal processing.*

### Friends

- pcps\_openc1\_acquisition\_cc\_sptr [pcps\\_make\\_openc1\\_acquisition\\_cc](#) (uint32\_t sampled\_ms, uint32\_t max\_dwells, uint32\_t doppler\_max, int64\_t fs\_in, int samples\_per\_ms, int samples\_per\_code, bool bit\_transition\_flag, bool dump, const std::string &dump\_filename, bool enable\_monitor\_output)

## 10.315.1 Detailed Description

This class implements a Parallel Code Phase Search Acquisition.

Check [An Open Source Galileo E1 Software Receiver](#), Algorithm 1, for a pseudocode description of this implementation.

Definition at line 83 of file [pcps\\_openc1\\_acquisition\\_cc.h](#).

## 10.315.2 Constructor & Destructor Documentation

### 10.315.2.1 ~pcps\_openc1\_acquisition\_cc()

```
pcps_openc1_acquisition_cc::~pcps_openc1_acquisition_cc ()
```

Default destructor.

## 10.315.3 Member Function Documentation

### 10.315.3.1 general\_work()

```
int pcps_openc1_acquisition_cc::general_work (
    int noutput_items,
    gr_vector_int & ninput_items,
    gr_vector_const_void_star & input_items,
    gr_vector_void_star & output_items)
```

Parallel Code Phase Search Acquisition signal processing.

### 10.315.3.2 init()

```
void pcps_openc1_acquisition_cc::init ()
```

Initializes acquisition algorithm.

### 10.315.3.3 mag()

```
uint32_t pcps_openc1_acquisition_cc::mag () const [inline]
```

Returns the maximum peak of grid search.  
Definition at line 104 of file [pcps\\_openc1\\_acquisition\\_cc.h](#).

### 10.315.3.4 openc1\_ready()

```
bool pcps_openc1_acquisition_cc::openc1_ready () const [inline]
```

Definition at line 182 of file [pcps\\_openc1\\_acquisition\\_cc.h](#).

**10.315.3.5 set\_active()**

```
void pcps_openc1_acquisition_cc::set_active (
    bool active) [inline]
```

Starts acquisition algorithm, turning from standby mode to active mode.

**Parameters**

<i>active</i>	- bool that activates/deactivates the block.
---------------	--

Definition at line 125 of file [pcps\\_openc1\\_acquisition\\_cc.h](#).

**10.315.3.6 set\_channel()**

```
void pcps_openc1_acquisition_cc::set_channel (
    uint32_t channel) [inline]
```

Set acquisition channel unique ID.

**Parameters**

<i>channel</i>	- receiver channel.
----------------	---------------------

Definition at line 141 of file [pcps\\_openc1\\_acquisition\\_cc.h](#).

**10.315.3.7 set\_channel\_fsm()**

```
void pcps_openc1_acquisition_cc::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline]
```

Set channel fsm associated to this acquisition instance.

Definition at line 149 of file [pcps\\_openc1\\_acquisition\\_cc.h](#).

**10.315.3.8 set\_doppler\_max()**

```
void pcps_openc1_acquisition_cc::set_doppler_max (
    uint32_t doppler_max) [inline]
```

Set maximum Doppler grid search.

**Parameters**

<i>doppler_max</i>	- Maximum Doppler shift considered in the grid search [Hz].
--------------------	---

Definition at line 168 of file [pcps\\_openc1\\_acquisition\\_cc.h](#).

**10.315.3.9 set\_doppler\_step()**

```
void pcps_openc1_acquisition_cc::set_doppler_step (
    uint32_t doppler_step) [inline]
```

Set Doppler steps for the grid search.

**Parameters**

<i>doppler_step</i>	- Frequency bin of the search grid [Hz].
---------------------	--

Definition at line 177 of file [pcps\\_openc1\\_acquisition\\_cc.h](#).

### 10.315.3.10 set\_gnss\_synchro()

```
void pcps_openc1_acquisition_cc::set_gnss_synchro (  
    Gnss_Synchro * p_gnss_synchro) [inline]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to exchange synchronization data between acquisition and tracking blocks.

## Parameters

<i>p_gnss_synchro</i>	Satellite information shared by the processing blocks.
-----------------------	--

Definition at line 96 of file [pcps\\_openc1\\_acquisition\\_cc.h](#).

**10.315.3.11 set\_local\_code()**

```
void pcps_openc1_acquisition_cc::set_local_code (
    std::complex< float > * code)
```

Sets local code for PCPS acquisition algorithm.

## Parameters

<i>code</i>	- Pointer to the PRN code.
-------------	----------------------------

**10.315.3.12 set\_state()**

```
void pcps_openc1_acquisition_cc::set_state (
    int state)
```

If set to 1, ensures that acquisition starts at the first available sample.

## Parameters

<i>state</i>	- int=1 forces start of acquisition
--------------	-------------------------------------

**10.315.3.13 set\_threshold()**

```
void pcps_openc1_acquisition_cc::set_threshold (
    float threshold) [inline]
```

Set statistics threshold of PCPS algorithm.

## Parameters

<i>threshold</i>	- Threshold for signal detection (check <a href="#">Navitec2012</a> , Algorithm 1, for a definition of this threshold).
------------------	---

Definition at line 159 of file [pcps\\_openc1\\_acquisition\\_cc.h](#).

The documentation for this class was generated from the following file:

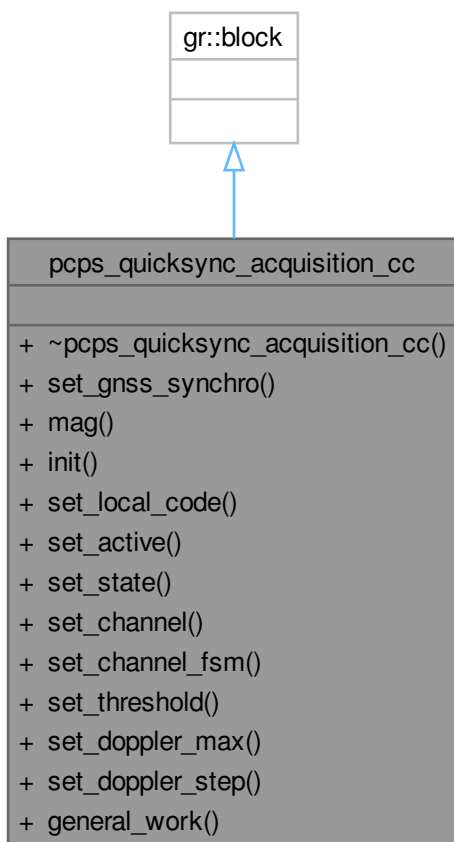
- [pcps\\_openc1\\_acquisition\\_cc.h](#)

**10.316 pcps\_quicksync\_acquisition\_cc Class Reference**

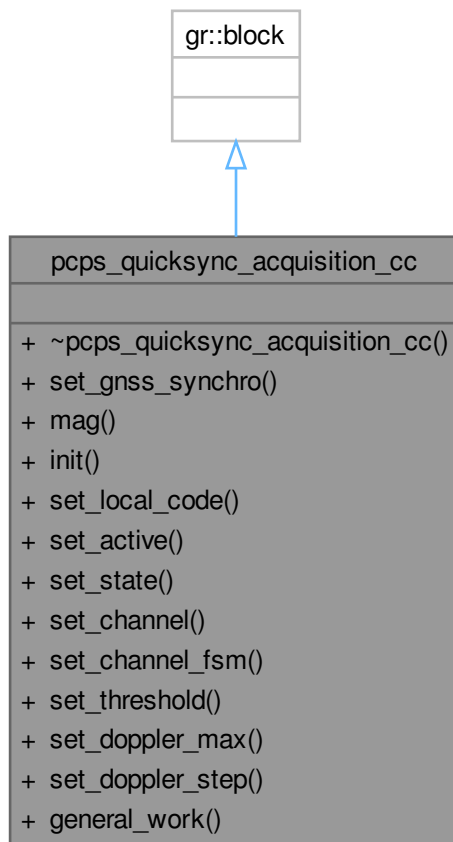
This class implements a Parallel Code Phase Search Acquisition with the implementation of the Sparse QuickSync Algorithm.

```
#include <pcps_quicksync_acquisition_cc.h>
```

Inheritance diagram for pcps\_quicksync\_acquisition\_cc:



Collaboration diagram for pcps\_quicksync\_acquisition\_cc:



## Public Member Functions

- `~pcps_quicksync_acquisition_cc ()`  
*Default destructor.*
- void `set_gnss_synchro (Gnss_Synchro *p_gnss_synchro)`  
*Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to exchange synchronization data between acquisition and tracking blocks.*
- `uint32_t mag () const`  
*Returns the maximum peak of grid search.*
- void `init ()`  
*Initializes acquisition algorithm.*
- void `set_local_code (std::complex< float > *code)`  
*Sets local code for PCPS acquisition algorithm.*
- void `set_active (bool active)`  
*Starts acquisition algorithm, turning from standby mode to active mode.*
- void `set_state (int32_t state)`  
*If set to 1, ensures that acquisition starts at the first available sample.*
- void `set_channel (uint32_t channel)`  
*Set acquisition channel unique ID.*



- void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm)  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold)  
*Set statistics threshold of PCPS algorithm.*
- void [set\\_doppler\\_max](#) (uint32\_t doppler\_max)  
*Set maximum Doppler grid search.*
- void [set\\_doppler\\_step](#) (uint32\_t doppler\_step)  
*Set Doppler steps for the grid search.*
- int [general\\_work](#) (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)  
*Parallel Code Phase Search Acquisition signal processing.*

## Friends

- pcps\_quicksync\_acquisition\_cc\_sptr [pcps\\_quicksync\\_make\\_acquisition\\_cc](#) (uint32\_t folding\_factor, uint32\_t sampled\_ms, uint32\_t max\_dwells, uint32\_t doppler\_max, int64\_t fs\_in, int32\_t samples\_per\_ms, int32\_t samples\_per\_code, bool bit\_transition\_flag, bool dump, const std::string &dump\_filename, bool enable\_monitor\_output)

## 10.316.1 Detailed Description

This class implements a Parallel Code Phase Search Acquisition with the implementation of the Sparse QuickSync Algorithm.

Check [Faster GPS via the Sparse Fourier Transform](#), for details of its implementation and functionality.

Definition at line 84 of file [pcps\\_quicksync\\_acquisition\\_cc.h](#).

## 10.316.2 Constructor & Destructor Documentation

### 10.316.2.1 ~pcps\_quicksync\_acquisition\_cc()

```
pcps_quicksync_acquisition_cc::~pcps_quicksync_acquisition_cc ()
```

Default destructor.

## 10.316.3 Member Function Documentation

### 10.316.3.1 general\_work()

```
int pcps_quicksync_acquisition_cc::general_work (
    int noutput_items,
    gr_vector_int & ninput_items,
    gr_vector_const_void_star & input_items,
    gr_vector_void_star & output_items)
```

Parallel Code Phase Search Acquisition signal processing.

### 10.316.3.2 init()

```
void pcps_quicksync_acquisition_cc::init ()
```

Initializes acquisition algorithm.

### 10.316.3.3 mag()

```
uint32_t pcps_quicksync_acquisition_cc::mag () const [inline]
```

Returns the maximum peak of grid search.  
Definition at line 105 of file [pcps\\_quicksync\\_acquisition\\_cc.h](#).

#### 10.316.3.4 set\_active()

```
void pcps_quicksync_acquisition_cc::set_active (
    bool active) [inline]
```

Starts acquisition algorithm, turning from standby mode to active mode.

##### Parameters

<i>active</i>	- bool that activates/deactivates the block.
---------------	--

Definition at line 126 of file [pcps\\_quicksync\\_acquisition\\_cc.h](#).

#### 10.316.3.5 set\_channel()

```
void pcps_quicksync_acquisition_cc::set_channel (
    uint32_t channel) [inline]
```

Set acquisition channel unique ID.

##### Parameters

<i>channel</i>	- receiver channel.
----------------	---------------------

Definition at line 142 of file [pcps\\_quicksync\\_acquisition\\_cc.h](#).

#### 10.316.3.6 set\_channel\_fsm()

```
void pcps_quicksync_acquisition_cc::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline]
```

Set channel fsm associated to this acquisition instance.

Definition at line 150 of file [pcps\\_quicksync\\_acquisition\\_cc.h](#).

#### 10.316.3.7 set\_doppler\_max()

```
void pcps_quicksync_acquisition_cc::set_doppler_max (
    uint32_t doppler_max) [inline]
```

Set maximum Doppler grid search.

##### Parameters

<i>doppler_max</i>	- Maximum Doppler shift considered in the grid search [Hz].
--------------------	---

Definition at line 169 of file [pcps\\_quicksync\\_acquisition\\_cc.h](#).

#### 10.316.3.8 set\_doppler\_step()

```
void pcps_quicksync_acquisition_cc::set_doppler_step (
    uint32_t doppler_step) [inline]
```

Set Doppler steps for the grid search.

##### Parameters

<i>doppler_step</i>	- Frequency bin of the search grid [Hz].
---------------------	--

Definition at line 178 of file [pcps\\_quicksync\\_acquisition\\_cc.h](#).

### 10.316.3.9 set\_gnss\_synchro()

```
void pcps_quicksync_acquisition_cc::set_gnss_synchro (  
    Gnss_Synchro * p_gnss_synchro) [inline]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to exchange synchronization data between acquisition and tracking blocks.

## Parameters

<i>p_gnss_synchro</i>	Satellite information shared by the processing blocks.
-----------------------	--

Definition at line 97 of file [pcps\\_quicksync\\_acquisition\\_cc.h](#).

**10.316.3.10 set\_local\_code()**

```
void pcps_quicksync_acquisition_cc::set_local_code (
    std::complex< float > * code)
```

Sets local code for PCPS acquisition algorithm.

## Parameters

<i>code</i>	- Pointer to the PRN code.
-------------	----------------------------

**10.316.3.11 set\_state()**

```
void pcps_quicksync_acquisition_cc::set_state (
    int32_t state)
```

If set to 1, ensures that acquisition starts at the first available sample.

## Parameters

<i>state</i>	- int=1 forces start of acquisition
--------------	-------------------------------------

**10.316.3.12 set\_threshold()**

```
void pcps_quicksync_acquisition_cc::set_threshold (
    float threshold) [inline]
```

Set statistics threshold of PCPS algorithm.

## Parameters

<i>threshold</i>	- Threshold for signal detection (check <a href="#">Navitec2012</a> , Algorithm 1, for a definition of this threshold).
------------------	---

Definition at line 160 of file [pcps\\_quicksync\\_acquisition\\_cc.h](#).

The documentation for this class was generated from the following file:

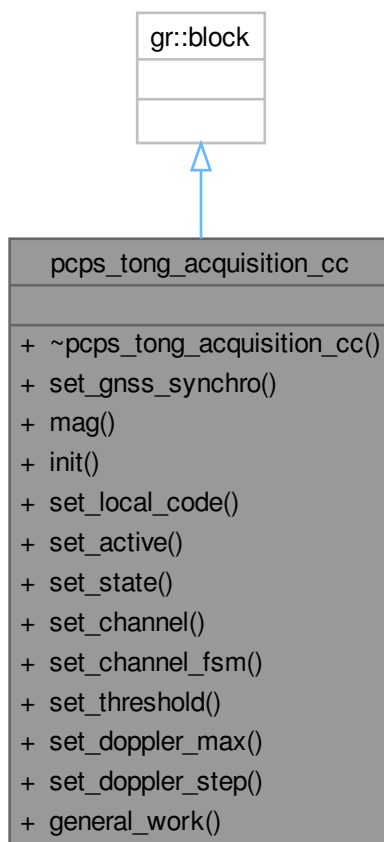
- [pcps\\_quicksync\\_acquisition\\_cc.h](#)

**10.317 pcps\_tong\_acquisition\_cc Class Reference**

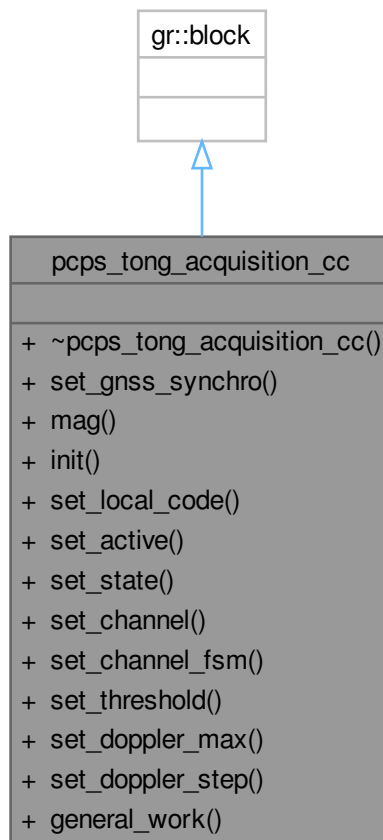
This class implements a Parallel Code Phase Search Acquisition with Tong algorithm.

```
#include <pcps_tong_acquisition_cc.h>
```

Inheritance diagram for pcps\_tong\_acquisition\_cc:



Collaboration diagram for `pcps_tong_acquisition_cc`:



## Public Member Functions

- `~pcps_tong_acquisition_cc ()`  
*Default destructor.*
- void `set_gnss_synchro (Gnss_Synchro *p_gnss_synchro)`  
*Set acquisition/tracking common `Gnss_Synchro` object pointer to exchange synchronization data between acquisition and tracking blocks.*
- `uint32_t mag () const`  
*Returns the maximum peak of grid search.*
- void `init ()`  
*Initializes acquisition algorithm.*
- void `set_local_code (std::complex< float > *code)`  
*Sets local code for TONG acquisition algorithm.*
- void `set_active (bool active)`  
*Starts acquisition algorithm, turning from standby mode to active mode.*
- void `set_state (int32_t state)`  
*If set to 1, ensures that acquisition starts at the first available sample.*
- void `set_channel (uint32_t channel)`  
*Set acquisition channel unique ID.*

- void [set\\_channel\\_fsm](#) (std::weak\_ptr< [ChannelFsm](#) > channel\_fsm)  
*Set channel fsm associated to this acquisition instance.*
- void [set\\_threshold](#) (float threshold)  
*Set statistics threshold of TONG algorithm.*
- void [set\\_doppler\\_max](#) (uint32\_t doppler\_max)  
*Set maximum Doppler grid search.*
- void [set\\_doppler\\_step](#) (uint32\_t doppler\_step)  
*Set Doppler steps for the grid search.*
- int [general\\_work](#) (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)  
*Parallel Code Phase Search Acquisition signal processing.*

## Friends

- pcps\_tong\_acquisition\_cc\_sptr [pcps\\_tong\\_make\\_acquisition\\_cc](#) (uint32\_t sampled\_ms, uint32\_t doppler\_max, int64\_t fs\_in, int32\_t samples\_per\_ms, int32\_t samples\_per\_code, uint32\_t tong\_init\_val, uint32\_t tong\_max\_val, uint32\_t tong\_max\_dwells, bool dump, const std::string &dump\_filename, bool enable\_monitor\_output)

## 10.317.1 Detailed Description

This class implements a Parallel Code Phase Search Acquisition with Tong algorithm.  
Definition at line 78 of file [pcps\\_tong\\_acquisition\\_cc.h](#).

## 10.317.2 Constructor & Destructor Documentation

### 10.317.2.1 ~pcps\_tong\_acquisition\_cc()

`pcps_tong_acquisition_cc::~~pcps_tong_acquisition_cc ()`  
Default destructor.

## 10.317.3 Member Function Documentation

### 10.317.3.1 general\_work()

```
int pcps_tong_acquisition_cc::general_work (
    int noutput_items,
    gr_vector_int & ninput_items,
    gr_vector_const_void_star & input_items,
    gr_vector_void_star & output_items)
```

Parallel Code Phase Search Acquisition signal processing.

### 10.317.3.2 init()

`void pcps_tong_acquisition_cc::init ()`  
Initializes acquisition algorithm.

### 10.317.3.3 mag()

`uint32_t pcps_tong_acquisition_cc::mag () const [inline]`  
Returns the maximum peak of grid search.  
Definition at line 99 of file [pcps\\_tong\\_acquisition\\_cc.h](#).

### 10.317.3.4 set\_active()

`void pcps_tong_acquisition_cc::set_active (
 bool active) [inline]`  
Starts acquisition algorithm, turning from standby mode to active mode.

## Parameters

<i>active</i>	- bool that activates/deactivates the block.
---------------	--

Definition at line 120 of file [pcps\\_tong\\_acquisition\\_cc.h](#).

**10.317.3.5 set\_channel()**

```
void pcps_tong_acquisition_cc::set_channel (
    uint32_t channel) [inline]
```

Set acquisition channel unique ID.

## Parameters

<i>channel</i>	- receiver channel.
----------------	---------------------

Definition at line 136 of file [pcps\\_tong\\_acquisition\\_cc.h](#).

**10.317.3.6 set\_channel\_fsm()**

```
void pcps_tong_acquisition_cc::set_channel_fsm (
    std::weak_ptr< ChannelFsm > channel_fsm) [inline]
```

Set channel fsm associated to this acquisition instance.

Definition at line 144 of file [pcps\\_tong\\_acquisition\\_cc.h](#).

**10.317.3.7 set\_doppler\_max()**

```
void pcps_tong_acquisition_cc::set_doppler_max (
    uint32_t doppler_max) [inline]
```

Set maximum Doppler grid search.

## Parameters

<i>doppler_max</i>	- Maximum Doppler shift considered in the grid search [Hz].
--------------------	---

Definition at line 163 of file [pcps\\_tong\\_acquisition\\_cc.h](#).

**10.317.3.8 set\_doppler\_step()**

```
void pcps_tong_acquisition_cc::set_doppler_step (
    uint32_t doppler_step) [inline]
```

Set Doppler steps for the grid search.

## Parameters

<i>doppler_step</i>	- Frequency bin of the search grid [Hz].
---------------------	--

Definition at line 172 of file [pcps\\_tong\\_acquisition\\_cc.h](#).

**10.317.3.9 set\_gnss\_synchro()**

```
void pcps_tong_acquisition_cc::set_gnss_synchro (
    Gnss_Synchro * p_gnss_synchro) [inline]
```

Set acquisition/tracking common [Gnss\\_Synchro](#) object pointer to exchange synchronization data between acquisition and tracking blocks.

## Parameters

<i>p_gnss_synchro</i>	Satellite information shared by the processing blocks.
-----------------------	--

Definition at line 91 of file [pcps\\_tong\\_acquisition\\_cc.h](#).



**10.317.3.10 set\_local\_code()**

```
void pcps_tong_acquisition_cc::set_local_code (
    std::complex< float > * code)
```

Sets local code for TONG acquisition algorithm.

**Parameters**

<i>code</i>	- Pointer to the PRN code.
-------------	----------------------------

**10.317.3.11 set\_state()**

```
void pcps_tong_acquisition_cc::set_state (
    int32_t state)
```

If set to 1, ensures that acquisition starts at the first available sample.

**Parameters**

<i>state</i>	- int=1 forces start of acquisition
--------------	-------------------------------------

**10.317.3.12 set\_threshold()**

```
void pcps_tong_acquisition_cc::set_threshold (
    float threshold) [inline]
```

Set statistics threshold of TONG algorithm.

**Parameters**

<i>threshold</i>	- Threshold for signal detection (check <a href="#">Navitec2012</a> , Algorithm 1, for a definition of this threshold).
------------------	---

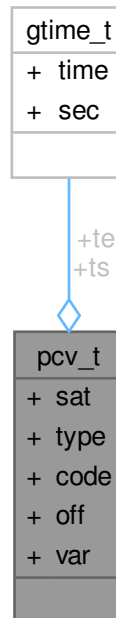
Definition at line 154 of file [pcps\\_tong\\_acquisition\\_cc.h](#).

The documentation for this class was generated from the following file:

- [pcps\\_tong\\_acquisition\\_cc.h](#)

## 10.318 pcv\_t Struct Reference

Collaboration diagram for pcv\_t:



### Public Attributes

- int [sat](#)
- char [type](#) [[MAXANT](#)]
- char [code](#) [[MAXANT](#)]
- [gtime\\_t](#) [ts](#)
- [gtime\\_t](#) [te](#)
- double [off](#) [[NFREQ](#)][3]
- double [var](#) [[NFREQ](#)][19]

### 10.318.1 Detailed Description

Definition at line [399](#) of file [rtklib.h](#).

### 10.318.2 Member Data Documentation

#### 10.318.2.1 code

```
char pcv_t::code[MAXANT]
```

Definition at line [403](#) of file [rtklib.h](#).

#### 10.318.2.2 off

```
double pcv_t::off[NFREQ][3]
```

Definition at line [405](#) of file [rtklib.h](#).

### 10.318.2.3 sat

```
int pcv_t::sat
```

Definition at line 401 of file [rtklib.h](#).

### 10.318.2.4 te

```
gtime_t pcv_t::te
```

Definition at line 404 of file [rtklib.h](#).

### 10.318.2.5 ts

```
gtime_t pcv_t::ts
```

Definition at line 404 of file [rtklib.h](#).

### 10.318.2.6 type

```
char pcv_t::type[MAXANT]
```

Definition at line 402 of file [rtklib.h](#).

### 10.318.2.7 var

```
double pcv_t::var[NFREQ][19]
```

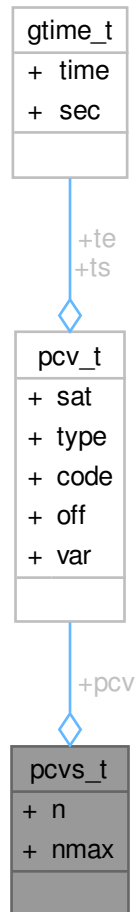
Definition at line 406 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.319 pcvs\_t Struct Reference

Collaboration diagram for pcvs\_t:



### Public Attributes

- `int n`
- `int nmax`
- `pcv_t * pcv`

### 10.319.1 Detailed Description

Definition at line 411 of file [rtklib.h](#).

### 10.319.2 Member Data Documentation

#### 10.319.2.1 n

```
int pcvs_t::n
```

Definition at line 413 of file [rtklib.h](#).

### 10.319.2.2 nmax

```
int pcvs_t::nmax
```

Definition at line 413 of file [rtklib.h](#).

### 10.319.2.3 pcv

```
pcv_t* pcvs_t::pcv
```

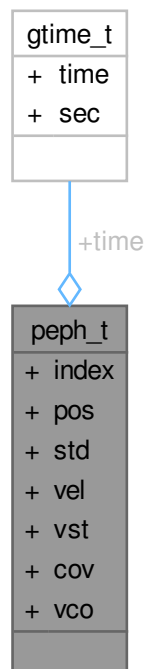
Definition at line 414 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.320 peph\_t Struct Reference

Collaboration diagram for peph\_t:



### Public Attributes

- [gtime\\_t](#) `time`
- int `index`
- double `pos` [MAXSAT][4]
- float `std` [MAXSAT][4]
- double `vel` [MAXSAT][4]
- float `vst` [MAXSAT][4]
- float `cov` [MAXSAT][3]
- float `vco` [MAXSAT][3]

### 10.320.1 Detailed Description

Definition at line 478 of file [rtklib.h](#).

### 10.320.2 Member Data Documentation

#### 10.320.2.1 cov

```
float peph_t::cov[MAXSAT][3]
```

Definition at line 486 of file [rtklib.h](#).

#### 10.320.2.2 index

```
int peph_t::index
```

Definition at line 481 of file [rtklib.h](#).

#### 10.320.2.3 pos

```
double peph_t::pos[MAXSAT][4]
```

Definition at line 482 of file [rtklib.h](#).

#### 10.320.2.4 std

```
float peph_t::std[MAXSAT][4]
```

Definition at line 483 of file [rtklib.h](#).

#### 10.320.2.5 time

```
gtime_t peph_t::time
```

Definition at line 480 of file [rtklib.h](#).

#### 10.320.2.6 vco

```
float peph_t::vco[MAXSAT][3]
```

Definition at line 487 of file [rtklib.h](#).

#### 10.320.2.7 vel

```
double peph_t::vel[MAXSAT][4]
```

Definition at line 484 of file [rtklib.h](#).

#### 10.320.2.8 vst

```
float peph_t::vst[MAXSAT][4]
```

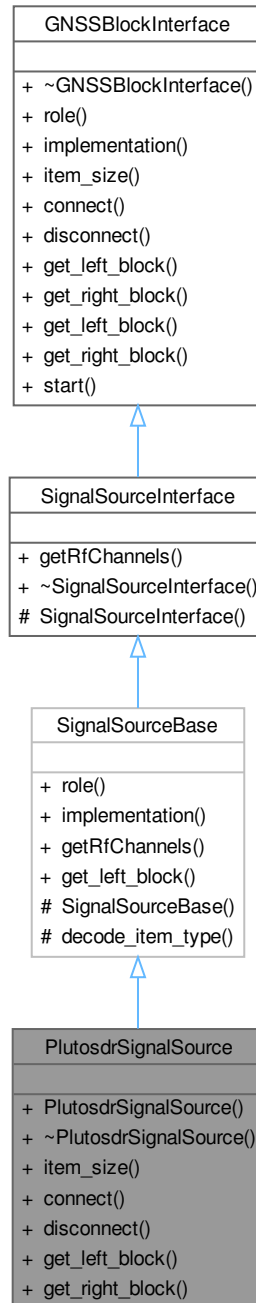
Definition at line 485 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

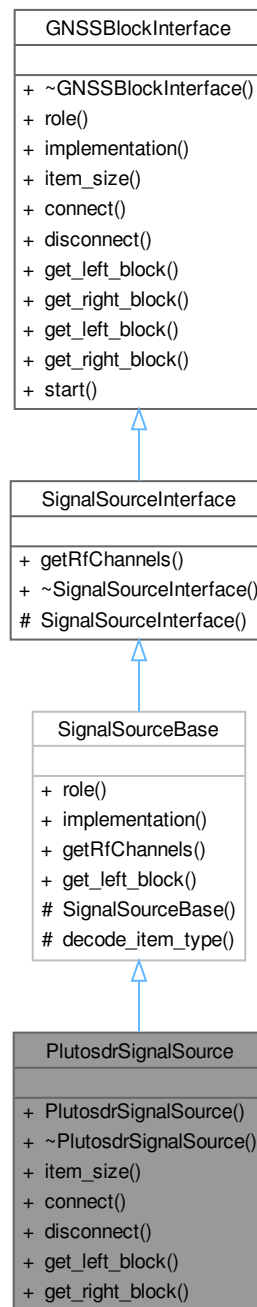
- [rtklib.h](#)

## 10.321 PlutosdrSignalSource Class Reference

Inheritance diagram for PlutosdrSignalSource:



Collaboration diagram for PlutosdrSignalSource:



## Public Member Functions

- **PlutosdrSignalSource** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in↔\_stream, unsigned int out\_stream, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override



## Public Member Functions inherited from [SignalSourceBase](#)

- `std::string` [role](#) () final
- `std::string` [implementation](#) () final
- `size_t` [getRfChannels](#) () const override
- `gr::basic_block_sptr` [get\\_left\\_block](#) () override

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual `gr::basic_block_sptr` [get\\_left\\_block](#) (int RF\_channel)
- virtual `gr::basic_block_sptr` [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

## Additional Inherited Members

## Protected Member Functions inherited from [SignalSourceBase](#)

- [SignalSourceBase](#) ([ConfigurationInterface](#) const \*configuration, `std::string` role, `std::string` impl)  
*Constructor.*
- `size_t` [decode\\_item\\_type](#) (`std::string` const &item\_type, `bool` \*is\_interleaved=nullptr, `bool` throw\_on\_↵ error=false)  
*utility for decoding passed ".item\_type" values*

### 10.321.1 Detailed Description

Definition at line 49 of file [plutosdr\\_signal\\_source.h](#).

### 10.321.2 Member Function Documentation

#### 10.321.2.1 connect()

```
void PlutosdrSignalSource::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.321.2.2 disconnect()

```
void PlutosdrSignalSource::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.321.2.3 get\_left\_block()

```
gr::basic_block_sptr PlutosdrSignalSource::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.321.2.4 get\_right\_block()

```
gr::basic_block_sptr PlutosdrSignalSource::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.321.2.5 item\_size()

```
size_t PlutosdrSignalSource::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

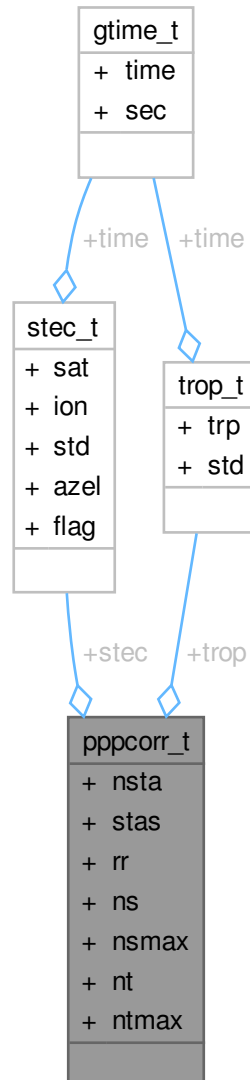
Definition at line 58 of file [plutosdr\\_signal\\_source.h](#).

The documentation for this class was generated from the following file:

- [plutosdr\\_signal\\_source.h](#)

## 10.322 pppcorr\_t Struct Reference

Collaboration diagram for pppcorr\_t:



### Public Attributes

- int `nsta`
- char `stas` [MAXSTA][8]
- double `rr` [MAXSTA][3]
- int `ns` [MAXSTA]
- int `nsmax` [MAXSTA]
- int `nt` [MAXSTA]
- int `ntmax` [MAXSTA]
- `stec_t` \* `stec` [MAXSTA]
- `trop_t` \* `trop` [MAXSTA]

### 10.322.1 Detailed Description

Definition at line 740 of file [rtklib.h](#).

### 10.322.2 Member Data Documentation

#### 10.322.2.1 ns

```
int pppcorr_t::ns[MAXSTA]
```

Definition at line 745 of file [rtklib.h](#).

#### 10.322.2.2 nsmax

```
int pppcorr_t::nsmax[MAXSTA]
```

Definition at line 745 of file [rtklib.h](#).

#### 10.322.2.3 nsta

```
int pppcorr_t::nsta
```

Definition at line 742 of file [rtklib.h](#).

#### 10.322.2.4 nt

```
int pppcorr_t::nt[MAXSTA]
```

Definition at line 746 of file [rtklib.h](#).

#### 10.322.2.5 ntmax

```
int pppcorr_t::ntmax[MAXSTA]
```

Definition at line 746 of file [rtklib.h](#).

#### 10.322.2.6 rr

```
double pppcorr_t::rr[MAXSTA][3]
```

Definition at line 744 of file [rtklib.h](#).

#### 10.322.2.7 stas

```
char pppcorr_t::stas[MAXSTA][8]
```

Definition at line 743 of file [rtklib.h](#).

#### 10.322.2.8 stec

```
stec_t* pppcorr_t::stec[MAXSTA]
```

Definition at line 747 of file [rtklib.h](#).

#### 10.322.2.9 trop

```
trop_t* pppcorr_t::trop[MAXSTA]
```

Definition at line 748 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.323 pps\_tcp\_rx Class Reference

Collaboration diagram for pps\_tcp\_rx:

pps_tcp_rx
+ is_connected
+ pps_tcp_rx()
+ ~pps_tcp_rx()
+ receive_pps()
+ send_cmd()
+ set_pps_samplestamp_queue()

### Public Member Functions

- void **receive\_pps** (const std::string &ip\_address, int port)
- bool **send\_cmd** (std::string cmd) const
- void **set\_pps\_samplestamp\_queue** (std::shared\_ptr< [Concurrent\\_Queue](#)< [PpsSamplestamp](#) > > queue)

### Public Attributes

- volatile bool [is\\_connected](#) {false}

### 10.323.1 Detailed Description

Definition at line 27 of file [ppstcprx.h](#).

### 10.323.2 Member Data Documentation

#### 10.323.2.1 is\_connected

```
volatile bool pps_tcp_rx::is_connected {false}
```

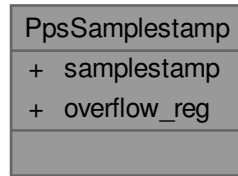
Definition at line 34 of file [ppstcprx.h](#).

The documentation for this class was generated from the following file:

- [ppstcprx.h](#)

## 10.324 PpsSamplestamp Class Reference

Collaboration diagram for PpsSamplestamp:



### Public Attributes

- `uint64_t samplestamp = 0`
- `uint32_t overflow_reg = 0`

### 10.324.1 Detailed Description

Definition at line 21 of file [pps\\_samplestamp.h](#).

### 10.324.2 Member Data Documentation

#### 10.324.2.1 overflow\_reg

`uint32_t PpsSamplestamp::overflow_reg = 0`

Definition at line 25 of file [pps\\_samplestamp.h](#).

#### 10.324.2.2 samplestamp

`uint64_t PpsSamplestamp::samplestamp = 0`

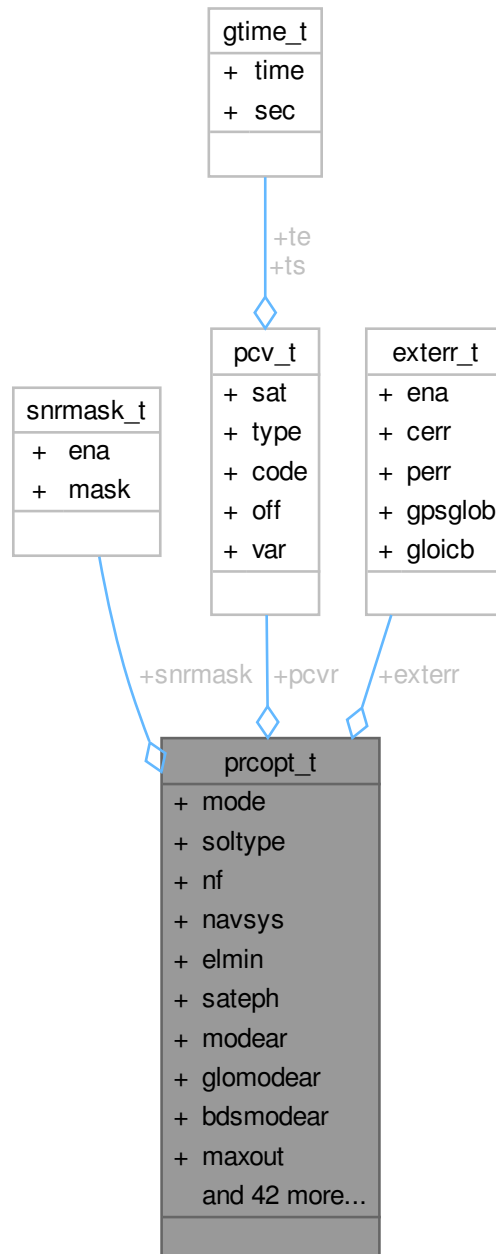
Definition at line 24 of file [pps\\_samplestamp.h](#).

The documentation for this class was generated from the following file:

- [pps\\_samplestamp.h](#)

## 10.325 prcopt\_t Struct Reference

Collaboration diagram for prcopt\_t:



### Public Attributes

- int [mode](#)
- int [soltype](#)
- int [nf](#)
- int [navsys](#)
- double [elmin](#)

- [snrmask\\_t](#) snrmask
- int [sateph](#)
- int [modear](#)
- int [glomodear](#)
- int [bdsmodear](#)
- int [maxout](#)
- int [minlock](#)
- int [minfix](#)
- int [armaxiter](#)
- int [ionoopt](#)
- int [tropopt](#)
- int [dynamics](#)
- int [tidecorr](#)
- int [niter](#)
- int [codesmooth](#)
- int [intpref](#)
- int [sbascorr](#)
- int [sbassatsel](#)
- int [rovpos](#)
- int [refpos](#)
- double [eratio](#) [NFREQ]
- double [err](#) [5]
- double [std](#) [3]
- double [prn](#) [6]
- double [sclkstab](#)
- double [thresar](#) [8]
- double [elmaskar](#)
- double [elmaskhold](#)
- double [thresslip](#)
- double [maxtdiff](#)
- double [maxinno](#)
- double [maxgdop](#)
- double [baseline](#) [2]
- double [ru](#) [3]
- double [rb](#) [3]
- char [anttype](#) [2][MAXANT]
- double [antdel](#) [2][3]
- [pcv\\_t](#) [pcvr](#) [2]
- unsigned char [exsats](#) [MAXSAT]
- int [maxaveep](#)
- int [initrst](#)
- int [outsingle](#)
- char [rxopt](#) [2][256]
- int [posopt](#) [6]
- int [syncsol](#)
- double [odisp](#) [2][6 \* 11]
- [exterr\\_t](#) [exterr](#)
- int [freqopt](#)
- char [pppopt](#) [256]
- bool [bancroft\\_init](#)

### 10.325.1 Detailed Description

Definition at line 942 of file [rtklib.h](#).

## 10.325.2 Member Data Documentation

### 10.325.2.1 antdel

double prcopt\_t::antdel[2][3]  
Definition at line 990 of file [rtklib.h](#).

### 10.325.2.2 anttype

char prcopt\_t::anttype[2][MAXANT]  
Definition at line 989 of file [rtklib.h](#).

### 10.325.2.3 armaxiter

int prcopt\_t::armaxiter  
Definition at line 957 of file [rtklib.h](#).

### 10.325.2.4 bancroft\_init

bool prcopt\_t::bancroft\_init  
Definition at line 1003 of file [rtklib.h](#).

### 10.325.2.5 baseline

double prcopt\_t::baseline[2]  
Definition at line 986 of file [rtklib.h](#).

### 10.325.2.6 bdsmodear

int prcopt\_t::bdsmodear  
Definition at line 953 of file [rtklib.h](#).

### 10.325.2.7 codesmooth

int prcopt\_t::codesmooth  
Definition at line 963 of file [rtklib.h](#).

### 10.325.2.8 dynamics

int prcopt\_t::dynamics  
Definition at line 960 of file [rtklib.h](#).

### 10.325.2.9 elmaskar

double prcopt\_t::elmaskar  
Definition at line 980 of file [rtklib.h](#).

### 10.325.2.10 elmaskhold

double prcopt\_t::elmaskhold  
Definition at line 981 of file [rtklib.h](#).

### 10.325.2.11 elmin

double prcopt\_t::elmin  
Definition at line 948 of file [rtklib.h](#).

### 10.325.2.12 eratio

double prcopt\_t::eratio[NFREQ]  
Definition at line 971 of file [rtklib.h](#).



**10.325.2.13 err**

```
double prcopt_t::err[5]
```

Definition at line 972 of file [rtklib.h](#).

**10.325.2.14 exsats**

```
unsigned char prcopt_t::exsats[MAXSAT]
```

Definition at line 992 of file [rtklib.h](#).

**10.325.2.15 exterr**

```
exterr_t prcopt_t::exterr
```

Definition at line 1000 of file [rtklib.h](#).

**10.325.2.16 freqopt**

```
int prcopt_t::freqopt
```

Definition at line 1001 of file [rtklib.h](#).

**10.325.2.17 glomodear**

```
int prcopt_t::glomodear
```

Definition at line 952 of file [rtklib.h](#).

**10.325.2.18 initrst**

```
int prcopt_t::initrst
```

Definition at line 994 of file [rtklib.h](#).

**10.325.2.19 intpref**

```
int prcopt_t::intpref
```

Definition at line 964 of file [rtklib.h](#).

**10.325.2.20 ionoopt**

```
int prcopt_t::ionoopt
```

Definition at line 958 of file [rtklib.h](#).

**10.325.2.21 maxaveep**

```
int prcopt_t::maxaveep
```

Definition at line 993 of file [rtklib.h](#).

**10.325.2.22 maxgdop**

```
double prcopt_t::maxgdop
```

Definition at line 985 of file [rtklib.h](#).

**10.325.2.23 maxinno**

```
double prcopt_t::maxinno
```

Definition at line 984 of file [rtklib.h](#).

**10.325.2.24 maxout**

```
int prcopt_t::maxout
```

Definition at line 954 of file [rtklib.h](#).

**10.325.2.25 maxtdiff**

```
double prcopt_t::maxtdiff
```

Definition at line 983 of file [rtklib.h](#).

**10.325.2.26 minfix**

```
int prcopt_t::minfix
```

Definition at line 956 of file [rtklib.h](#).

**10.325.2.27 minlock**

```
int prcopt_t::minlock
```

Definition at line 955 of file [rtklib.h](#).

**10.325.2.28 mode**

```
int prcopt_t::mode
```

Definition at line 944 of file [rtklib.h](#).

**10.325.2.29 modear**

```
int prcopt_t::modear
```

Definition at line 951 of file [rtklib.h](#).

**10.325.2.30 navsys**

```
int prcopt_t::navsys
```

Definition at line 947 of file [rtklib.h](#).

**10.325.2.31 nf**

```
int prcopt_t::nf
```

Definition at line 946 of file [rtklib.h](#).

**10.325.2.32 niter**

```
int prcopt_t::niter
```

Definition at line 962 of file [rtklib.h](#).

**10.325.2.33 odisp**

```
double prcopt_t::odisp[2][6 * 11]
```

Definition at line 999 of file [rtklib.h](#).

**10.325.2.34 outsingle**

```
int prcopt_t::outsingle
```

Definition at line 995 of file [rtklib.h](#).

**10.325.2.35 pcvr**

```
pcv_t prcopt_t::pcvr[2]
```

Definition at line 991 of file [rtklib.h](#).

**10.325.2.36 posopt**

```
int prcopt_t::posopt[6]
```

Definition at line 997 of file [rtklib.h](#).

**10.325.2.37 pppopt**

char prcopt\_t::pppopt[256]  
Definition at line 1002 of file [rtklib.h](#).

**10.325.2.38 prn**

double prcopt\_t::prn[6]  
Definition at line 977 of file [rtklib.h](#).

**10.325.2.39 rb**

double prcopt\_t::rb[3]  
Definition at line 988 of file [rtklib.h](#).

**10.325.2.40 refpos**

int prcopt\_t::refpos  
Definition at line 968 of file [rtklib.h](#).

**10.325.2.41 rnxopt**

char prcopt\_t::rnxopt[2][256]  
Definition at line 996 of file [rtklib.h](#).

**10.325.2.42 rovpos**

int prcopt\_t::rovpos  
Definition at line 967 of file [rtklib.h](#).

**10.325.2.43 ru**

double prcopt\_t::ru[3]  
Definition at line 987 of file [rtklib.h](#).

**10.325.2.44 sateph**

int prcopt\_t::sateph  
Definition at line 950 of file [rtklib.h](#).

**10.325.2.45 sbascorr**

int prcopt\_t::sbascorr  
Definition at line 965 of file [rtklib.h](#).

**10.325.2.46 sbassatsel**

int prcopt\_t::sbassatsel  
Definition at line 966 of file [rtklib.h](#).

**10.325.2.47 sclkstab**

double prcopt\_t::sclkstab  
Definition at line 978 of file [rtklib.h](#).

**10.325.2.48 snrmask**

[snrmask\\_t](#) prcopt\_t::snrmask  
Definition at line 949 of file [rtklib.h](#).

**10.325.2.49 soltype**

```
int prcopt_t::soltype
```

Definition at line 945 of file [rtklib.h](#).

**10.325.2.50 std**

```
double prcopt_t::std[3]
```

Definition at line 976 of file [rtklib.h](#).

**10.325.2.51 syncsol**

```
int prcopt_t::syncsol
```

Definition at line 998 of file [rtklib.h](#).

**10.325.2.52 thresar**

```
double prcopt_t::thresar[8]
```

Definition at line 979 of file [rtklib.h](#).

**10.325.2.53 thresslip**

```
double prcopt_t::thresslip
```

Definition at line 982 of file [rtklib.h](#).

**10.325.2.54 tidecorr**

```
int prcopt_t::tidecorr
```

Definition at line 961 of file [rtklib.h](#).

**10.325.2.55 tropopt**

```
int prcopt_t::tropopt
```

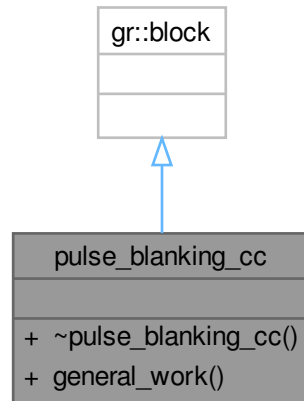
Definition at line 959 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

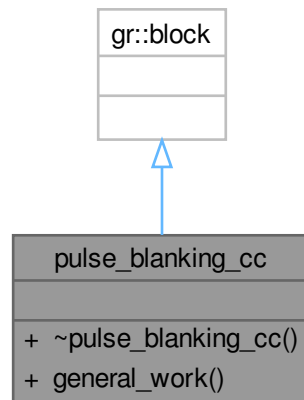
- [rtklib.h](#)

## 10.326 pulse\_blanking\_cc Class Reference

Inheritance diagram for pulse\_blanking\_cc:



Collaboration diagram for pulse\_blanking\_cc:



### Public Member Functions

- `int` **general\_work** (`int` noutput\_items \_\_attribute\_\_((unused)), `gr_vector_int` &ninput\_items \_\_attribute\_\_((unused)), `gr_vector_const_void_star` &input\_items, `gr_vector_void_star` &output\_items)

### Friends

- `pulse_blanking_cc_sptr` **make\_pulse\_blanking\_cc** (`float` pfa, `int32_t` length, `int32_t` n\_segments\_est, `int32_t` n\_segments\_reset)

### 10.326.1 Detailed Description

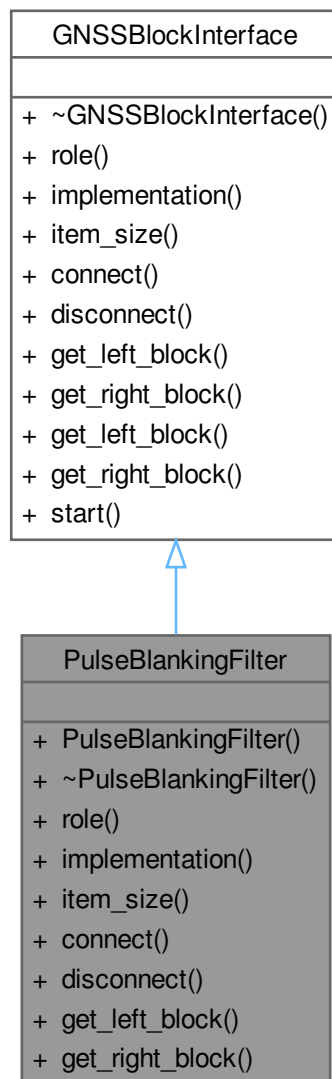
Definition at line 44 of file [pulse\\_blanking\\_cc.h](#).

The documentation for this class was generated from the following file:

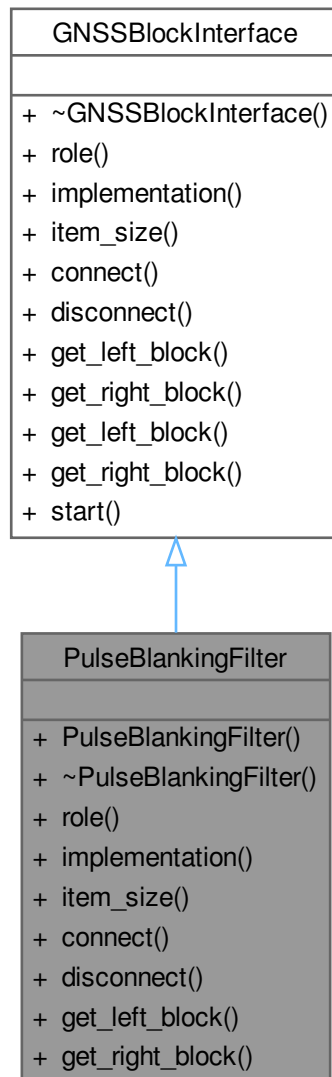
- [pulse\\_blanking\\_cc.h](#)

## 10.327 PulseBlankingFilter Class Reference

Inheritance diagram for PulseBlankingFilter:



Collaboration diagram for PulseBlankingFilter:



### Public Member Functions

- **PulseBlankingFilter** (const [ConfigurationInterface](#) \*configuration, std::string role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "Pulse\_Blanking\_Filter".
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual `gr::basic_block_sptr` [get\\_left\\_block](#) (int RF\_channel)
- virtual `gr::basic_block_sptr` [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.327.1 Detailed Description

Definition at line 39 of file [pulse\\_blanking\\_filter.h](#).

### 10.327.2 Member Function Documentation

#### 10.327.2.1 [connect\(\)](#)

```
void PulseBlankingFilter::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.327.2.2 [disconnect\(\)](#)

```
void PulseBlankingFilter::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.327.2.3 [get\\_left\\_block\(\)](#)

```
gr::basic_block_sptr PulseBlankingFilter::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.327.2.4 [get\\_right\\_block\(\)](#)

```
gr::basic_block_sptr PulseBlankingFilter::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.327.2.5 [implementation\(\)](#)

```
std::string PulseBlankingFilter::implementation () [inline], [override], [virtual]
```

Returns "Pulse\_Blanking\_Filter".

Implements [GNSSBlockInterface](#).

Definition at line 54 of file [pulse\\_blanking\\_filter.h](#).

#### 10.327.2.6 [item\\_size\(\)](#)

```
size_t PulseBlankingFilter::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 59 of file [pulse\\_blanking\\_filter.h](#).

#### 10.327.2.7 [role\(\)](#)

```
std::string PulseBlankingFilter::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 48 of file [pulse\\_blanking\\_filter.h](#).

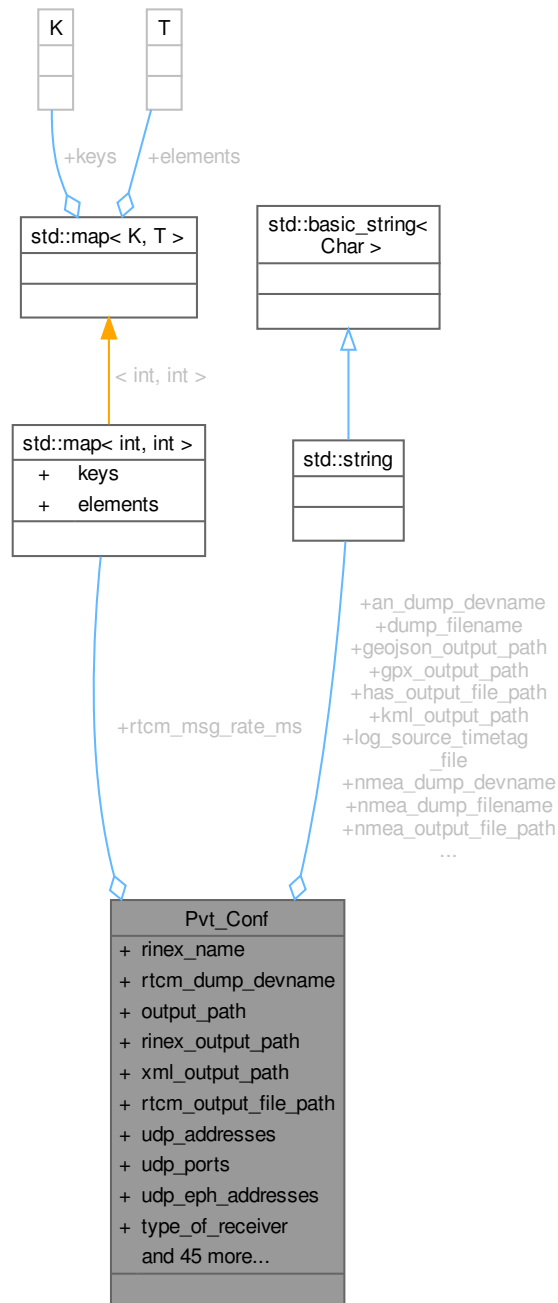
The documentation for this class was generated from the following file:

- [pulse\\_blanking\\_filter.h](#)



## 10.328 Pvt\_Conf Class Reference

Collaboration diagram for Pvt\_Conf:



### Public Attributes

- `std::map< int, int >` `rtcn_msg_rate_ms`
- `std::string` `rinex_name` = `std::string("-")`
- `std::string` `dump_filename`
- `std::string` `nmea_dump_filename`
- `std::string` `nmea_dump_devname`

- `std::string rtm_dump_devname`
- `std::string an_dump_devname`
- `std::string output_path = std::string(".")`
- `std::string rinex_output_path = std::string(".")`
- `std::string gpx_output_path = std::string(".")`
- `std::string geojson_output_path = std::string(".")`
- `std::string nmea_output_file_path = std::string(".")`
- `std::string kml_output_path = std::string(".")`
- `std::string xml_output_path = std::string(".")`
- `std::string rtm_output_file_path = std::string(".")`
- `std::string has_output_file_path = std::string(".")`
- `std::string udp_addresses`
- `std::string udp_ports`
- `std::string udp_eph_addresses`
- `std::string log_source_timetag_file`
- `uint32_t type_of_receiver = 0`
- `uint32_t observable_interval_ms = 20`
- `int32_t output_rate_ms = 0`
- `int32_t display_rate_ms = 0`
- `int32_t kml_rate_ms = 20`
- `int32_t gpx_rate_ms = 20`
- `int32_t geojson_rate_ms = 20`
- `int32_t nmea_rate_ms = 20`
- `int32_t rinex_version = 0`
- `int32_t rinexobs_rate_ms = 0`
- `int32_t an_rate_ms = 20`
- `int32_t max_obs_block_rx_clock_offset_ms = 40`
- `int udp_eph_port = 0`
- `int rtk_trace_level = 0`
- `uint16_t rtm_tcp_port = 0`
- `uint16_t rtm_station_id = 0`
- `bool flag_nmea_tty_port = false`
- `bool flag_rtm_server = false`
- `bool flag_rtm_tty_port = false`
- `bool output_enabled = true`
- `bool rinex_output_enabled = true`
- `bool gpx_output_enabled = true`
- `bool geojson_output_enabled = true`
- `bool nmea_output_file_enabled = true`
- `bool an_output_enabled = false`
- `bool kml_output_enabled = true`
- `bool xml_output_enabled = true`
- `bool rtm_output_file_enabled = true`
- `bool monitor_enabled = false`
- `bool monitor_ephemeris_enabled = false`
- `bool protobuf_enabled = true`
- `bool enable_rx_clock_correction = true`
- `bool show_local_time_zone = false`
- `bool pre_2009_file = false`
- `bool dump = false`
- `bool dump_mat = true`
- `bool log_source_timetag = false`
- `bool use_e6_for_pvt = true`
- `bool use_has_corrections = true`
- `bool use_unhealthy_sats = false`

- bool `osnma_strict` = false
- bool `enable_pvt_kf` = false
- double `measures_ecef_pos_sd_m` = 1.0
- double `measures_ecef_vel_sd_ms` = 0.1
- double `system_ecef_pos_sd_m` = 0.01
- double `system_ecef_vel_sd_ms` = 0.001

### 10.328.1 Detailed Description

Definition at line 30 of file [pvt\\_conf.h](#).

### 10.328.2 Member Data Documentation

#### 10.328.2.1 `an_dump_devname`

```
std::string Pvt_Conf::an_dump_devname
```

Definition at line 40 of file [pvt\\_conf.h](#).

#### 10.328.2.2 `an_output_enabled`

```
bool Pvt_Conf::an_output_enabled = false
```

Definition at line 82 of file [pvt\\_conf.h](#).

#### 10.328.2.3 `an_rate_ms`

```
int32_t Pvt_Conf::an_rate_ms = 20
```

Definition at line 66 of file [pvt\\_conf.h](#).

#### 10.328.2.4 `display_rate_ms`

```
int32_t Pvt_Conf::display_rate_ms = 0
```

Definition at line 59 of file [pvt\\_conf.h](#).

#### 10.328.2.5 `dump`

```
bool Pvt_Conf::dump = false
```

Definition at line 92 of file [pvt\\_conf.h](#).

#### 10.328.2.6 `dump_filename`

```
std::string Pvt_Conf::dump_filename
```

Definition at line 36 of file [pvt\\_conf.h](#).

#### 10.328.2.7 `dump_mat`

```
bool Pvt_Conf::dump_mat = true
```

Definition at line 93 of file [pvt\\_conf.h](#).

#### 10.328.2.8 `enable_pvt_kf`

```
bool Pvt_Conf::enable_pvt_kf = false
```

Definition at line 101 of file [pvt\\_conf.h](#).

#### 10.328.2.9 `enable_rx_clock_correction`

```
bool Pvt_Conf::enable_rx_clock_correction = true
```

Definition at line 89 of file [pvt\\_conf.h](#).

#### 10.328.2.10 flag\_nmea\_tty\_port

bool Pvt\_Conf::flag\_nmea\_tty\_port = false  
Definition at line 74 of file [pvt\\_conf.h](#).

#### 10.328.2.11 flag\_rtcn\_server

bool Pvt\_Conf::flag\_rtcn\_server = false  
Definition at line 75 of file [pvt\\_conf.h](#).

#### 10.328.2.12 flag\_rtcn\_tty\_port

bool Pvt\_Conf::flag\_rtcn\_tty\_port = false  
Definition at line 76 of file [pvt\\_conf.h](#).

#### 10.328.2.13 geojson\_output\_enabled

bool Pvt\_Conf::geojson\_output\_enabled = true  
Definition at line 80 of file [pvt\\_conf.h](#).

#### 10.328.2.14 geojson\_output\_path

std::string Pvt\_Conf::geojson\_output\_path = std::string(".")  
Definition at line 44 of file [pvt\\_conf.h](#).

#### 10.328.2.15 geojson\_rate\_ms

int32\_t Pvt\_Conf::geojson\_rate\_ms = 20  
Definition at line 62 of file [pvt\\_conf.h](#).

#### 10.328.2.16 gpx\_output\_enabled

bool Pvt\_Conf::gpx\_output\_enabled = true  
Definition at line 79 of file [pvt\\_conf.h](#).

#### 10.328.2.17 gpx\_output\_path

std::string Pvt\_Conf::gpx\_output\_path = std::string(".")  
Definition at line 43 of file [pvt\\_conf.h](#).

#### 10.328.2.18 gpx\_rate\_ms

int32\_t Pvt\_Conf::gpx\_rate\_ms = 20  
Definition at line 61 of file [pvt\\_conf.h](#).

#### 10.328.2.19 has\_output\_file\_path

std::string Pvt\_Conf::has\_output\_file\_path = std::string(".")  
Definition at line 49 of file [pvt\\_conf.h](#).

#### 10.328.2.20 kml\_output\_enabled

bool Pvt\_Conf::kml\_output\_enabled = true  
Definition at line 83 of file [pvt\\_conf.h](#).

#### 10.328.2.21 kml\_output\_path

std::string Pvt\_Conf::kml\_output\_path = std::string(".")  
Definition at line 46 of file [pvt\\_conf.h](#).

**10.328.2.22 kml\_rate\_ms**

```
int32_t Pvt_Conf::kml_rate_ms = 20
```

Definition at line 60 of file [pvt\\_conf.h](#).

**10.328.2.23 log\_source\_timetag**

```
bool Pvt_Conf::log_source_timetag = false
```

Definition at line 94 of file [pvt\\_conf.h](#).

**10.328.2.24 log\_source\_timetag\_file**

```
std::string Pvt_Conf::log_source_timetag_file
```

Definition at line 53 of file [pvt\\_conf.h](#).

**10.328.2.25 max\_obs\_block\_rx\_clock\_offset\_ms**

```
int32_t Pvt_Conf::max_obs_block_rx_clock_offset_ms = 40
```

Definition at line 67 of file [pvt\\_conf.h](#).

**10.328.2.26 measures\_ecef\_pos\_sd\_m**

```
double Pvt_Conf::measures_ecef_pos_sd_m = 1.0
```

Definition at line 102 of file [pvt\\_conf.h](#).

**10.328.2.27 measures\_ecef\_vel\_sd\_ms**

```
double Pvt_Conf::measures_ecef_vel_sd_ms = 0.1
```

Definition at line 103 of file [pvt\\_conf.h](#).

**10.328.2.28 monitor\_enabled**

```
bool Pvt_Conf::monitor_enabled = false
```

Definition at line 86 of file [pvt\\_conf.h](#).

**10.328.2.29 monitor\_ephemeris\_enabled**

```
bool Pvt_Conf::monitor_ephemeris_enabled = false
```

Definition at line 87 of file [pvt\\_conf.h](#).

**10.328.2.30 nmea\_dump\_devname**

```
std::string Pvt_Conf::nmea_dump_devname
```

Definition at line 38 of file [pvt\\_conf.h](#).

**10.328.2.31 nmea\_dump\_filename**

```
std::string Pvt_Conf::nmea_dump_filename
```

Definition at line 37 of file [pvt\\_conf.h](#).

**10.328.2.32 nmea\_output\_file\_enabled**

```
bool Pvt_Conf::nmea_output_file_enabled = true
```

Definition at line 81 of file [pvt\\_conf.h](#).

**10.328.2.33 nmea\_output\_file\_path**

```
std::string Pvt_Conf::nmea_output_file_path = std::string(".")
```

Definition at line 45 of file [pvt\\_conf.h](#).

**10.328.2.34 nmea\_rate\_ms**

```
int32_t Pvt_Conf::nmea_rate_ms = 20
```

Definition at line 63 of file [pvt\\_conf.h](#).

**10.328.2.35 observable\_interval\_ms**

```
uint32_t Pvt_Conf::observable_interval_ms = 20
```

Definition at line 56 of file [pvt\\_conf.h](#).

**10.328.2.36 osnma\_strict**

```
bool Pvt_Conf::osnma_strict = false
```

Definition at line 98 of file [pvt\\_conf.h](#).

**10.328.2.37 output\_enabled**

```
bool Pvt_Conf::output_enabled = true
```

Definition at line 77 of file [pvt\\_conf.h](#).

**10.328.2.38 output\_path**

```
std::string Pvt_Conf::output_path = std::string(".")
```

Definition at line 41 of file [pvt\\_conf.h](#).

**10.328.2.39 output\_rate\_ms**

```
int32_t Pvt_Conf::output_rate_ms = 0
```

Definition at line 58 of file [pvt\\_conf.h](#).

**10.328.2.40 pre\_2009\_file**

```
bool Pvt_Conf::pre_2009_file = false
```

Definition at line 91 of file [pvt\\_conf.h](#).

**10.328.2.41 protobuf\_enabled**

```
bool Pvt_Conf::protobuf_enabled = true
```

Definition at line 88 of file [pvt\\_conf.h](#).

**10.328.2.42 rinex\_name**

```
std::string Pvt_Conf::rinex_name = std::string("-")
```

Definition at line 35 of file [pvt\\_conf.h](#).

**10.328.2.43 rinex\_output\_enabled**

```
bool Pvt_Conf::rinex_output_enabled = true
```

Definition at line 78 of file [pvt\\_conf.h](#).

**10.328.2.44 rinex\_output\_path**

```
std::string Pvt_Conf::rinex_output_path = std::string(".")
```

Definition at line 42 of file [pvt\\_conf.h](#).

**10.328.2.45 rinex\_version**

```
int32_t Pvt_Conf::rinex_version = 0
```

Definition at line 64 of file [pvt\\_conf.h](#).

**10.328.2.46 rinexobs\_rate\_ms**

int32\_t Pvt\_Conf::rinexobs\_rate\_ms = 0  
Definition at line 65 of file [pvt\\_conf.h](#).

**10.328.2.47 rtcm\_dump\_devname**

std::string Pvt\_Conf::rtcm\_dump\_devname  
Definition at line 39 of file [pvt\\_conf.h](#).

**10.328.2.48 rtcm\_msg\_rate\_ms**

std::map<int, int> Pvt\_Conf::rtcm\_msg\_rate\_ms  
Definition at line 33 of file [pvt\\_conf.h](#).

**10.328.2.49 rtcm\_output\_file\_enabled**

bool Pvt\_Conf::rtcm\_output\_file\_enabled = true  
Definition at line 85 of file [pvt\\_conf.h](#).

**10.328.2.50 rtcm\_output\_file\_path**

std::string Pvt\_Conf::rtcm\_output\_file\_path = std::string(".")  
Definition at line 48 of file [pvt\\_conf.h](#).

**10.328.2.51 rtcm\_station\_id**

uint16\_t Pvt\_Conf::rtcm\_station\_id = 0  
Definition at line 72 of file [pvt\\_conf.h](#).

**10.328.2.52 rtcm\_tcp\_port**

uint16\_t Pvt\_Conf::rtcm\_tcp\_port = 0  
Definition at line 71 of file [pvt\\_conf.h](#).

**10.328.2.53 rtk\_trace\_level**

int Pvt\_Conf::rtk\_trace\_level = 0  
Definition at line 69 of file [pvt\\_conf.h](#).

**10.328.2.54 show\_local\_time\_zone**

bool Pvt\_Conf::show\_local\_time\_zone = false  
Definition at line 90 of file [pvt\\_conf.h](#).

**10.328.2.55 system\_ecef\_pos\_sd\_m**

double Pvt\_Conf::system\_ecef\_pos\_sd\_m = 0.01  
Definition at line 104 of file [pvt\\_conf.h](#).

**10.328.2.56 system\_ecef\_vel\_sd\_ms**

double Pvt\_Conf::system\_ecef\_vel\_sd\_ms = 0.001  
Definition at line 105 of file [pvt\\_conf.h](#).

**10.328.2.57 type\_of\_receiver**

uint32\_t Pvt\_Conf::type\_of\_receiver = 0  
Definition at line 55 of file [pvt\\_conf.h](#).

#### 10.328.2.58 udp\_addresses

`std::string Pvt_Conf::udp_addresses`  
Definition at line 50 of file [pvt\\_conf.h](#).

#### 10.328.2.59 udp\_eph\_addresses

`std::string Pvt_Conf::udp_eph_addresses`  
Definition at line 52 of file [pvt\\_conf.h](#).

#### 10.328.2.60 udp\_eph\_port

`int Pvt_Conf::udp_eph_port = 0`  
Definition at line 68 of file [pvt\\_conf.h](#).

#### 10.328.2.61 udp\_ports

`std::string Pvt_Conf::udp_ports`  
Definition at line 51 of file [pvt\\_conf.h](#).

#### 10.328.2.62 use\_e6\_for\_pvt

`bool Pvt_Conf::use_e6_for_pvt = true`  
Definition at line 95 of file [pvt\\_conf.h](#).

#### 10.328.2.63 use\_has\_corrections

`bool Pvt_Conf::use_has_corrections = true`  
Definition at line 96 of file [pvt\\_conf.h](#).

#### 10.328.2.64 use\_unhealthy\_sats

`bool Pvt_Conf::use_unhealthy_sats = false`  
Definition at line 97 of file [pvt\\_conf.h](#).

#### 10.328.2.65 xml\_output\_enabled

`bool Pvt_Conf::xml_output_enabled = true`  
Definition at line 84 of file [pvt\\_conf.h](#).

#### 10.328.2.66 xml\_output\_path

`std::string Pvt_Conf::xml_output_path = std::string(".")`  
Definition at line 47 of file [pvt\\_conf.h](#).

The documentation for this class was generated from the following file:

- [pvt\\_conf.h](#)

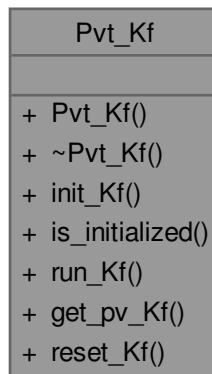
## 10.329 Pvt\_Kf Class Reference

Kalman Filter for Position and Velocity.

```
#include <pvt_kf.h>
```



Collaboration diagram for Pvt\_Kf:



### Public Member Functions

- void **init\_Kf** (const arma::vec &p, const arma::vec &v, double update\_interval\_s, double measures\_ecef\_pos\_sd\_m, double measures\_ecef\_vel\_sd\_ms, double system\_ecef\_pos\_sd\_m, double system\_ecef\_vel\_sd\_ms)
- bool **is\_initialized** () const
- void **run\_Kf** (const arma::vec &p, const arma::vec &v)
- void **get\_pv\_Kf** (arma::vec &p, arma::vec &v) const
- void **reset\_Kf** ()

### 10.329.1 Detailed Description

Kalman Filter for Position and Velocity.

Definition at line 33 of file [pvt\\_kf.h](#).

The documentation for this class was generated from the following file:

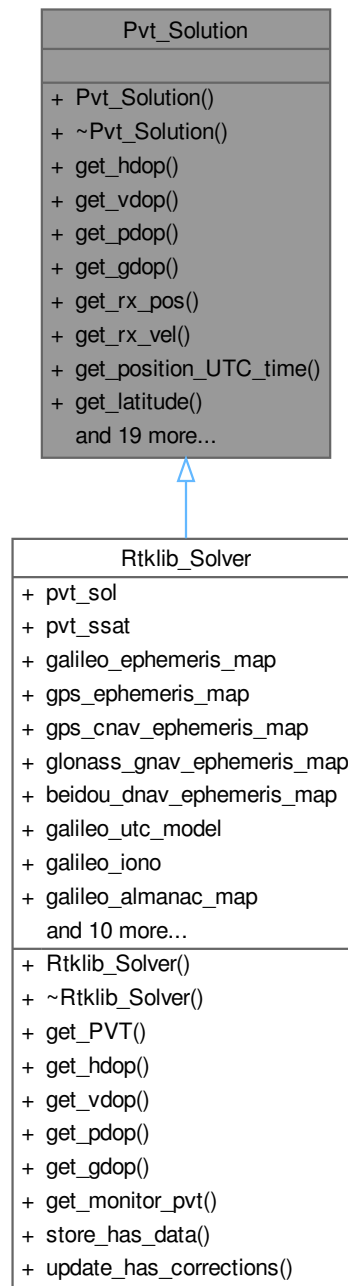
- [pvt\\_kf.h](#)

## 10.330 Pvt\_Solution Class Reference

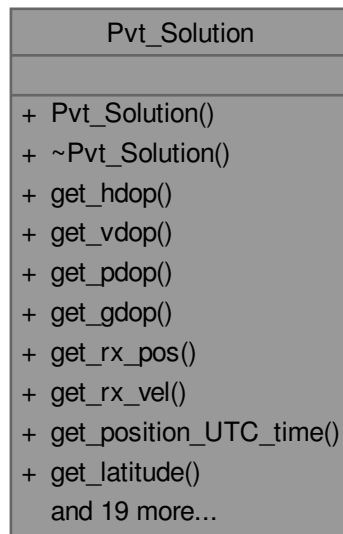
Base class for a PVT solution.

```
#include <pvt_solution.h>
```

Inheritance diagram for Pvt\_Solution:



Collaboration diagram for Pvt\_Solution:



### Public Member Functions

- virtual double **get\_hdop** () const =0
- virtual double **get\_vdop** () const =0
- virtual double **get\_pdop** () const =0
- virtual double **get\_gdop** () const =0
- std::array< double, 3 > **get\_rx\_pos** () const
- std::array< double, 3 > **get\_rx\_vel** () const
- boost::posix\_time::ptime **get\_position\_UTC\_time** () const
- double **get\_latitude** () const  
*Get RX position Latitude WGS84 [deg].*
- double **get\_longitude** () const  
*Get RX position Longitude WGS84 [deg].*
- double **get\_height** () const  
*Get RX position height WGS84 [m].*
- double **get\_time\_offset\_s** () const  
*Get RX time offset [s].*
- double **get\_clock\_drift\_ppm** () const  
*Get the Rx clock drift [ppm].*
- double **get\_speed\_over\_ground** () const  
*Get RX speed over ground [m/s].*
- double **get\_course\_over\_ground** () const  
*Get RX course over ground [deg].*
- int **get\_num\_valid\_observations** () const  
*Get the number of valid pseudorange observations (valid satellites)*
- bool **is\_pre\_2009** () const
- bool **is\_valid\_position** () const
- void **set\_rx\_pos** (const std::array< double, 3 > &pos)

- Set position: X, Y, Z in Cartesian ECEF coordinates [m].*
- void `set_rx_vel` (const std::array< double, 3 > &vel)
- Set velocity: East [m/s], North [m/s], Up [m/s].*
- void `set_position_UTC_time` (const boost::posix\_time::ptime &pt)
- void `set_time_offset_s` (double offset)
- Set RX time offset [s].*
- void `set_clock_drift_ppm` (double clock\_drift\_ppm)
- Set the Rx clock drift [ppm].*
- void `set_speed_over_ground` (double speed\_m\_s)
- Set RX speed over ground [m/s].*
- void `set_course_over_ground` (double cog\_deg)
- Set RX course over ground [deg].*
- void `set_valid_position` (bool is\_valid)
- void `set_num_valid_observations` (int num)
- Set the number of valid pseudorange observations (valid satellites)*
- void `set_pre_2009_file` (bool pre\_2009\_file)
- Flag for the week rollover computation in post processing mode for signals older than 2009.*

### 10.330.1 Detailed Description

Base class for a PVT solution.

Definition at line 34 of file [pvt\\_solution.h](#).

### 10.330.2 Member Function Documentation

#### 10.330.2.1 `get_clock_drift_ppm()`

```
double Pvt_Solution::get_clock_drift_ppm () const
```

Get the Rx clock drift [ppm].

#### 10.330.2.2 `get_course_over_ground()`

```
double Pvt_Solution::get_course_over_ground () const
```

Get RX course over ground [deg].

#### 10.330.2.3 `get_height()`

```
double Pvt_Solution::get_height () const
```

Get RX position height WGS84 [m].

#### 10.330.2.4 `get_latitude()`

```
double Pvt_Solution::get_latitude () const
```

Get RX position Latitude WGS84 [deg].

#### 10.330.2.5 `get_longitude()`

```
double Pvt_Solution::get_longitude () const
```

Get RX position Longitude WGS84 [deg].

#### 10.330.2.6 `get_num_valid_observations()`

```
int Pvt_Solution::get_num_valid_observations () const
```

Get the number of valid pseudorange observations (valid satellites)

#### 10.330.2.7 get\_speed\_over\_ground()

double Pvt\_Solution::get\_speed\_over\_ground () const  
Get RX speed over ground [m/s].

#### 10.330.2.8 get\_time\_offset\_s()

double Pvt\_Solution::get\_time\_offset\_s () const  
Get RX time offset [s].

#### 10.330.2.9 set\_clock\_drift\_ppm()

void Pvt\_Solution::set\_clock\_drift\_ppm (  
    double clock\_drift\_ppm)  
Set the Rx clock drift [ppm].

#### 10.330.2.10 set\_course\_over\_ground()

void Pvt\_Solution::set\_course\_over\_ground (  
    double cog\_deg)  
Set RX course over ground [deg].

#### 10.330.2.11 set\_num\_valid\_observations()

void Pvt\_Solution::set\_num\_valid\_observations (  
    int num)  
Set the number of valid pseudorange observations (valid satellites)

#### 10.330.2.12 set\_pre\_2009\_file()

void Pvt\_Solution::set\_pre\_2009\_file (  
    bool pre\_2009\_file)  
Flag for the week rollover computation in post processing mode for signals older than 2009.

#### 10.330.2.13 set\_rx\_pos()

void Pvt\_Solution::set\_rx\_pos (  
    const std::array< double, 3 > & pos)  
Set position: X, Y, Z in Cartesian ECEF coordinates [m].

#### 10.330.2.14 set\_rx\_vel()

void Pvt\_Solution::set\_rx\_vel (  
    const std::array< double, 3 > & vel)  
Set velocity: East [m/s], North [m/s], Up [m/s].

#### 10.330.2.15 set\_speed\_over\_ground()

void Pvt\_Solution::set\_speed\_over\_ground (  
    double speed\_m\_s)  
Set RX speed over ground [m/s].

#### 10.330.2.16 set\_time\_offset\_s()

void Pvt\_Solution::set\_time\_offset\_s (  
    double offset)  
Set RX time offset [s].

The documentation for this class was generated from the following file:

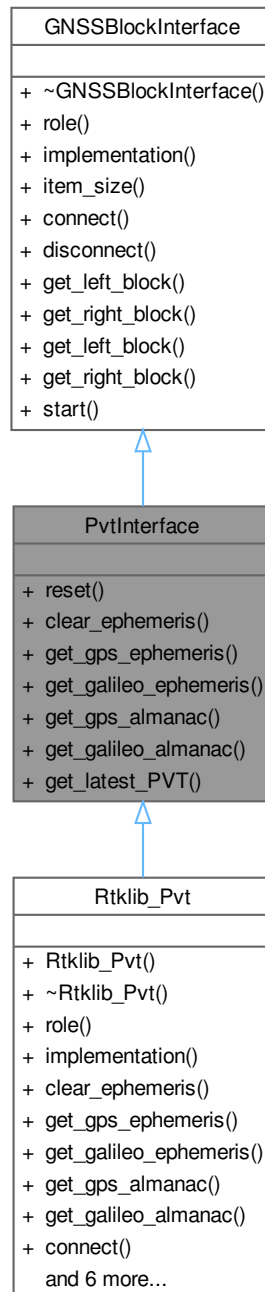
- [pvt\\_solution.h](#)

## 10.331 PvtInterface Class Reference

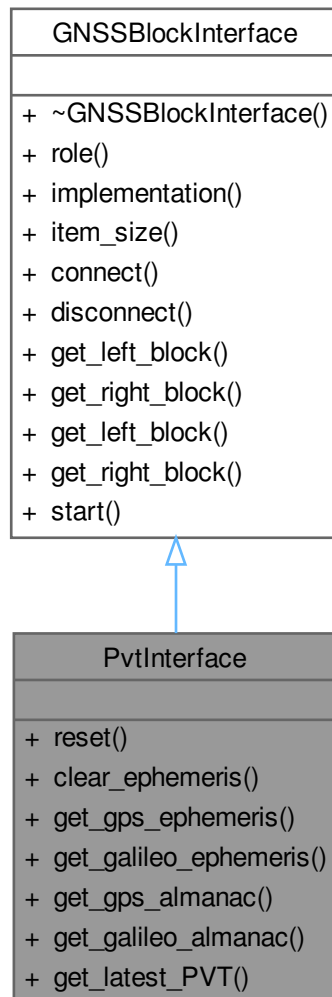
This class represents an interface to a PVT block.

```
#include <pvt_interface.h>
```

Inheritance diagram for PvtInterface:



Collaboration diagram for PvtInterface:



### Public Member Functions

- virtual void **reset** ()=0
- virtual void **clear\_ephemeris** ()=0
- virtual std::map< int, [Gps\\_Ephemeris](#) > **get\_gps\_ephemeris** () const =0
- virtual std::map< int, [Galileo\\_Ephemeris](#) > **get\_galileo\_ephemeris** () const =0
- virtual std::map< int, [Gps\\_Almanac](#) > **get\_gps\_almanac** () const =0
- virtual std::map< int, [Galileo\\_Almanac](#) > **get\_galileo\_almanac** () const =0
- virtual bool **get\_latest\_PVT** (double \*longitude\_deg, double \*latitude\_deg, double \*height\_m, double \*ground\_speed\_kmh, double \*course\_over\_ground\_deg, time\_t \*UTC\_time)=0

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual std::string [role](#) ()=0
- virtual std::string [implementation](#) ()=0
- virtual size\_t [item\\_size](#) ()=0

- virtual void [connect](#) (gr::top\_block\_sptr top\_block)=0
- virtual void [disconnect](#) (gr::top\_block\_sptr top\_block)=0
- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) ()=0
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) ()=0
- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.331.1 Detailed Description

This class represents an interface to a PVT block.

Abstract class for PVT interfaces, derived from [GNSSBlockInterface](#). Since all its methods are virtual, this class cannot be instantiated directly, and a subclass can only be instantiated directly if all inherited pure virtual methods have been implemented by that class or a parent class.

Definition at line 48 of file [pvt\\_interface.h](#).

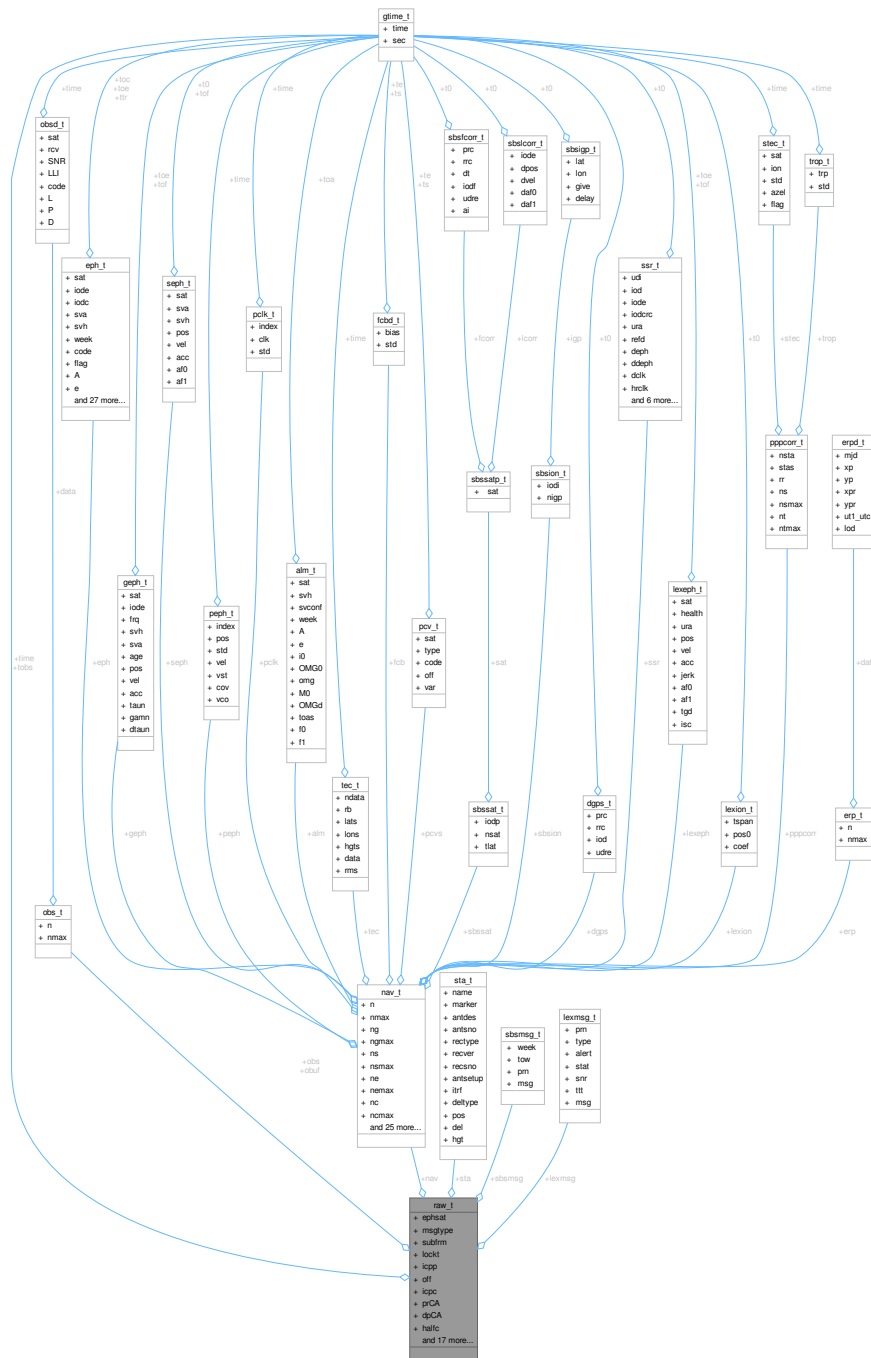
The documentation for this class was generated from the following file:

- [pvt\\_interface.h](#)



## 10.332 raw\_t Struct Reference

Collaboration diagram for raw\_t:



### Public Attributes

- [gtime\\_t](#) time
- [gtime\\_t](#) tobs
- [obs\\_t](#) obs
- [obs\\_t](#) obuf
- [nav\\_t](#) nav
- [sta\\_t](#) sta

- int [ephsat](#)
- [sbsmsg\\_t](#) [sbsmsg](#)
- char [msgtype](#) [256]
- unsigned char [subfrm](#) [MAXSAT][380]
- [lexmsg\\_t](#) [lexmsg](#)
- double [lockt](#) [MAXSAT][NFREQ+NEXOBS]
- double [icpp](#) [MAXSAT]
- double [off](#) [MAXSAT]
- double [icpc](#)
- double [prCA](#) [MAXSAT]
- double [dpCA](#) [MAXSAT]
- unsigned char [halfc](#) [MAXSAT][NFREQ+NEXOBS]
- char [freqn](#) [MAXOBS]
- int [nbyte](#)
- int [len](#)
- int [iod](#)
- int [tod](#)
- int [tbase](#)
- int [flag](#)
- int [outtype](#)
- unsigned char [buff](#) [MAXRAWLEN]
- char [opt](#) [256]
- double [receive\\_time](#)
- unsigned int [plen](#)
- unsigned int [pbyte](#)
- unsigned int [page](#)
- unsigned int [reply](#)
- int [week](#)
- unsigned char [pbuff](#) [255+4+2]

### 10.332.1 Detailed Description

Definition at line [1201](#) of file [rtklib.h](#).

### 10.332.2 Member Data Documentation

#### 10.332.2.1 buff

```
unsigned char raw_t::buff[MAXRAWLEN]
```

Definition at line [1226](#) of file [rtklib.h](#).

#### 10.332.2.2 dpCA

```
double raw_t::dpCA[MAXSAT]
```

Definition at line [1216](#) of file [rtklib.h](#).

#### 10.332.2.3 ephsat

```
int raw_t::ephsat
```

Definition at line [1209](#) of file [rtklib.h](#).

#### 10.332.2.4 flag

```
int raw_t::flag
```

Definition at line [1224](#) of file [rtklib.h](#).

#### 10.332.2.5 freqn

`char raw_t::freqn[MAXOBS]`  
Definition at line [1218](#) of file [rtklib.h](#).

#### 10.332.2.6 halfc

`unsigned char raw_t::halfc[MAXSAT][NFREQ+NEXOBS]`  
Definition at line [1217](#) of file [rtklib.h](#).

#### 10.332.2.7 icpc

`double raw_t::icpc`  
Definition at line [1215](#) of file [rtklib.h](#).

#### 10.332.2.8 icpp

`double raw_t::icpp[MAXSAT]`  
Definition at line [1215](#) of file [rtklib.h](#).

#### 10.332.2.9 iod

`int raw_t::iod`  
Definition at line [1221](#) of file [rtklib.h](#).

#### 10.332.2.10 len

`int raw_t::len`  
Definition at line [1220](#) of file [rtklib.h](#).

#### 10.332.2.11 lexmsg

`lexmsg\_t raw_t::lexmsg`  
Definition at line [1213](#) of file [rtklib.h](#).

#### 10.332.2.12 lockt

`double raw_t::lockt[MAXSAT][NFREQ+NEXOBS]`  
Definition at line [1214](#) of file [rtklib.h](#).

#### 10.332.2.13 msgtype

`char raw_t::msgtype[256]`  
Definition at line [1211](#) of file [rtklib.h](#).

#### 10.332.2.14 nav

`nav\_t raw_t::nav`  
Definition at line [1207](#) of file [rtklib.h](#).

#### 10.332.2.15 nbyte

`int raw_t::nbyte`  
Definition at line [1219](#) of file [rtklib.h](#).

#### 10.332.2.16 obs

`obs\_t raw_t::obs`  
Definition at line [1205](#) of file [rtklib.h](#).

**10.332.2.17 obuf**

```
obs_t raw_t::obuf
```

Definition at line 1206 of file [rtklib.h](#).

**10.332.2.18 off**

```
double raw_t::off[MAXSAT]
```

Definition at line 1215 of file [rtklib.h](#).

**10.332.2.19 opt**

```
char raw_t::opt[256]
```

Definition at line 1227 of file [rtklib.h](#).

**10.332.2.20 outtype**

```
int raw_t::outtype
```

Definition at line 1225 of file [rtklib.h](#).

**10.332.2.21 page**

```
unsigned int raw_t::page
```

Definition at line 1231 of file [rtklib.h](#).

**10.332.2.22 pbuff**

```
unsigned char raw_t::pbuff[255+4+2]
```

Definition at line 1234 of file [rtklib.h](#).

**10.332.2.23 pbyte**

```
unsigned int raw_t::pbyte
```

Definition at line 1230 of file [rtklib.h](#).

**10.332.2.24 plen**

```
unsigned int raw_t::plen
```

Definition at line 1229 of file [rtklib.h](#).

**10.332.2.25 prCA**

```
double raw_t::prCA[MAXSAT]
```

Definition at line 1216 of file [rtklib.h](#).

**10.332.2.26 receive\_time**

```
double raw_t::receive_time
```

Definition at line 1228 of file [rtklib.h](#).

**10.332.2.27 reply**

```
unsigned int raw_t::reply
```

Definition at line 1232 of file [rtklib.h](#).

**10.332.2.28 sbsmsg**

```
sbsmsg_t raw_t::sbsmsg
```

Definition at line 1210 of file [rtklib.h](#).

**10.332.2.29 sta**

```
sta_t raw_t::sta
```

Definition at line 1208 of file [rtklib.h](#).

**10.332.2.30 subfrm**

```
unsigned char raw_t::subfrm[MAXSAT][380]
```

Definition at line 1212 of file [rtklib.h](#).

**10.332.2.31 tbase**

```
int raw_t::tbase
```

Definition at line 1223 of file [rtklib.h](#).

**10.332.2.32 time**

```
gtime_t raw_t::time
```

Definition at line 1203 of file [rtklib.h](#).

**10.332.2.33 tobs**

```
gtime_t raw_t::tobs
```

Definition at line 1204 of file [rtklib.h](#).

**10.332.2.34 tod**

```
int raw_t::tod
```

Definition at line 1222 of file [rtklib.h](#).

**10.332.2.35 week**

```
int raw_t::week
```

Definition at line 1233 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

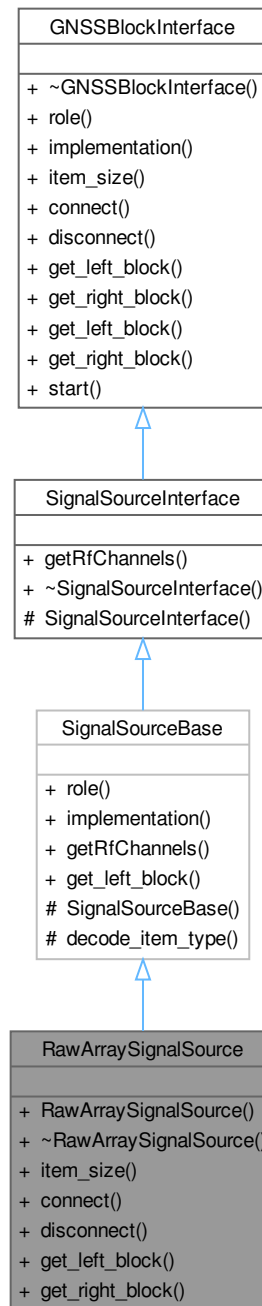
- [rtklib.h](#)

## 10.333 RawArraySignalSource Class Reference

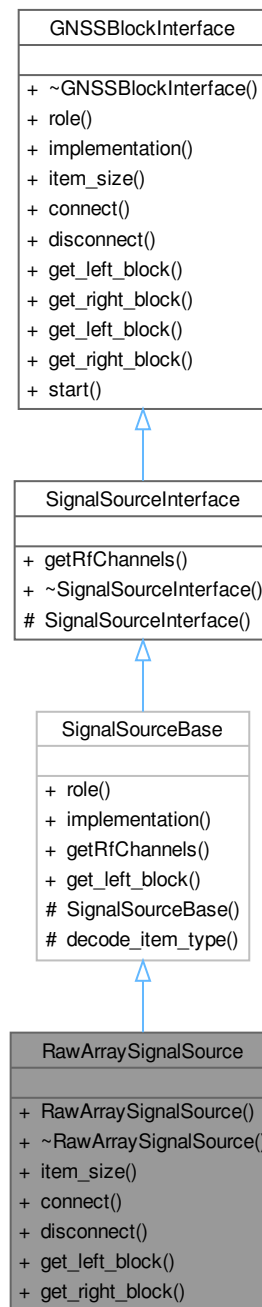
This class reads samples from an antenna array RF front-end signal sampler.

```
#include <raw_array_signal_source.h>
```

Inheritance diagram for RawArraySignalSource:



Collaboration diagram for RawArraySignalSource:



## Public Member Functions

- **RawArraySignalSource** (const [ConfigurationInterface](#) \*configuration, std::string role, unsigned int in\_stream, unsigned int out\_stream, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override

## Public Member Functions inherited from [SignalSourceBase](#)

- `std::string` [role](#) () final
- `std::string` [implementation](#) () final
- `size_t` [getRfChannels](#) () const override
- `gr::basic_block_sptr` [get\\_left\\_block](#) () override

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual `gr::basic_block_sptr` [get\\_left\\_block](#) (int RF\_channel)
- virtual `gr::basic_block_sptr` [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

## Additional Inherited Members

## Protected Member Functions inherited from [SignalSourceBase](#)

- [SignalSourceBase](#) ([ConfigurationInterface](#) const \*configuration, `std::string` role, `std::string` impl)  
*Constructor.*
- `size_t` [decode\\_item\\_type](#) (`std::string` const &item\_type, bool \*is\_interleaved=nullptr, bool throw\_on\_↵ error=false)  
*utility for decoding passed ".item\_type" values*

### 10.333.1 Detailed Description

This class reads samples from an antenna array RF front-end signal sampler.  
Definition at line 42 of file [raw\\_array\\_signal\\_source.h](#).

### 10.333.2 Member Function Documentation

#### 10.333.2.1 connect()

```
void RawArraySignalSource::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.333.2.2 disconnect()

```
void RawArraySignalSource::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.333.2.3 get\_left\_block()

```
gr::basic_block_sptr RawArraySignalSource::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.333.2.4 get\_right\_block()

```
gr::basic_block_sptr RawArraySignalSource::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.333.2.5 item\_size()

```
size_t RawArraySignalSource::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 51 of file [raw\\_array\\_signal\\_source.h](#).

The documentation for this class was generated from the following file:

- [raw\\_array\\_signal\\_source.h](#)

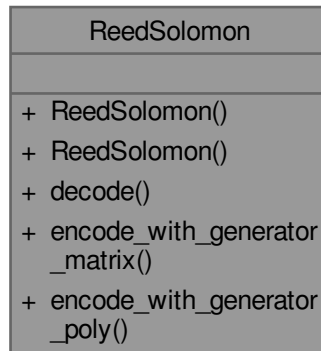


## 10.334 ReedSolomon Class Reference

Class implementing a Reed-Solomon encoder and decoder RS(255,K,d) where  $k=255-nroots$  is the information vector length and  $d=nroots+1$  is the minimum Hamming distance, with symbols of 8 bits. It allows shortened RS codes.

```
#include <reed_solomon.h>
```

Collaboration diagram for ReedSolomon:



### Public Member Functions

- [ReedSolomon](#) (const std::string &gnss\_signal=std::string())  
*Default constructor. Constructs a Reed Solomon object. The `encode_with_generator_poly` and `encode_with_generator_matrix` methods are available for testing purposes.*
- [ReedSolomon](#) (int nroots, int minpoly, int prim, int fcr, int pad=0, int shortening=0, const std::vector< uint8\_t > &genpoly\_coeff=std::vector< uint8\_t >{}, const std::vector< std::vector< uint8\_t > > &gen\_matrix=std::vector< std::vector< uint8\_t > >{}))  
*Custom constructor for RS(255, 255-nroots, nroots+1). Parameters:*
- int [decode](#) (std::vector< uint8\_t > &data\_to\_decode, const std::vector< int > &erasure\_positions=std::vector< int >{}) const  
*Decode an encoded block.*
- std::vector< uint8\_t > [encode\\_with\\_generator\\_matrix](#) (const std::vector< uint8\_t > &data\_to\_encode) const  
*Encode data with the generator matrix (for testing purposes)*
- std::vector< uint8\_t > [encode\\_with\\_generator\\_poly](#) (const std::vector< uint8\_t > &data\_to\_encode) const  
*Encode data with the generator polynomial (for testing purposes)*

### 10.334.1 Detailed Description

Class implementing a Reed-Solomon encoder and decoder RS(255,K,d) where  $k=255-nroots$  is the information vector length and  $d=nroots+1$  is the minimum Hamming distance, with symbols of 8 bits. It allows shortened RS codes.

Definition at line 40 of file [reed\\_solomon.h](#).

### 10.334.2 Constructor & Destructor Documentation

#### 10.334.2.1 ReedSolomon() [1/2]

```
ReedSolomon::ReedSolomon (
    const std::string & gnss_signal = std::string()) [explicit]
```

Default constructor. Constructs a Reed Solomon object. The `encode_with_generator_poly` and `encode_with_generator_matrix` methods are available for testing purposes.

`gnss_signal`: empty or "E6B" sets the Galileo E6B RS parameters. "E1B" sets the Galileo E1B (INAV) RS parameters.

### 10.334.2.2 ReedSolomon() [2/2]

```
ReedSolomon::ReedSolomon (
    int nroots,
    int minpoly,
    int prim,
    int fcr,
    int pad = 0,
    int shortening = 0,
    const std::vector< uint8_t > & genpoly_coeff = std::vector< uint8_t > {},
    const std::vector< std::vector< uint8_t > > & gen_matrix = std::vector< std::vector< uint8_t > > {} )
```

Custom constructor for RS(255, 255-nroots, nroots+1). Parameters:

`nroots` - the number of roots in the RS code generator polynomial, which is the same as the number of parity symbols in a block.

`minpoly` - primitive polynomial.

`prim` - the primitive root of the generator polynomial.

`fcr` - first consecutive root of the Reed-Solomon generator polynomial.

`pad` - the number of pad symbols in a block. If not defined, it defaults to 0.

`shortening` - value of the shortening parameter. Defaults to 0. If different to 0, it defines a shortened RS encoder/decoder.

`genpoly_coeff` - a vector of (nroots+1) elements containing the generator polynomial coefficients. Only used for encoding. Defaults to empty. If defined, the `encode_with_generator_poly` method can be used.

`gen_matrix` - a (255-shortening)x(255-nroots-shortening) matrix containing the elements of the generator matrix. Only used for encoding. Defaults to empty. If defined, the `encode_with_generator_matrix` method can be used.

## 10.334.3 Member Function Documentation

### 10.334.3.1 decode()

```
int ReedSolomon::decode (
    std::vector< uint8_t > & data_to_decode,
    const std::vector< int > & erasure_positions = std::vector< int > {} ) const
```

Decode an encoded block.

The decoded symbols are at the first 255-nroots-shortening elements of the `data_to_decode` vector.

The second parameter is optional, and contains a vector of erasure positions to be passed to the decoding algorithm. Defaults to empty.

Returns the number of corrected errors, or -1 if decoding failed.

### 10.334.3.2 encode\_with\_generator\_matrix()

```
std::vector< uint8_t > ReedSolomon::encode_with_generator_matrix (
    const std::vector< uint8_t > & data_to_encode) const
```

Encode data with the generator matrix (for testing purposes)

Returns the encoded vector. It is set to all zeros if the generator matrix is not defined.

### 10.334.3.3 encode\_with\_generator\_poly()

```
std::vector< uint8_t > ReedSolomon::encode_with_generator_poly (
    const std::vector< uint8_t > & data_to_encode) const
```

Encode data with the generator polynomial (for testing purposes)

Returns the encoded vector. It is set to all zeros if the generator polynomial is not defined.

The documentation for this class was generated from the following file:

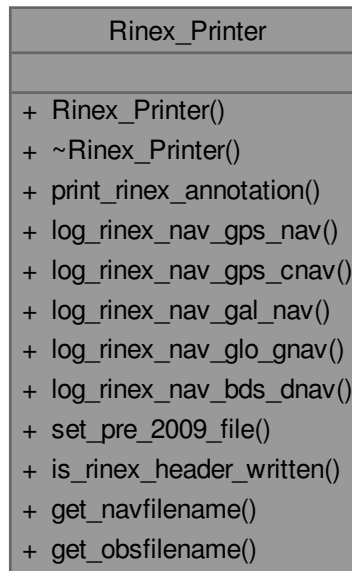
- [reed\\_solomon.h](#)

## 10.335 Rinex\_Printer Class Reference

Class that handles the generation of Receiver INdependent EXchange format (RINEX) files.

```
#include <rinex_printer.h>
```

Collaboration diagram for Rinex\_Printer:



### Public Member Functions

- [Rinex\\_Printer](#) (int version=0, const std::string &base\_path=".", const std::string &base\_name="-")  
*Constructor. Creates GNSS Navigation and Observables RINEX files.*
- [~Rinex\\_Printer](#) ()  
*Destructor. Removes created files if empty.*
- void [print\\_rinex\\_annotation](#) (const [Rtklib\\_Solver](#) \*pvt\_solver, const std::map< int, [Gnss\\_Synchro](#) > &gnss↔  
 \_observables\_map, double rx\_time, int type\_of\_rx, bool flag\_write\_RINEX\_obs\_output)  
*Print RINEX annotation. If it is the first annotation, it also prints the RINEX headers for navigation and observation files. If it is not the first annotation, it only annotates the observation, and updates the navigation header if UTC data was not available when writing it for the first time. The meaning of type\_of\_rx is as follows:*
- void [log\\_rinex\\_nav\\_gps\\_nav](#) (int type\_of\_rx, const std::map< int32\_t, [Gps\\_Ephemeris](#) > &new\_eph)  
*Print RINEX annotation for GPS NAV message.*
- void [log\\_rinex\\_nav\\_gps\\_cnav](#) (int type\_of\_rx, const std::map< int32\_t, [Gps\\_CNAV\\_Ephemeris](#) > &new\_↔  
 cnav\_eph)  
*Print RINEX annotation for GPS CNAV message.*
- void [log\\_rinex\\_nav\\_gal\\_nav](#) (int type\_of\_rx, const std::map< int32\_t, [Galileo\\_Ephemeris](#) > &new\_gal\_eph)  
*Print RINEX annotation for Galileo NAV message.*
- void [log\\_rinex\\_nav\\_glo\\_gnav](#) (int type\_of\_rx, const std::map< int32\_t, [Glonass\\_Gnav\\_Ephemeris](#) > &new\_↔  
 glo\_eph)  
*Print RINEX annotation for Glonass GNAV message.*
- void [log\\_rinex\\_nav\\_bds\\_dnav](#) (int type\_of\_rx, const std::map< int32\_t, [Beidou\\_Dnav\\_Ephemeris](#) > &new\_↔  
 bds\_eph)  
*Print RINEX annotation for BeiDou DNAV message.*

- void [set\\_pre\\_2009\\_file](#) (bool pre\_2009\_file)  
*Set processing for signals older than 2009.*
- bool [is\\_rinex\\_header\\_written](#) () const  
*Returns true is the RINEX file headers are already written.*
- std::vector< std::string > [get\\_navfilename](#) () const  
*Returns name of RINEX navigation file(s)*
- std::string [get\\_obsfilename](#) () const  
*Returns name of RINEX observation file.*

### 10.335.1 Detailed Description

Class that handles the generation of Receiver INdependent EXchange format (RINEX) files.  
Definition at line 83 of file [rinex\\_printer.h](#).

### 10.335.2 Constructor & Destructor Documentation

#### 10.335.2.1 Rinex\_Printer()

```
Rinex_Printer::Rinex_Printer (
    int version = 0,
    const std::string & base_path = ".",
    const std::string & base_name = "-") [explicit]
```

Constructor. Creates GNSS Navigation and Observables RINEX files.

#### 10.335.2.2 ~Rinex\_Printer()

```
Rinex_Printer::~~Rinex_Printer ()
```

Destructor. Removes created files if empty.

### 10.335.3 Member Function Documentation

#### 10.335.3.1 get\_navfilename()

```
std::vector< std::string > Rinex_Printer::get_navfilename () const [inline]
```

Returns name of RINEX navigation file(s)

Definition at line 222 of file [rinex\\_printer.h](#).

#### 10.335.3.2 get\_obsfilename()

```
std::string Rinex_Printer::get_obsfilename () const [inline]
```

Returns name of RINEX observation file.

Definition at line 230 of file [rinex\\_printer.h](#).

#### 10.335.3.3 is\_rinex\_header\_written()

```
bool Rinex_Printer::is_rinex_header_written () const [inline]
```

Returns true is the RINEX file headers are already written.

Definition at line 214 of file [rinex\\_printer.h](#).

#### 10.335.3.4 log\_rinex\_nav\_bds\_dnav()

```
void Rinex_Printer::log_rinex_nav_bds_dnav (
    int type_of_rx,
    const std::map< int32_t, Beidou_Dnav_Ephemeris > & new_bds_eph)
```

Print RINEX annotation for BeiDou DNAV message.

**10.335.3.5 log\_rinex\_nav\_gal\_nav()**

```
void Rinex_Printer::log_rinex_nav_gal_nav (
    int type_of_rx,
    const std::map< int32_t, Galileo_Ephemeris > & new_gal_eph)
```

Print RINEX annotation for Galileo NAV message.

**10.335.3.6 log\_rinex\_nav\_glo\_gnav()**

```
void Rinex_Printer::log_rinex_nav_glo_gnav (
    int type_of_rx,
    const std::map< int32_t, Glonass_Gnav_Ephemeris > & new_glo_eph)
```

Print RINEX annotation for Glonass GNAV message.

**10.335.3.7 log\_rinex\_nav\_gps\_cnav()**

```
void Rinex_Printer::log_rinex_nav_gps_cnav (
    int type_of_rx,
    const std::map< int32_t, Gps_CNAV_Ephemeris > & new_cnav_eph)
```

Print RINEX annotation for GPS CNAV message.

**10.335.3.8 log\_rinex\_nav\_gps\_nav()**

```
void Rinex_Printer::log_rinex_nav_gps_nav (
    int type_of_rx,
    const std::map< int32_t, Gps_Ephemeris > & new_eph)
```

Print RINEX annotation for GPS NAV message.

**10.335.3.9 print\_rinex\_annotation()**

```
void Rinex_Printer::print_rinex_annotation (
    const Rtklib_Solver * pvt_solver,
    const std::map< int, Gnss_Synchro > & gnss_observables_map,
    double rx_time,
    int type_of_rx,
    bool flag_write_RINEX_obs_output)
```

Print RINEX annotation. If it is the first annotation, it also prints the RINEX headers for navigation and observation files. If it is not the first annotation, it only annotates the observation, and updates the navigation header if UTC data was not available when writing it for the first time. The meaning of type\_of\_rx is as follows:

type_of_rx	Signals
0	Unknown
1	GPS L1 C/A
2	GPS L2C
3	GPS L5
4	Galileo E1B
5	Galileo E5a
6	Galileo E5b
7	GPS L1 C/A + GPS L2C
8	GPS L1 C/A + GPS L5
9	GPS L1 C/A + Galileo E1B
10	GPS L1 C/A + Galileo E5a
11	GPS L1 C/A + Galileo E5b
12	Galileo E1B + GPS L2C
13	Galileo E5a + GPS L5
14	Galileo E1B + Galileo E5a
15	Galileo E1B + Galileo E5b
16	GPS L2C + GPS L5

type_of_rx	Signals
17	GPS L2C + Galileo E5a
20	GPS L5 + Galileo E5b
21	GPS L1 C/A + Galileo E1B + GPS L2C
22	GPS L1 C/A + Galileo E1B + GPS L5
23	GLONASS L1 C/A
24	GLONASS L2 C/A
25	GLONASS L1 C/A + GLONASS L2 C/A
26	GPS L1 C/A + GLONASS L1 C/A
27	Galileo E1B + GLONASS L1 C/A
28	GPS L2C + GLONASS L1 C/A
29	GPS L1 C/A + GLONASS L2 C/A
30	Galileo E1B + GLONASS L2 C/A
31	GPS L2C + GLONASS L2 C/A
32	GPS L1 C/A + Galileo E1B + GPS L5 + Galileo E5a
33	GPS L1 C/A + Galileo E1B + Galileo E5a
100	Galileo E6B
101	Galileo E1B + Galileo E6B
102	Galileo E5a + Galileo E6B
103	Galileo E5b + Galileo E6B
104	Galileo E1B + Galileo E5a + Galileo E6B
105	Galileo E1B + Galileo E5b + Galileo E6B
106	GPS L1 C/A + Galileo E1B + Galileo E6B
107	GPS L1 C/A + Galileo E6B
108	GPS L1 C/A + Galileo E1B + GPS L5 + Galileo E5a + Galileo E6B
500	BeiDou B1I
501	BeiDou B1I + GPS L1 C/A
502	BeiDou B1I + Galileo E1B
503	BeiDou B1I + GLONASS L1 C/A
504	BeiDou B1I + GPS L1 C/A + Galileo E1B
505	BeiDou B1I + GPS L1 C/A + GLONASS L1 C/A + Galileo E1B
506	BeiDou B1I + BeiDou B3I
600	BeiDou B3I
601	BeiDou B3I + GPS L2C
602	BeiDou B3I + GLONASS L2 C/A
603	BeiDou B3I + GPS L2C + GLONASS L2 C/A
604	BeiDou B3I + GPS L1 C/A
605	BeiDou B3I + Galileo E1B
606	BeiDou B3I + GLONASS L1 C/A
607	BeiDou B3I + GPS L1 C/A + Galileo E1B
608	BeiDou B3I + GPS L1 C/A + Galileo E1B + BeiDou B1I
609	BeiDou B3I + GPS L1 C/A + Galileo E1B + GLONASS L1 C/A
610	BeiDou B3I + GPS L1 C/A + Galileo E1B + GLONASS L1 C/A + BeiDou B1I
1000	GPS L1 C/A + GPS L2C + GPS L5
1001	GPS L1 C/A + Galileo E1B + GPS L2C + GPS L5 + Galileo E5a

### 10.335.3.10 set\_pre\_2009\_file()

```
void Rinex_Printer::set_pre_2009_file (
    bool pre_2009_file)
```

Set processing for signals older than 2009.

The documentation for this class was generated from the following file:

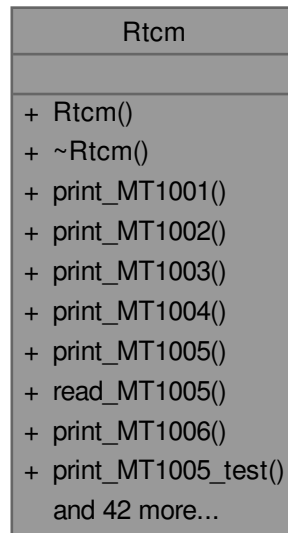
- [rinex\\_printer.h](#)

## 10.336 RtcM Class Reference

This class implements the generation and reading of some Message Types defined in the RTCM 3.2 Standard, plus some utilities to handle messages.

```
#include <rtcM.h>
```

Collaboration diagram for RtcM:



### Public Member Functions

- [RtcM](#) (uint16\_t port=2101)
 

*Default constructor that sets TCP port of the RTCM message server and RTCM Station ID. 2101 is the standard RTCM port according to the Internet Assigned Numbers Authority (IANA). See <https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml>.*
- std::string [print\\_MT1001](#) (const [Gps\\_Ephemeris](#) &gps\_eph, double obs\_time, const std::map< int32\_t, [Gnss\\_Synchro](#) > &observables, uint16\_t station\_id)
 

*Prints message type 1001 (L1-Only GPS RTK Observables)*
- std::string [print\\_MT1002](#) (const [Gps\\_Ephemeris](#) &gps\_eph, double obs\_time, const std::map< int32\_t, [Gnss\\_Synchro](#) > &observables, uint16\_t station\_id)
 

*Prints message type 1002 (Extended L1-Only GPS RTK Observables)*
- std::string [print\\_MT1003](#) (const [Gps\\_Ephemeris](#) &ephL1, const [Gps\\_CNAV\\_Ephemeris](#) &ephL2, double obs\_time, const std::map< int32\_t, [Gnss\\_Synchro](#) > &observables, uint16\_t station\_id)
 

*Prints message type 1003 (L1 & L2 GPS RTK Observables)*
- std::string [print\\_MT1004](#) (const [Gps\\_Ephemeris](#) &ephL1, const [Gps\\_CNAV\\_Ephemeris](#) &ephL2, double obs\_time, const std::map< int32\_t, [Gnss\\_Synchro](#) > &observables, uint16\_t station\_id)
 

*Prints message type 1004 (Extended L1 & L2 GPS RTK Observables)*
- std::string [print\\_MT1005](#) (uint32\_t ref\_id, double ecef\_x, double ecef\_y, double ecef\_z, bool gps, bool glonass, bool galileo, bool non\_physical, bool single\_oscillator, uint32\_t quarter\_cycle\_indicator)
 

*Prints message type 1005 (Stationary Antenna Reference Point)*
- int32\_t [read\\_MT1005](#) (const std::string &message, uint32\_t &ref\_id, double &ecef\_x, double &ecef\_y, double &ecef\_z, bool &gps, bool &glonass, bool &galileo)
 

*Verifies and reads messages of type 1005 (Stationary Antenna Reference Point). Returns 1 if anything goes wrong, 0 otherwise.*

- std::string [print\\_MT1006](#) (uint32\_t ref\_id, double ecef\_x, double ecef\_y, double ecef\_z, bool gps, bool glonass, bool galileo, bool non\_physical, bool single\_oscillator, uint32\_t quarter\_cycle\_indicator, double height)  
*Prints message type 1006 (Stationary Antenna Reference Point, with Height Information)*
- std::string [print\\_MT1005\\_test](#) ()  
*For testing purposes.*
- std::string [print\\_MT1008](#) (uint32\_t ref\_id, const std::string &antenna\_descriptor, uint32\_t antenna\_setup\_id, const std::string &antenna\_serial\_number)  
*Prints message type 1008 (Antenna Descriptor & Serial Number)*
- std::string [print\\_MT1009](#) (const [Glonass\\_Gnav\\_Ephemeris](#) &glonass\_gnav\_eph, double obs\_time, const std::map< int32\_t, [Gnss\\_Synchro](#) > &observables, uint16\_t station\_id)  
*Prints L1-Only GLONASS RTK Observables.*
- std::string [print\\_MT1010](#) (const [Glonass\\_Gnav\\_Ephemeris](#) &glonass\_gnav\_eph, double obs\_time, const std::map< int32\_t, [Gnss\\_Synchro](#) > &observables, uint16\_t station\_id)  
*Prints Extended L1-Only GLONASS RTK Observables.*
- std::string [print\\_MT1011](#) (const [Glonass\\_Gnav\\_Ephemeris](#) &glonass\_gnav\_ephL1, const [Glonass\\_Gnav\\_Ephemeris](#) &glonass\_gnav\_ephL2, double obs\_time, const std::map< int32\_t, [Gnss\\_Synchro](#) > &observables, uint16\_t station\_id)  
*Prints L1&L2 GLONASS RTK Observables.*
- std::string [print\\_MT1012](#) (const [Glonass\\_Gnav\\_Ephemeris](#) &glonass\_gnav\_ephL1, const [Glonass\\_Gnav\\_Ephemeris](#) &glonass\_gnav\_ephL2, double obs\_time, const std::map< int32\_t, [Gnss\\_Synchro](#) > &observables, uint16\_t station\_id)  
*Prints Extended L1&L2 GLONASS RTK Observables.*
- std::string [print\\_MT1019](#) (const [Gps\\_Ephemeris](#) &gps\_eph)  
*Prints message type 1019 (GPS Ephemeris), should be broadcast in the event that the IODC does not match the IODE, and every 2 minutes.*
- int32\_t [read\\_MT1019](#) (const std::string &message, [Gps\\_Ephemeris](#) &gps\_eph) const  
*Verifies and reads messages of type 1019 (GPS Ephemeris). Returns 1 if anything goes wrong, 0 otherwise.*
- std::string [print\\_MT1020](#) (const [Glonass\\_Gnav\\_Ephemeris](#) &glonass\_gnav\_eph, const [Glonass\\_Gnav\\_Utc\\_Model](#) &glonass\_gnav\_utc\_model)  
*Prints message type 1020 (GLONASS Ephemeris).*
- int32\_t [read\\_MT1020](#) (const std::string &message, [Glonass\\_Gnav\\_Ephemeris](#) &glonass\_gnav\_eph, [Glonass\\_Gnav\\_Utc\\_Model](#) &glonass\_gnav\_utc\_model) const  
*Verifies and reads messages of type 1020 (GLONASS Ephemeris).*
- std::string [print\\_MT1029](#) (uint32\_t ref\_id, const [Gps\\_Ephemeris](#) &gps\_eph, double obs\_time, const std::string &message)  
*Prints message type 1029 (Unicode Text String)*
- std::string [print\\_MT1045](#) (const [Galileo\\_Ephemeris](#) &gal\_eph)  
*Prints message type 1045 (Galileo Ephemeris), should be broadcast every 2 minutes.*
- int32\_t [read\\_MT1045](#) (const std::string &message, [Galileo\\_Ephemeris](#) &gal\_eph) const  
*Verifies and reads messages of type 1045 (Galileo Ephemeris). Returns 1 if anything goes wrong, 0 otherwise.*
- std::string [print\\_MSM\\_1](#) (const [Gps\\_Ephemeris](#) &gps\_eph, const [Gps\\_CNAV\\_Ephemeris](#) &gps\_cnav\_eph, const [Galileo\\_Ephemeris](#) &gal\_eph, const [Glonass\\_Gnav\\_Ephemeris](#) &glonass\_gnav\_eph, double obs\_time, const std::map< int32\_t, [Gnss\\_Synchro](#) > &observables, uint32\_t ref\_id, uint32\_t clock\_steering\_indicator, uint32\_t external\_clock\_indicator, int32\_t smooth\_int, bool divergence\_free, bool more\_messages)  
*Prints messages of type MSM1 (Compact GNSS observables)*
- std::string [print\\_MSM\\_2](#) (const [Gps\\_Ephemeris](#) &gps\_eph, const [Gps\\_CNAV\\_Ephemeris](#) &gps\_cnav\_eph, const [Galileo\\_Ephemeris](#) &gal\_eph, const [Glonass\\_Gnav\\_Ephemeris](#) &glonass\_gnav\_eph, double obs\_time, const std::map< int32\_t, [Gnss\\_Synchro](#) > &observables, uint32\_t ref\_id, uint32\_t clock\_steering\_indicator, uint32\_t external\_clock\_indicator, int32\_t smooth\_int, bool divergence\_free, bool more\_messages)  
*Prints messages of type MSM2 (Compact GNSS phaseranges)*



- `std::string print_MSM_3` (const [Gps\\_Ephemeris](#) &gps\_eph, const [Gps\\_CNAV\\_Ephemeris](#) &gps\_cnav\_eph, const [Galileo\\_Ephemeris](#) &gal\_eph, const [Glonass\\_Gnav\\_Ephemeris](#) &glo\_gnav\_eph, double obs\_time, const std::map< int32\_t, [Gnss\\_Synchro](#) > &observables, uint32\_t ref\_id, uint32\_t clock\_steering\_indicator, uint32\_t external\_clock\_indicator, int32\_t smooth\_int, bool divergence\_free, bool more\_messages)  
*Prints messages of type MSM3 (Compact GNSS pseudoranges and phaseranges)*
- `std::string print_MSM_4` (const [Gps\\_Ephemeris](#) &gps\_eph, const [Gps\\_CNAV\\_Ephemeris](#) &gps\_cnav\_eph, const [Galileo\\_Ephemeris](#) &gal\_eph, const [Glonass\\_Gnav\\_Ephemeris](#) &glo\_gnav\_eph, double obs\_time, const std::map< int32\_t, [Gnss\\_Synchro](#) > &observables, uint32\_t ref\_id, uint32\_t clock\_steering\_indicator, uint32\_t external\_clock\_indicator, int32\_t smooth\_int, bool divergence\_free, bool more\_messages)  
*Prints messages of type MSM4 (Full GNSS pseudoranges and phaseranges plus CNR)*
- `std::string print_MSM_5` (const [Gps\\_Ephemeris](#) &gps\_eph, const [Gps\\_CNAV\\_Ephemeris](#) &gps\_cnav\_eph, const [Galileo\\_Ephemeris](#) &gal\_eph, const [Glonass\\_Gnav\\_Ephemeris](#) &glo\_gnav\_eph, double obs\_time, const std::map< int32\_t, [Gnss\\_Synchro](#) > &observables, uint32\_t ref\_id, uint32\_t clock\_steering\_indicator, uint32\_t external\_clock\_indicator, int32\_t smooth\_int, bool divergence\_free, bool more\_messages)  
*Prints messages of type MSM5 (Full GNSS pseudoranges, phaseranges, phaserange rate and CNR)*
- `std::string print_MSM_6` (const [Gps\\_Ephemeris](#) &gps\_eph, const [Gps\\_CNAV\\_Ephemeris](#) &gps\_cnav\_eph, const [Galileo\\_Ephemeris](#) &gal\_eph, const [Glonass\\_Gnav\\_Ephemeris](#) &glo\_gnav\_eph, double obs\_time, const std::map< int32\_t, [Gnss\\_Synchro](#) > &observables, uint32\_t ref\_id, uint32\_t clock\_steering\_indicator, uint32\_t external\_clock\_indicator, int32\_t smooth\_int, bool divergence\_free, bool more\_messages)  
*Prints messages of type MSM6 (Full GNSS pseudoranges and phaseranges plus CNR, high resolution)*
- `std::string print_MSM_7` (const [Gps\\_Ephemeris](#) &gps\_eph, const [Gps\\_CNAV\\_Ephemeris](#) &gps\_cnav\_eph, const [Galileo\\_Ephemeris](#) &gal\_eph, const [Glonass\\_Gnav\\_Ephemeris](#) &glo\_gnav\_eph, double obs\_time, const std::map< int32\_t, [Gnss\\_Synchro](#) > &observables, uint32\_t ref\_id, uint32\_t clock\_steering\_indicator, uint32\_t external\_clock\_indicator, int32\_t smooth\_int, bool divergence\_free, bool more\_messages)  
*Prints messages of type MSM7 (Full GNSS pseudoranges, phaseranges, phaserange rate and CNR, high resolution)*
- `std::vector< std::string > print_IGM01` (const [Galileo\\_HAS\\_data](#) &has\_data)  
*Prints messages of type IGM01 (SSR Orbit Correction)*
- `std::vector< std::string > print_IGM02` (const [Galileo\\_HAS\\_data](#) &has\_data)  
*Prints messages of type IGM02 (SSR Clock Correction)*
- `std::vector< std::string > print_IGM03` (const [Galileo\\_HAS\\_data](#) &has\_data)  
*Prints messages of type IGM03 (SSR Combined Orbit and Clock Correction)*
- `std::vector< std::string > print_IGM05` (const [Galileo\\_HAS\\_data](#) &has\_data)  
*Prints messages of type IGM05 (SSR Bias Correction)*
- `uint32_t lock_time` (const [Gps\\_Ephemeris](#) &eph, double obs\_time, const [Gnss\\_Synchro](#) &gnss\_synchro)  
*Returns the time period in which GPS L1 signals have been continually tracked.*
- `uint32_t lock_time` (const [Gps\\_CNAV\\_Ephemeris](#) &eph, double obs\_time, const [Gnss\\_Synchro](#) &gnss\_synchro)  
*Returns the time period in which GPS L2 signals have been continually tracked.*
- `uint32_t lock_time` (const [Galileo\\_Ephemeris](#) &eph, double obs\_time, const [Gnss\\_Synchro](#) &gnss\_synchro)  
*Returns the time period in which Galileo signals have been continually tracked.*
- `uint32_t lock_time` (const [Glonass\\_Gnav\\_Ephemeris](#) &eph, double obs\_time, const [Gnss\\_Synchro](#) &gnss\_synchro)  
*Locks time period in which GLONASS signals have been continually tracked.*
- `std::string bin_to_hex` (const std::string &s) const  
*Returns a string of hexadecimal symbols from a string of binary symbols.*
- `std::string hex_to_bin` (const std::string &s) const  
*Returns a string of binary symbols from a string of hexadecimal symbols.*
- `std::string bin_to_binary_data` (const std::string &s) const  
*Returns a string of binary data from a string of binary symbols.*
- `std::string binary_data_to_bin` (const std::string &s) const  
*Returns a string of binary symbols from a string of binary data.*
- `uint32_t bin_to_uint` (const std::string &s) const  
*Returns an uint32\_t from a string of binary symbols.*

- `int32_t bin_to_int (const std::string &s) const`
- `double bin_to_double (const std::string &s) const`  
Returns double from a string of binary symbols.
- `int32_t bin_to_sint (const std::string &s) const`
- `uint64_t hex_to_uint (const std::string &s) const`  
Returns an `uint64_t` from a string of hexadecimal symbols.
- `int64_t hex_to_int (const std::string &s) const`  
Returns a `int64_t` from a string of hexadecimal symbols.
- `bool check_CRC (const std::string &message) const`  
Checks that the CRC of a RTCM package is correct.
- `void run_server ()`  
Starts running the server.
- `void stop_server ()`  
Stops the server.
- `void send_message (const std::string &msg)`  
Sends a message through the server to all connected clients.
- `bool is_server_running () const`  
Returns true if the server is running, false otherwise.

### 10.336.1 Detailed Description

This class implements the generation and reading of some Message Types defined in the RTCM 3.2 Standard, plus some utilities to handle messages.

Generation of the following Message Types: 1001, 1002, 1003, 1004, 1005, 1006, 1008, 1019, 1020, 1029, 1045

Decoding of the following Message Types: 1019, 1045

Generation of the following Multiple Signal Messages: MSM1 (message types 1071, 1091) MSM2 (message types 1072, 1092) MSM3 (message types 1073, 1093) MSM4 (message types 1074, 1094) MSM5 (message types 1075, 1095) MSM6 (message types 1076, 1096) MSM7 (message types 1077, 1097)

RTCM 3 message format (size in bits): +-----+-----+-----+-----+ | preamble | 000000  
| length | data message | parity | +-----+-----+-----+-----+ |<- 8 --->|<- 6 -->|<- 10  
--->|<- length x 8 --->|<- 24 -->| +-----+-----+-----+-----+ +

(C) Carles Fernandez-Prades, 2015. cfernandez(at)cttc.es

Definition at line 97 of file [rtcm.h](#).

### 10.336.2 Constructor & Destructor Documentation

#### 10.336.2.1 Rtcn()

```
Rtcn::Rtcn (
    uint16_t port = 2101) [explicit]
```

Default constructor that sets TCP port of the RTCM message server and RTCM Station ID. 2101 is the standard RTCM port according to the Internet Assigned Numbers Authority (IANA). See <https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml>.

References [Rtcn\(\)](#).

Referenced by [Rtcn\(\)](#).

Here is the call graph for this function:



### 10.336.3 Member Function Documentation

#### 10.336.3.1 bin\_to\_binary\_data()

```
std::string RtcM::bin_to_binary_data (  
    const std::string & s) const
```

Returns a string of binary data from a string of binary symbols.

#### 10.336.3.2 bin\_to\_double()

```
double RtcM::bin_to_double (  
    const std::string & s) const
```

Returns double from a string of binary symbols.

#### 10.336.3.3 bin\_to\_hex()

```
std::string RtcM::bin_to_hex (  
    const std::string & s) const
```

Returns a string of hexadecimal symbols from a string of binary symbols.

#### 10.336.3.4 bin\_to\_uint()

```
uint32_t RtcM::bin_to_uint (  
    const std::string & s) const
```

Returns an uint32\_t from a string of binary symbols.

#### 10.336.3.5 binary\_data\_to\_bin()

```
std::string RtcM::binary_data_to_bin (  
    const std::string & s) const
```

Returns a string of binary symbols from a string of binary data.

#### 10.336.3.6 check\_CRC()

```
bool RtcM::check_CRC (  
    const std::string & message) const
```

Checks that the CRC of a RTCM package is correct.

#### 10.336.3.7 hex\_to\_bin()

```
std::string RtcM::hex_to_bin (  
    const std::string & s) const
```

Returns a string of binary symbols from a string of hexadecimal symbols.

#### 10.336.3.8 hex\_to\_int()

```
int64_t RtcM::hex_to_int (  
    const std::string & s) const
```

Returns a int64\_t from a string of hexadecimal symbols.

#### 10.336.3.9 hex\_to\_uint()

```
uint64_t RtcM::hex_to_uint (  
    const std::string & s) const
```

Returns an uint64\_t from a string of hexadecimal symbols.

#### 10.336.3.10 is\_server\_running()

```
bool RtcM::is_server_running () const
```

Returns true if the server is running, false otherwise.

**10.336.3.11 lock\_time() [1/4]**

```
uint32_t RtcM::lock_time (
    const Galileo_Ephemeris & eph,
    double obs_time,
    const Gnss_Synchro & gnss_synchro)
```

Returns the time period in which Galileo signals have been continually tracked.

**10.336.3.12 lock\_time() [2/4]**

```
uint32_t RtcM::lock_time (
    const Glonass_Gnav_Ephemeris & eph,
    double obs_time,
    const Gnss_Synchro & gnss_synchro)
```

Locks time period in which GLONASS signals have been continually tracked.

**Note**

Code added as part of GSoC 2017 program

**Parameters**

<i>eph</i>	GLONASS GNAV Broadcast Ephemeris
<i>obs_time</i>	Time of observation at the moment of printing
<i>observables</i>	Set of observables as defined by the platform

**Returns**

Returns the time period in which GLONASS signals have been continually tracked.

**10.336.3.13 lock\_time() [3/4]**

```
uint32_t RtcM::lock_time (
    const Gps_CNAV_Ephemeris & eph,
    double obs_time,
    const Gnss_Synchro & gnss_synchro)
```

Returns the time period in which GPS L2 signals have been continually tracked.

**10.336.3.14 lock\_time() [4/4]**

```
uint32_t RtcM::lock_time (
    const Gps_Ephemeris & eph,
    double obs_time,
    const Gnss_Synchro & gnss_synchro)
```

Returns the time period in which GPS L1 signals have been continually tracked.

**10.336.3.15 print\_IGM01()**

```
std::vector< std::string > RtcM::print_IGM01 (
    const Galileo_HAS_data & has_data)
```

Prints messages of type IGM01 (SSR Orbit Correction)

**10.336.3.16 print\_IGM02()**

```
std::vector< std::string > RtcM::print_IGM02 (
    const Galileo_HAS_data & has_data)
```

Prints messages of type IGM02 (SSR Clock Correction)

**10.336.3.17 print\_IGM03()**

```
std::vector< std::string > RtcM::print_IGM03 (
    const Galileo_HAS_data & has_data)
```

Prints messages of type IGM03 (SSR Combined Orbit and Clock Correction)

**10.336.3.18 print\_IGM05()**

```
std::vector< std::string > RtcM::print_IGM05 (
    const Galileo_HAS_data & has_data)
```

Prints messages of type IGM05 (SSR Bias Correction)

**10.336.3.19 print\_MSM\_1()**

```
std::string RtcM::print_MSM_1 (
    const Gps_Ephemeris & gps_eph,
    const Gps_CNAV_Ephemeris & gps_cnav_eph,
    const Galileo_Ephemeris & gal_eph,
    const Glonass_Gnav_Ephemeris & glo_gnav_eph,
    double obs_time,
    const std::map< int32_t, Gnss_Synchro > & observables,
    uint32_t ref_id,
    uint32_t clock_steering_indicator,
    uint32_t external_clock_indicator,
    int32_t smooth_int,
    bool divergence_free,
    bool more_messages)
```

Prints messages of type MSM1 (Compact GNSS observables)

**10.336.3.20 print\_MSM\_2()**

```
std::string RtcM::print_MSM_2 (
    const Gps_Ephemeris & gps_eph,
    const Gps_CNAV_Ephemeris & gps_cnav_eph,
    const Galileo_Ephemeris & gal_eph,
    const Glonass_Gnav_Ephemeris & glo_gnav_eph,
    double obs_time,
    const std::map< int32_t, Gnss_Synchro > & observables,
    uint32_t ref_id,
    uint32_t clock_steering_indicator,
    uint32_t external_clock_indicator,
    int32_t smooth_int,
    bool divergence_free,
    bool more_messages)
```

Prints messages of type MSM2 (Compact GNSS phaseranges)

**10.336.3.21 print\_MSM\_3()**

```
std::string RtcM::print_MSM_3 (
    const Gps_Ephemeris & gps_eph,
    const Gps_CNAV_Ephemeris & gps_cnav_eph,
    const Galileo_Ephemeris & gal_eph,
    const Glonass_Gnav_Ephemeris & glo_gnav_eph,
    double obs_time,
    const std::map< int32_t, Gnss_Synchro > & observables,
    uint32_t ref_id,
    uint32_t clock_steering_indicator,
    uint32_t external_clock_indicator,
    int32_t smooth_int,
```

```

    bool divergence_free,
    bool more_messages)

```

Prints messages of type MSM3 (Compact GNSS pseudoranges and phaseranges)

### 10.336.3.22 print\_MSM\_4()

```

std::string RtcM::print_MSM_4 (
    const Gps_Ephemeris & gps_eph,
    const Gps_CNAV_Ephemeris & gps_cnav_eph,
    const Galileo_Ephemeris & gal_eph,
    const Glonass_Gnav_Ephemeris & glo_gnav_eph,
    double obs_time,
    const std::map< int32_t, Gnss_Synchro > & observables,
    uint32_t ref_id,
    uint32_t clock_steering_indicator,
    uint32_t external_clock_indicator,
    int32_t smooth_int,
    bool divergence_free,
    bool more_messages)

```

Prints messages of type MSM4 (Full GNSS pseudoranges and phaseranges plus CNR)

### 10.336.3.23 print\_MSM\_5()

```

std::string RtcM::print_MSM_5 (
    const Gps_Ephemeris & gps_eph,
    const Gps_CNAV_Ephemeris & gps_cnav_eph,
    const Galileo_Ephemeris & gal_eph,
    const Glonass_Gnav_Ephemeris & glo_gnav_eph,
    double obs_time,
    const std::map< int32_t, Gnss_Synchro > & observables,
    uint32_t ref_id,
    uint32_t clock_steering_indicator,
    uint32_t external_clock_indicator,
    int32_t smooth_int,
    bool divergence_free,
    bool more_messages)

```

Prints messages of type MSM5 (Full GNSS pseudoranges, phaseranges, phaserange rate and CNR)

### 10.336.3.24 print\_MSM\_6()

```

std::string RtcM::print_MSM_6 (
    const Gps_Ephemeris & gps_eph,
    const Gps_CNAV_Ephemeris & gps_cnav_eph,
    const Galileo_Ephemeris & gal_eph,
    const Glonass_Gnav_Ephemeris & glo_gnav_eph,
    double obs_time,
    const std::map< int32_t, Gnss_Synchro > & observables,
    uint32_t ref_id,
    uint32_t clock_steering_indicator,
    uint32_t external_clock_indicator,
    int32_t smooth_int,
    bool divergence_free,
    bool more_messages)

```

Prints messages of type MSM6 (Full GNSS pseudoranges and phaseranges plus CNR, high resolution)

### 10.336.3.25 print\_MSM\_7()

```

std::string RtcM::print_MSM_7 (
    const Gps_Ephemeris & gps_eph,

```

```

    const Gps_CNAV_Ephemeris & gps_cnav_eph,
    const Galileo_Ephemeris & gal_eph,
    const Glonass_Gnav_Ephemeris & glo_gnav_eph,
    double obs_time,
    const std::map< int32_t, Gnss_Synchro > & observables,
    uint32_t ref_id,
    uint32_t clock_steering_indicator,
    uint32_t external_clock_indicator,
    int32_t smooth_int,
    bool divergence_free,
    bool more_messages)

```

Prints messages of type MSM7 (Full GNSS pseudoranges, phaseranges, phaserange rate and CNR, high resolution)

#### 10.336.3.26 print\_MT1001()

```

std::string RtcM::print_MT1001 (
    const Gps_Ephemeris & gps_eph,
    double obs_time,
    const std::map< int32_t, Gnss_Synchro > & observables,
    uint16_t station_id)

```

Prints message type 1001 (L1-Only GPS RTK Observables)

#### 10.336.3.27 print\_MT1002()

```

std::string RtcM::print_MT1002 (
    const Gps_Ephemeris & gps_eph,
    double obs_time,
    const std::map< int32_t, Gnss_Synchro > & observables,
    uint16_t station_id)

```

Prints message type 1002 (Extended L1-Only GPS RTK Observables)

#### 10.336.3.28 print\_MT1003()

```

std::string RtcM::print_MT1003 (
    const Gps_Ephemeris & ephL1,
    const Gps_CNAV_Ephemeris & ephL2,
    double obs_time,
    const std::map< int32_t, Gnss_Synchro > & observables,
    uint16_t station_id)

```

Prints message type 1003 (L1 & L2 GPS RTK Observables)

#### 10.336.3.29 print\_MT1004()

```

std::string RtcM::print_MT1004 (
    const Gps_Ephemeris & ephL1,
    const Gps_CNAV_Ephemeris & ephL2,
    double obs_time,
    const std::map< int32_t, Gnss_Synchro > & observables,
    uint16_t station_id)

```

Prints message type 1004 (Extended L1 & L2 GPS RTK Observables)

#### 10.336.3.30 print\_MT1005()

```

std::string RtcM::print_MT1005 (
    uint32_t ref_id,
    double ecef_x,
    double ecef_y,
    double ecef_z,

```

```

    bool gps,
    bool glonass,
    bool galileo,
    bool non_physical,
    bool single_oscillator,
    uint32_t quarter_cycle_indicator)

```

Prints message type 1005 (Stationary Antenna Reference Point)

#### 10.336.3.31 `print_MT1005_test()`

```
std::string RtcM::print_MT1005_test ()
```

For testing purposes.

#### 10.336.3.32 `print_MT1006()`

```

std::string RtcM::print_MT1006 (
    uint32_t ref_id,
    double ecef_x,
    double ecef_y,
    double ecef_z,
    bool gps,
    bool glonass,
    bool galileo,
    bool non_physical,
    bool single_oscillator,
    uint32_t quarter_cycle_indicator,
    double height)

```

Prints message type 1006 (Stationary Antenna Reference Point, with Height Information)

#### 10.336.3.33 `print_MT1008()`

```

std::string RtcM::print_MT1008 (
    uint32_t ref_id,
    const std::string & antenna_descriptor,
    uint32_t antenna_setup_id,
    const std::string & antenna_serial_number)

```

Prints message type 1008 (Antenna Descriptor & Serial Number)

#### 10.336.3.34 `print_MT1009()`

```

std::string RtcM::print_MT1009 (
    const Glonass\_Gnav\_Ephemeris & glonass_gnav_eph,
    double obs_time,
    const std::map< int32_t, Gnss\_Synchro > & observables,
    uint16_t station_id)

```

Prints L1-Only GLONASS RTK Observables.

This GLONASS message type is not generally used or supported; type 1012 is to be preferred.

#### Note

Code added as part of GSoC 2017 program

#### Parameters

<i>glonass_gnav_eph</i>	GLONASS GNAV Broadcast Ephemeris
<i>obs_time</i>	Time of observation at the moment of printing
<i>observables</i>	Set of observables as defined by the platform



**Returns**

string with message contents

**10.336.3.35 print\_MT1010()**

```
std::string RtcM::print_MT1010 (
    const Glonass_Gnav_Ephemeris & glonass_gnav_eph,
    double obs_time,
    const std::map< int32_t, Gnss_Synchro > & observables,
    uint16_t station_id)
```

Prints Extended L1-Only GLONASS RTK Observables.

This GLONASS message type is used when only L1 data is present and bandwidth is very tight, often 1012 is used in such cases.

**Note**

Code added as part of GSoC 2017 program

**Parameters**

<i>glonass_gnav_eph</i>	GLONASS GNAV Broadcast Ephemeris
<i>obs_time</i>	Time of observation at the moment of printing
<i>observables</i>	Set of observables as defined by the platform

**Returns**

string with message contents

**10.336.3.36 print\_MT1011()**

```
std::string RtcM::print_MT1011 (
    const Glonass_Gnav_Ephemeris & glonass_gnav_ephL1,
    const Glonass_Gnav_Ephemeris & glonass_gnav_ephL2,
    double obs_time,
    const std::map< int32_t, Gnss_Synchro > & observables,
    uint16_t station_id)
```

Prints L1&L2 GLONASS RTK Observables.

This GLONASS message type is not generally used or supported; type 1012 is to be preferred

**Note**

Code added as part of GSoC 2017 program

**Parameters**

<i>glonass_gnav_eph</i>	GLONASS GNAV Broadcast Ephemeris
<i>obs_time</i>	Time of observation at the moment of printing
<i>observables</i>	Set of observables as defined by the platform

**Returns**

string with message contents

**10.336.3.37 print\_MT1012()**

```
std::string RtcM::print_MT1012 (
    const Glonass_Gnav_Ephemeris & glonass_gnav_ephL1,
    const Glonass_Gnav_Ephemeris & glonass_gnav_ephL2,
    double obs_time,
    const std::map< int32_t, Gnss_Synchro > & observables,
    uint16_t station_id)
```

Prints Extended L1&L2 GLONASS RTK Observables.

This GLONASS message type is the most common observational message type, with L1/L2/SNR content. This is one of the most common messages found.

**Note**

Code added as part of GSoC 2017 program

**Parameters**

<i>glonass_gnav_eph</i>	GLONASS GNAV Broadcast Ephemeris
<i>obs_time</i>	Time of observation at the moment of printing
<i>observables</i>	Set of observables as defined by the platform

**Returns**

string with message contents

**10.336.3.38 print\_MT1019()**

```
std::string RtcM::print_MT1019 (
    const Gps_Ephemeris & gps_eph)
```

Prints message type 1019 (GPS Ephemeris), should be broadcast in the event that the IODC does not match the IODE, and every 2 minutes.

**10.336.3.39 print\_MT1020()**

```
std::string RtcM::print_MT1020 (
    const Glonass_Gnav_Ephemeris & glonass_gnav_eph,
    const Glonass_Gnav_Utc_Model & glonass_gnav_utc_model)
```

Prints message type 1020 (GLONASS Ephemeris).

**Note**

Code added as part of GSoC 2017 program

**Parameters**

<i>glonass_gnav_eph</i>	GLONASS GNAV Broadcast Ephemeris
<i>glonass_gnav_utc_model</i>	GLONASS GNAV Clock Information

**Returns**

Returns message type as a string type

**10.336.3.40 print\_MT1029()**

```
std::string RtcM::print_MT1029 (
    uint32_t ref_id,
    const Gps_Ephemeris & gps_eph,
    double obs_time,
    const std::string & message)
```

Prints message type 1029 (Unicode Text String)

**10.336.3.41 print\_MT1045()**

```
std::string RtcM::print_MT1045 (
    const Galileo_Ephemeris & gal_eph)
```

Prints message type 1045 (Galileo Ephemeris), should be broadcast every 2 minutes.

**10.336.3.42 read\_MT1005()**

```
int32_t RtcM::read_MT1005 (
    const std::string & message,
    uint32_t & ref_id,
    double & ecef_x,
    double & ecef_y,
    double & ecef_z,
    bool & gps,
    bool & glonass,
    bool & galileo)
```

Verifies and reads messages of type 1005 (Stationary Antenna Reference Point). Returns 1 if anything goes wrong, 0 otherwise.

**10.336.3.43 read\_MT1019()**

```
int32_t RtcM::read_MT1019 (
    const std::string & message,
    Gps_Ephemeris & gps_eph) const
```

Verifies and reads messages of type 1019 (GPS Ephemeris). Returns 1 if anything goes wrong, 0 otherwise.

**10.336.3.44 read\_MT1020()**

```
int32_t RtcM::read_MT1020 (
    const std::string & message,
    Glonass_Gnav_Ephemeris & glonass_gnav_eph,
    Glonass_Gnav_Utc_Model & glonass_gnav_utc_model) const
```

Verifies and reads messages of type 1020 (GLONASS Ephemeris).

**Note**

Code added as part of GSoC 2017 program

**Parameters**

<i>message</i>	Message to read as a string type
<i>glonass_gnav_eph</i>	GLONASS GNAV Broadcast Ephemeris
<i>glonass_gnav_utc_model</i>	GLONASS GNAV Clock Information

**Returns**

Returns 1 if anything goes wrong, 0 otherwise.

**10.336.3.45 read\_MT1045()**

```
int32_t RtcM::read_MT1045 (
    const std::string & message,
    Galileo_Ephemeris & gal_eph) const
```

Verifies and reads messages of type 1045 (Galileo Ephemeris). Returns 1 if anything goes wrong, 0 otherwise.

**10.336.3.46 run\_server()**

```
void RtcM::run_server ()
```

Starts running the server.

### 10.336.3.47 send\_message()

```
void RtcM::send_message (
    const std::string & msg)
```

Sends a message through the server to all connected clients.

### 10.336.3.48 stop\_server()

```
void RtcM::stop_server ()
```

Stops the server.

The documentation for this class was generated from the following file:

- [rtcM.h](#)

## 10.337 RtcM\_Printer Class Reference

This class provides a implementation of a subset of the RTCM Standard 10403.2 messages.

```
#include <rtcM_printer.h>
```

Collaboration diagram for RtcM\_Printer:



### Public Member Functions

- [RtcM\\_Printer](#) (const std::string &filename, bool flag\_rtcM\_file\_dump, bool flag\_rtcM\_server, bool flag\_rtcM←\_tty\_port, uint16\_t rtcM\_tcp\_port, uint16\_t rtcM\_station\_id, const std::string &rtcM\_dump\_devname, bool time\_tag\_name=true, const std::string &base\_path=".")  
*Default constructor.*
- [~RtcM\\_Printer](#) ()  
*Default destructor.*
- void [Print\\_RtcM\\_Messages](#) (const [Rtklib\\_Solver](#) \*pvt\_solver, const std::map< int, [Gnss\\_Synchro](#) > &gnss←\_observables\_map, double rx\_time, int32\_t type\_of\_rx, int32\_t rtcM\_MSM\_rate\_ms, int32\_t rtcM\_MT1019←\_rate\_ms, int32\_t rtcM\_MT1020\_rate\_ms, int32\_t rtcM\_MT1045\_rate\_ms, int32\_t rtcM\_MT1077\_rate\_ms, int32\_t rtcM\_MT1097\_rate\_ms, bool flag\_write\_RTCM\_MSM\_output, bool flag\_write\_RTCM\_1019\_output, bool flag\_write\_RTCM\_1020\_output, bool flag\_write\_RTCM\_1045\_output, bool enable\_rx\_clock\_correction)  
*Print RTCM messages.*
- uint32\_t **lock\_time** (const [Gps\\_Ephemeris](#) &eph, double obs\_time, const [Gnss\\_Synchro](#) &gnss\_synchro)

- uint32\_t **lock\_time** (const [Gps\\_CNAV\\_Ephemeris](#) &eph, double obs\_time, const [Gnss\\_Synchro](#) &gnss\_synchro)
- uint32\_t **lock\_time** (const [Galileo\\_Ephemeris](#) &eph, double obs\_time, const [Gnss\\_Synchro](#) &gnss\_synchro)
- uint32\_t **lock\_time** (const [Glonass\\_Gnav\\_Ephemeris](#) &eph, double obs\_time, const [Gnss\\_Synchro](#) &gnss\_synchro)

*Locks time for logging given GLONASS GNAV Broadcast Ephemeris.*

- void **Print\_IGM\_Messages** (const [Galileo\\_HAS\\_data](#) &has\_data)
- std::string **print\_MT1005\_test** ()

*For testing purposes.*

### 10.337.1 Detailed Description

This class provides a implementation of a subset of the RTCM Standard 10403.2 messages.  
Definition at line 48 of file [rtcM\\_printer.h](#).

### 10.337.2 Constructor & Destructor Documentation

#### 10.337.2.1 RtcM\_Printer()

```
RtcM_Printer::RtcM_Printer (
    const std::string & filename,
    bool flag_rtcM_file_dump,
    bool flag_rtcM_server,
    bool flag_rtcM_tty_port,
    uint16_t rtcM_tcp_port,
    uint16_t rtcM_station_id,
    const std::string & rtcM_dump_devname,
    bool time_tag_name = true,
    const std::string & base_path = ".")
```

Default constructor.

#### 10.337.2.2 ~RtcM\_Printer()

```
RtcM_Printer::~~RtcM_Printer ()
```

Default destructor.

### 10.337.3 Member Function Documentation

#### 10.337.3.1 lock\_time()

```
uint32_t RtcM_Printer::lock_time (
    const Glonass\_Gnav\_Ephemeris & eph,
    double obs_time,
    const Gnss\_Synchro & gnss_synchro)
```

Locks time for logging given GLONASS GNAV Broadcast Ephemeris.

##### Note

Code added as part of GSoC 2017 program \params glonass\_gnav\_eph GLONASS GNAV Broadcast Ephemeris \params obs\_time Time of observation at the moment of printing \params observables Set of observables as defined by the platform

##### Returns

locked time during logging process

#### 10.337.3.2 print\_MT1005\_test()

```
std::string RtcM_Printer::print_MT1005_test ()
```

For testing purposes.

### 10.337.3.3 Print\_Rtcm\_Messages()

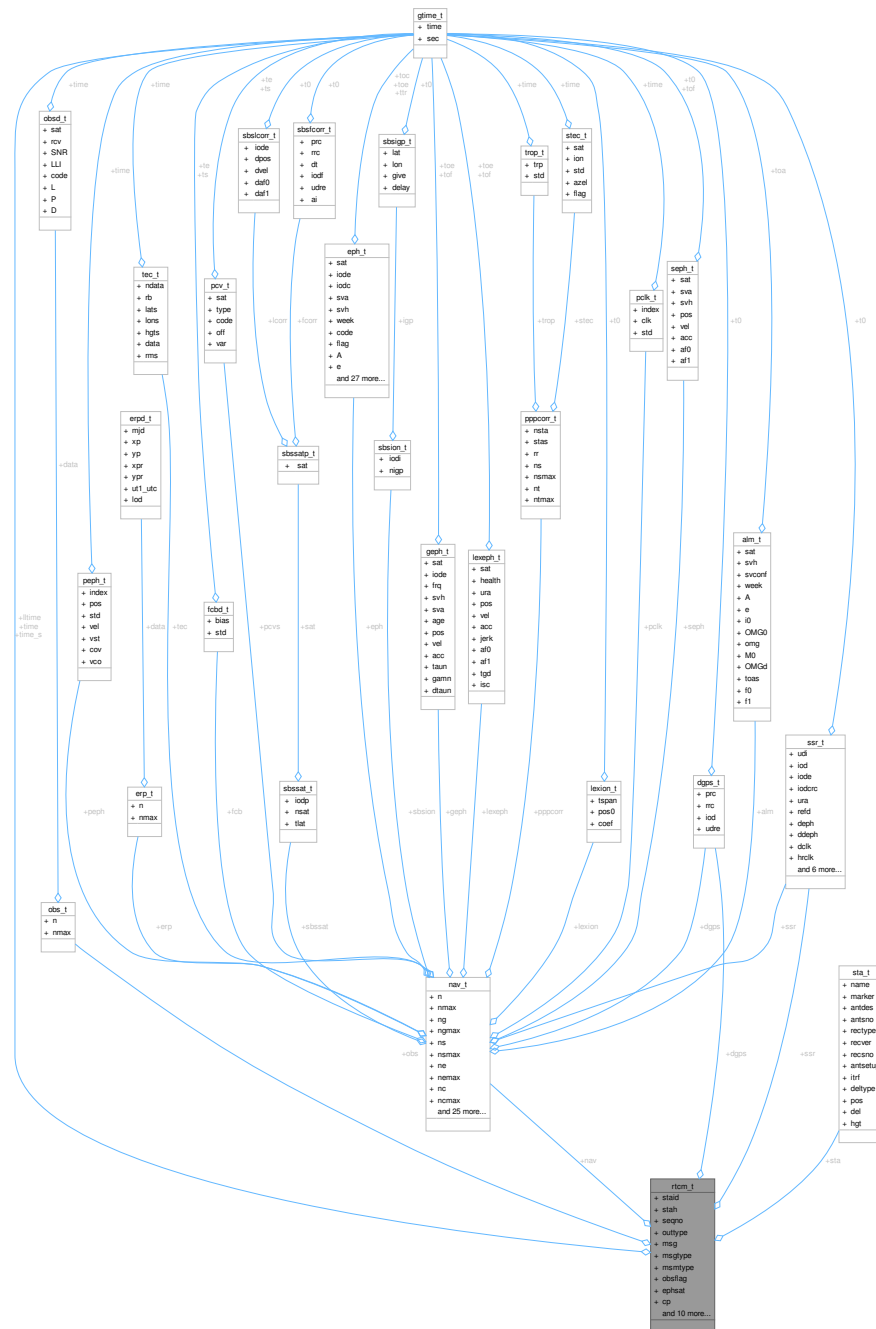
```
void Rtcm_Printer::Print_Rtcm_Messages (
    const Rtklib\_Solver * pvt_solver,
    const std::map< int, Gnss\_Synchro > & gnss_observables_map,
    double rx_time,
    int32_t type_of_rx,
    int32_t rtcm_MSM_rate_ms,
    int32_t rtcm_MT1019_rate_ms,
    int32_t rtcm_MT1020_rate_ms,
    int32_t rtcm_MT1045_rate_ms,
    int32_t rtcm_MT1077_rate_ms,
    int32_t rtcm_MT1097_rate_ms,
    bool flag_write_RTCM_MSM_output,
    bool flag_write_RTCM_1019_output,
    bool flag_write_RTCM_1020_output,
    bool flag_write_RTCM_1045_output,
    bool enable_rx_clock_correction)
```

Print RTCM messages.

The documentation for this class was generated from the following file:

- [rtcm\\_printer.h](#)

Collaboration diagram for rtcn\_t:



- int staid
- int stah
- int seqno
- int outtype
- gtime\_t time
- gtime\_t time\_s
- obs\_t obs

- [nav\\_t](#) `nav`
- [sta\\_t](#) `sta`
- [dgps\\_t](#) \* `dgps`
- [ssr\\_t](#) `ssr` [MAXSAT]
- `char` [msg](#) [128]
- `char` [msgtype](#) [256]
- `char` [msmtype](#) [6][128]
- `int` [obsflag](#)
- `int` [ephsat](#)
- `double` [cp](#) [MAXSAT][NFREQ+NEXOBS]
- `unsigned short` [lock](#) [MAXSAT][NFREQ+NEXOBS]
- `unsigned short` [loss](#) [MAXSAT][NFREQ+NEXOBS]
- [gtime\\_t](#) `lltime` [MAXSAT][NFREQ+NEXOBS]
- `int` [nbyte](#)
- `int` [nbit](#)
- `int` [len](#)
- `unsigned char` [buff](#) [1200]
- `unsigned int` [word](#)
- `unsigned int` [nmsg2](#) [100]
- `unsigned int` [nmsg3](#) [400]
- `char` [opt](#) [256]

### 10.338.1 Detailed Description

Definition at line 874 of file [rtklib.h](#).

### 10.338.2 Member Data Documentation

#### 10.338.2.1 buff

```
unsigned char rtkcm_t::buff[1200]
```

Definition at line 899 of file [rtklib.h](#).

#### 10.338.2.2 cp

```
double rtkcm_t::cp[MAXSAT][NFREQ+NEXOBS]
```

Definition at line 892 of file [rtklib.h](#).

#### 10.338.2.3 dgps

```
dgps\_t* rtkcm_t::dgps
```

Definition at line 885 of file [rtklib.h](#).

#### 10.338.2.4 ephsat

```
int rtkcm_t::ephsat
```

Definition at line 891 of file [rtklib.h](#).

#### 10.338.2.5 len

```
int rtkcm_t::len
```

Definition at line 898 of file [rtklib.h](#).

#### 10.338.2.6 lltime

```
gtime\_t rtkcm_t::lltime[MAXSAT][NFREQ+NEXOBS]
```

Definition at line 895 of file [rtklib.h](#).



### 10.338.2.7 lock

unsigned short rtcn\_t::lock[MAXSAT] [NFREQ+NEXOBS]  
Definition at line 893 of file [rtklib.h](#).

### 10.338.2.8 loss

unsigned short rtcn\_t::loss[MAXSAT] [NFREQ+NEXOBS]  
Definition at line 894 of file [rtklib.h](#).

### 10.338.2.9 msg

char rtcn\_t::msg[128]  
Definition at line 887 of file [rtklib.h](#).

### 10.338.2.10 msgtype

char rtcn\_t::msgtype[256]  
Definition at line 888 of file [rtklib.h](#).

### 10.338.2.11 msmttype

char rtcn\_t::msmttype[6][128]  
Definition at line 889 of file [rtklib.h](#).

### 10.338.2.12 nav

[nav\\_t](#) rtcn\_t::nav  
Definition at line 883 of file [rtklib.h](#).

### 10.338.2.13 nbit

int rtcn\_t::nbit  
Definition at line 897 of file [rtklib.h](#).

### 10.338.2.14 nbyte

int rtcn\_t::nbyte  
Definition at line 896 of file [rtklib.h](#).

### 10.338.2.15 nmsg2

unsigned int rtcn\_t::nmsg2[100]  
Definition at line 901 of file [rtklib.h](#).

### 10.338.2.16 nmsg3

unsigned int rtcn\_t::nmsg3[400]  
Definition at line 902 of file [rtklib.h](#).

### 10.338.2.17 obs

[obs\\_t](#) rtcn\_t::obs  
Definition at line 882 of file [rtklib.h](#).

### 10.338.2.18 obsflag

int rtcn\_t::obsflag  
Definition at line 890 of file [rtklib.h](#).

**10.338.2.19 opt**

```
char rtc_m_t::opt[256]
```

Definition at line 903 of file [rtklib.h](#).

**10.338.2.20 outtype**

```
int rtc_m_t::outtype
```

Definition at line 879 of file [rtklib.h](#).

**10.338.2.21 seqno**

```
int rtc_m_t::seqno
```

Definition at line 878 of file [rtklib.h](#).

**10.338.2.22 ssr**

```
ssr_t rtc_m_t::ssr[MAXSAT]
```

Definition at line 886 of file [rtklib.h](#).

**10.338.2.23 sta**

```
sta_t rtc_m_t::sta
```

Definition at line 884 of file [rtklib.h](#).

**10.338.2.24 stah**

```
int rtc_m_t::stah
```

Definition at line 877 of file [rtklib.h](#).

**10.338.2.25 staid**

```
int rtc_m_t::staid
```

Definition at line 876 of file [rtklib.h](#).

**10.338.2.26 time**

```
gtime_t rtc_m_t::time
```

Definition at line 880 of file [rtklib.h](#).

**10.338.2.27 time\_s**

```
gtime_t rtc_m_t::time_s
```

Definition at line 881 of file [rtklib.h](#).

**10.338.2.28 word**

```
unsigned int rtc_m_t::word
```

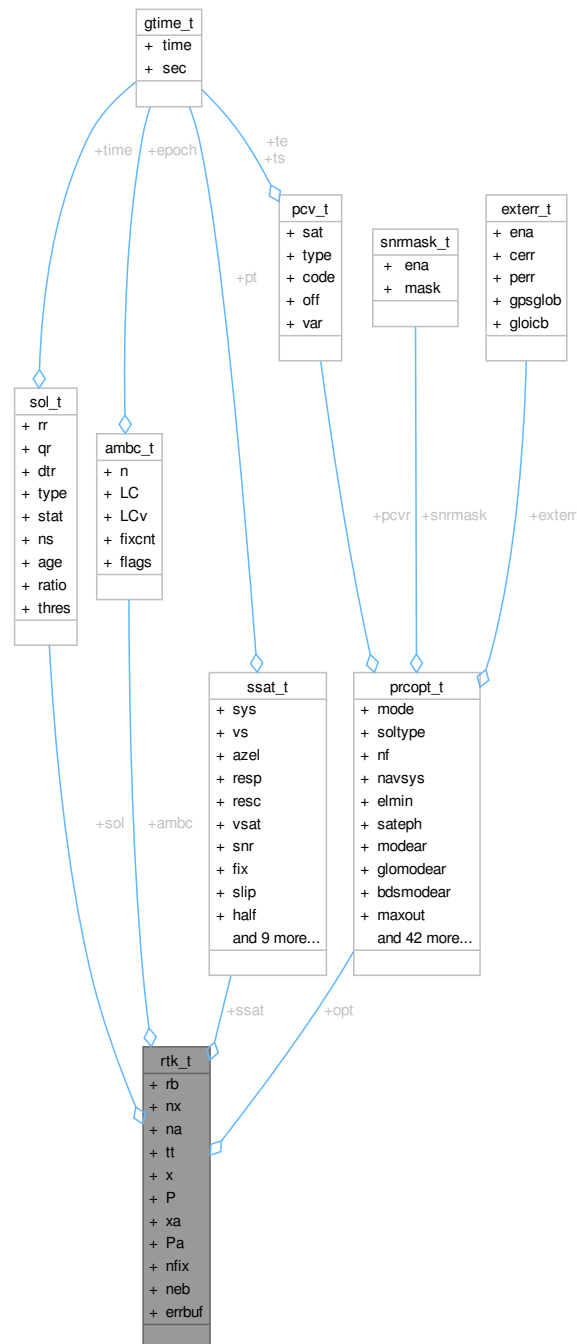
Definition at line 900 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.339 rtk\_t Struct Reference

Collaboration diagram for rtk\_t:



### Public Attributes

- [sol\\_t sol](#)
- double [rb](#) [6]
- int [nx](#)
- int [na](#)
- double [tt](#)

- double \* [x](#)
- double \* [P](#)
- double \* [xa](#)
- double \* [Pa](#)
- int [nfix](#)
- [ambc\\_t](#) [ambc](#) [MAXSAT]
- [ssat\\_t](#) [ssat](#) [MAXSAT]
- int [neb](#)
- char [errbuf](#) [MAXERRMSG]
- [prcopt\\_t](#) [opt](#)

### 10.339.1 Detailed Description

Definition at line [1066](#) of file [rtklib.h](#).

### 10.339.2 Member Data Documentation

#### 10.339.2.1 [ambc](#)

[ambc\\_t](#) [rtk\\_t::ambc](#) [MAXSAT]

Definition at line [1075](#) of file [rtklib.h](#).

#### 10.339.2.2 [errbuf](#)

char [rtk\\_t::errbuf](#) [MAXERRMSG]

Definition at line [1078](#) of file [rtklib.h](#).

#### 10.339.2.3 [na](#)

int [rtk\\_t::na](#)

Definition at line [1070](#) of file [rtklib.h](#).

#### 10.339.2.4 [neb](#)

int [rtk\\_t::neb](#)

Definition at line [1077](#) of file [rtklib.h](#).

#### 10.339.2.5 [nfix](#)

int [rtk\\_t::nfix](#)

Definition at line [1074](#) of file [rtklib.h](#).

#### 10.339.2.6 [nx](#)

int [rtk\\_t::nx](#)

Definition at line [1070](#) of file [rtklib.h](#).

#### 10.339.2.7 [opt](#)

[prcopt\\_t](#) [rtk\\_t::opt](#)

Definition at line [1079](#) of file [rtklib.h](#).

#### 10.339.2.8 [P](#)

double \* [rtk\\_t::P](#)

Definition at line [1072](#) of file [rtklib.h](#).

#### 10.339.2.9 Pa

double \* rtk\_t::Pa

Definition at line 1073 of file [rtklib.h](#).

#### 10.339.2.10 rb

double rtk\_t::rb[6]

Definition at line 1069 of file [rtklib.h](#).

#### 10.339.2.11 sol

[sol\\_t](#) rtk\_t::sol

Definition at line 1068 of file [rtklib.h](#).

#### 10.339.2.12 ssat

[ssat\\_t](#) rtk\_t::ssat [MAXSAT]

Definition at line 1076 of file [rtklib.h](#).

#### 10.339.2.13 tt

double rtk\_t::tt

Definition at line 1071 of file [rtklib.h](#).

#### 10.339.2.14 x

double\* rtk\_t::x

Definition at line 1072 of file [rtklib.h](#).

#### 10.339.2.15 xa

double\* rtk\_t::xa

Definition at line 1073 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

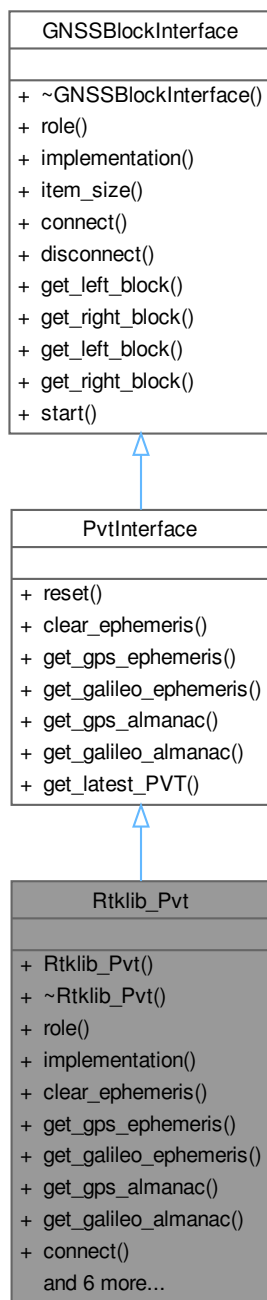
- [rtklib.h](#)

## 10.340 Rtklib\_Pvt Class Reference

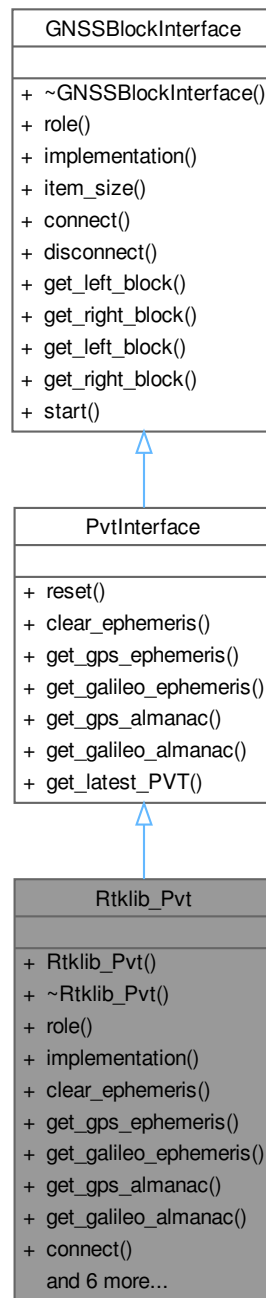
This class implements a [PvtInterface](#) for the RTKLIB PVT block.

```
#include <rtklib_pvt.h>
```

Inheritance diagram for Rtklib\_Pvt:



Collaboration diagram for Rtklib\_Pvt:



## Public Member Functions

- **Rtklib\_Pvt** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override
  - Returns "RTKLIB\_PVT".
- void [clear\\_ephemeris](#) () override

- `std::map< int, Gps\_Ephemeris > get_gps_ephemeris ()` const override
  - `std::map< int, Galileo\_Ephemeris > get_galileo_ephemeris ()` const override
  - `std::map< int, Gps\_Almanac > get_gps_almanac ()` const override
  - `std::map< int, Galileo\_Almanac > get_galileo_almanac ()` const override
  - `void connect (gr::top_block_sptr top_block)` override
  - `void disconnect (gr::top_block_sptr top_block)` override
  - `gr::basic_block_sptr get_left_block ()` override
  - `gr::basic_block_sptr get_right_block ()` override
  - `void reset ()` override
  - `size_t item_size ()` override
- All blocks must have an `item_size()` function implementation.*
- `bool get_latest_PVT (double *longitude_deg, double *latitude_deg, double *height_m, double *ground_speed_kmh, double *course_over_ground_deg, time_t *UTC_time)` override

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual `gr::basic_block_sptr get_left_block (int RF_channel)`
- virtual `gr::basic_block_sptr get_right_block (int RF_channel)`
- virtual `void start ()`

*Start the flow of samples if needed.*

### 10.340.1 Detailed Description

This class implements a [PvtInterface](#) for the RTKLIB PVT block.

Global configuration options used:

`GNSS-SDR.pre_2009_file` - flag indicating a file older than 2009 rollover should be processed (false) `GNSS-SDR.observable_interval_ms` - (20)

It supports the following configuration options:

`.dump` - (false) `.dump_filename` - ("/pvt.dat") `.dump_mat` - (true) `.rtk_trace_level` - debug level for the RTKLIB methods (0)

`.output_rate_ms` - (500) Note that the actual rate is the least common multiple of this value and `GNSS-SDR.observable_interval_ms` `.display_rate_ms` - (500)

`.flag_nmea_tty_port` - (false) `.nmea_dump_filename` - ("/nmea\_pvt.nmea") `.nmea_dump_devname` - ("/dev/tty1")

`.rinex_version` - (3) overridden by `-RINEX_version=n.nn` command line argument `.rinexobs_rate_ms` - rate at which RINEX observations are written (1000). Note that the actual rate is the least common multiple of this value and `.output_rate_ms` `.rinex_name` - (-RINEX\_name command-line argument)

`.flag_rtcmtty_port` - (false) `.rtcm_dump_devname` - ("/dev/pts/1") `.flag_rtcmtcp_server` - (false) `.rtcm_tcp_port` - (2101) `.rtcm_station_id` - (1234) Output rates ... all values are LCM with the computed output rate (above) `.rtcm_MT1019_rate_ms` - (5000) `.rtcm_MT1020_rate_ms` - (5000) `.rtcm_MT1045_rate_ms` - (5000) `.rtcm_MSM_rate_ms` - (1000) `.rtcm_MT1077_rate_ms` - (.rtcm\_MSM\_rate\_ms) `.rtcm_MT1087_rate_ms` - (.rtcm\_MSM\_rate\_ms) `.rtcm_MT1097_rate_ms` - (.rtcm\_MSM\_rate\_ms)

`.kml_rate_ms` - (1000) `.gpx_rate_ms` - (1000) `.geojson_rate_ms` - (1000) `.nmea_rate_ms` - (1000)

`.positioning_mode` - the RTKLIB positioning mode. ("Single") Supported values are "Single", "Static", "Kinematic", "PPP\_Static" and "PPP\_Kinematic". Unsupported modes include DGPS/DGNSS, Moving Baseline, Fixed, and PPP-fixed `.num_bands` - number of frequencies to use, between 1 and 3. Default is based on the channels configured `.elevation_mask` - (15.0). Value must be in the range [0,90.0] `.dynamics_model` - (0) 0:none, 1:velocity, 2:acceleration

`.iono_model` - ("OFF"). Supported values are "OFF", "Broadcast", "SBAS", "Iono-Free-LC", "Estimate\_STEC", "IONEX". Unsupported values include QZSS broadcast, QZSS LEX, and SLANT TEC. `.trop_model` - ("OFF"). Supported values are "OFF", "Saastamoinen", "SBAS", "Estimate\_ZTD", and "Estimate\_ZTD\_Grad". Unsupported values include ZTD correction and ZTD+grad correction `.phwindup` - phase windup correction for PPP modes (0)

`.reject_GPS_IIA` - whether the GPS Block IIA satellites in eclipse are excluded (0). Only applies in PPP-\* modes `.raim_fde` - whether RAIM (receiver autonomous integrity monitoring) FDE (fault detection and exclusion) is enabled (0)

`.earth_tide` - (0) `.navigation_system` - mask of navigation systems to use. Default based on configured channels 0x01:GPS, 0x02:SBAS, 0x04:GLONASS, 0x08:Galileo, 0x10:QZSS, 0x20:BeiDou, 0x40:IRNS, 0x80:LEO

`.AR_GPS` - Ambiguity Resolution mode for GPS ("Continuous"). Supported values are "OFF", "Continuous", "Instantaneous", "Fix-and-Hold", "PPP-AR". Unsupported values include PPP-AR ILS, WLNL, and TCAR. `.AR_GLO`



- Ambiguity Resolution mode for GLONASS (1). Value must be in the range [0,3]. (0:off,1:on,2:auto cal,3:ext cal)  
 .AR\_DBS - Ambiguity Resolution Mode for BeiDou (1). Value must be in the range [0,1]. (0:off,1:on) .min\_ratio\_↵  
 to\_fix\_ambiguity - (3.0) .min\_lock\_to\_fix\_ambiguity - (0) .min\_elevation\_to\_fix\_ambiguity - minimum elevation (deg)  
 to fix integer ambiguity (0.0) .outage\_reset\_ambiguity - (5) .slip\_threshold - (0.05) .threshold\_reject\_gdop - if GDOP  
 is over this value, the observable is excluded (30.0) .threshold\_reject\_innovation - if innovation is over this value, the  
 observable is excluded (30.0) .number\_filter\_iter - number of iterations for the estimation filter (1) .bias\_0 - (30.0)  
 .iono\_0 - (0.03) .trop\_0 - (0.3) .sigma\_bias - process noise stddev of carrier-phase bias(ambiguity)(cycle/sqrt(s)) (1e-  
 4) .sigma\_iono - process noise stddev of vertical ionospheric delay per 10km baseline (m/sqrt(s)) (1e-3) .sigma\_↵  
 \_trop - process noise stddev of zenith tropospheric delay (m/sqrt(s)) (1e-4) .sigma\_acch - process noise stddev of  
 the receiver acceleration horizontal component (m/s2/sqrt(s)) (1e-2) .sigma\_accv - process noise stddev of the re-  
 ceiver acceleration vertical component (m/s2/sqrt(s)) (1e-2) .sigma\_pos - (0.0) .code\_phase\_error\_ratio\_l1 - (100.0)  
 .code\_phase\_error\_ratio\_l2 - (100.0) .code\_phase\_error\_ratio\_l5 - (100.0) .carrier\_phase\_error\_factor\_a - (0.003)  
 .carrier\_phase\_error\_factor\_b - (0.003)  
 .output\_enabled - (true) .rinex\_output\_enabled - (.output\_enabled) .gpx\_output\_enabled - (.output\_enabled)  
 .geojson\_output\_enabled - (.output\_enabled) .kml\_output\_enabled - (.output\_enabled) .xml\_output\_enabled -  
 (.output\_enabled) .nmea\_output\_enabled - (.output\_enabled) .rtcm\_output\_enabled - (false)  
 .output\_path - directory to which output files are written (".") .rinex\_output\_path - (.output\_path) .gpx\_output\_↵  
 path - (.output\_path) .geojson\_output\_path - (.output\_path) .kml\_output\_path - (.output\_path) .xml\_output\_path -  
 (.output\_path) .nmea\_output\_path - (.output\_path) .rtcm\_output\_path - (.output\_path)  
 .enable\_monitor - enable the PVT monitor (false) .monitor\_client\_addresses - ("127.0.0.1") .monitor\_udp\_port - DO  
 NOT USE THE DEFAULT (1234) .enable\_protobuf - serialize using protocol buffers (true). Monitor.enable\_protobuf  
 if true, sets this to true  
 .enable\_monitor\_ephemeris - enable the ephemeris monitor (false) .monitor\_ephemeris\_client\_addresses -  
 ("127.0.0.1") .monitor\_ephemeris\_udp\_port - DO NOT USE THE DEFAULT (1234)  
 .show\_local\_time\_zone - (false) .enable\_rx\_clock\_correction - (false) .max\_clock\_offset\_ms - (40)  
 Definition at line 173 of file [rtklib\\_pvt.h](#).

## 10.340.2 Member Function Documentation

### 10.340.2.1 clear\_ephemeris()

```
void Rtklib_Pvt::clear_ephemeris () [override], [virtual]
```

Implements [PvtInterface](#).

### 10.340.2.2 connect()

```
void Rtklib_Pvt::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.340.2.3 disconnect()

```
void Rtklib_Pvt::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.340.2.4 get\_galileo\_almanac()

```
std::map< int, Galileo_Almanac > Rtklib_Pvt::get_galileo_almanac () const [override], [virtual]
```

Implements [PvtInterface](#).

### 10.340.2.5 get\_galileo\_ephemeris()

```
std::map< int, Galileo_Ephemeris > Rtklib_Pvt::get_galileo_ephemeris () const [override],
[virtual]
```

Implements [PvtInterface](#).

### 10.340.2.6 get\_gps\_almanac()

```
std::map< int, Gps_Almanac > Rtklib_Pvt::get_gps_almanac () const [override], [virtual]
```

Implements [PvtInterface](#).

#### 10.340.2.7 `get_gps_ephemeris()`

```
std::map< int, Gps\_Ephemeris > Rtklib_Pvt::get_gps_ephemeris () const [override], [virtual]
```

Implements [PvtInterface](#).

#### 10.340.2.8 `get_latest_PVT()`

```
bool Rtklib_Pvt::get_latest_PVT (
    double * longitude_deg,
    double * latitude_deg,
    double * height_m,
    double * ground_speed_kmh,
    double * course_over_ground_deg,
    time_t * UTC_time) [override], [virtual]
```

Implements [PvtInterface](#).

#### 10.340.2.9 `get_left_block()`

```
gr::basic_block_sptr Rtklib_Pvt::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.340.2.10 `get_right_block()`

```
gr::basic_block_sptr Rtklib_Pvt::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.340.2.11 `implementation()`

```
std::string Rtklib_Pvt::implementation () [inline], [override], [virtual]
```

Returns "RTKLIB\_PVT".

Implements [GNSSBlockInterface](#).

Definition at line 189 of file [rtklib\\_pvt.h](#).

#### 10.340.2.12 `item_size()`

```
size_t Rtklib_Pvt::item_size () [inline], [override], [virtual]
```

All blocks must have an `item_size()` function implementation.

Implements [GNSSBlockInterface](#).

Definition at line 211 of file [rtklib\\_pvt.h](#).

#### 10.340.2.13 `reset()`

```
void Rtklib_Pvt::reset () [inline], [override], [virtual]
```

Implements [PvtInterface](#).

Definition at line 205 of file [rtklib\\_pvt.h](#).

#### 10.340.2.14 `role()`

```
std::string Rtklib_Pvt::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 183 of file [rtklib\\_pvt.h](#).

The documentation for this class was generated from the following file:

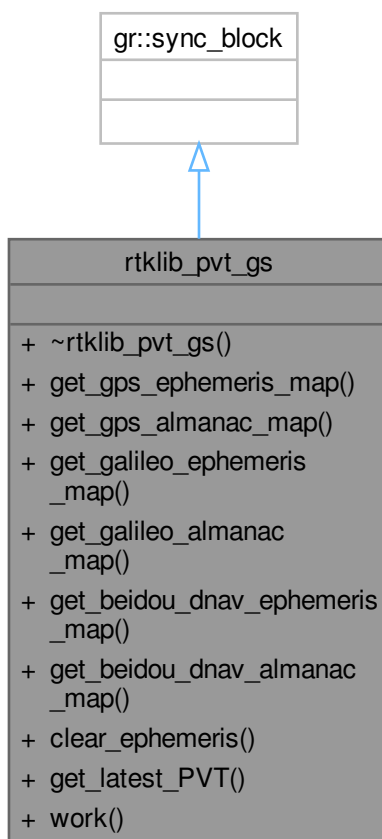
- [rtklib\\_pvt.h](#)

## 10.341 rtklib\_pvt\_gs Class Reference

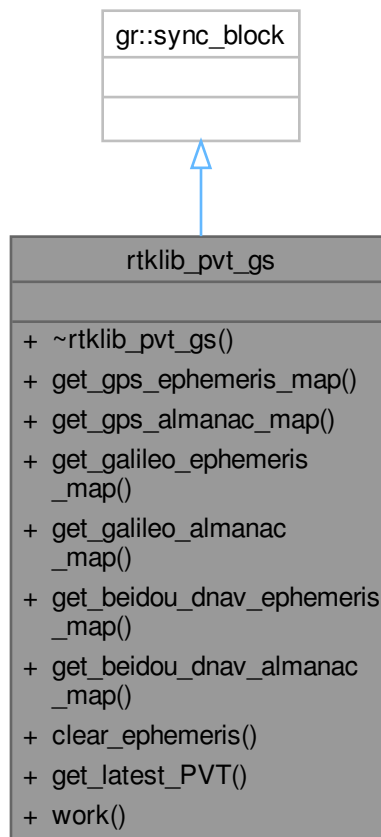
This class implements a block that computes the PVT solution using the RTKLIB integrated library.

```
#include <rtklib_pvt_gs.h>
```

Inheritance diagram for rtklib\_pvt\_gs:



Collaboration diagram for rtklib\_pvt\_gs:



## Public Member Functions

- `~rtklib_pvt_gs ()`  
*Default destructor.*
- `std::map< int, Gps\_Ephemeris > get_gps_ephemeris_map ()` const  
*Get latest set of GPS ephemeris from PVT block.*
- `std::map< int, Gps\_Almanac > get_gps_almanac_map ()` const  
*Get latest set of GPS almanac from PVT block.*
- `std::map< int, Galileo\_Ephemeris > get_galileo_ephemeris_map ()` const  
*Get latest set of Galileo ephemeris from PVT block.*
- `std::map< int, Galileo\_Almanac > get_galileo_almanac_map ()` const  
*Get latest set of Galileo almanac from PVT block.*
- `std::map< int, Beidou\_Dnav\_Ephemeris > get_beidou_dnav_ephemeris_map ()` const  
*Get latest set of BeiDou DNAV ephemeris from PVT block.*
- `std::map< int, Beidou\_Dnav\_Almanac > get_beidou_dnav_almanac_map ()` const  
*Get latest set of BeiDou DNAV almanac from PVT block.*
- `void clear_ephemeris ()`  
*Clear all ephemeris information and the almanacs for GPS and Galileo.*
- `bool get_latest_PVT (double *longitude_deg, double *latitude_deg, double *height_m, double *ground_speed_kmh, double *course_over_ground_deg, time_t *UTC_time)` const

*Get the latest Position WGS84 [deg], Ground Velocity, Course over Ground, and UTC Time, if available.*

- int [work](#) (int noutput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)  
*PVT Signal Processing.*

## Friends

- rtklib\_pvt\_gs\_sptr [rtklib\\_make\\_pvt\\_gs](#) (uint32\_t nchannels, const [Pvt\\_Conf](#) &conf\_, const [rtk\\_t](#) &rtk)

## 10.341.1 Detailed Description

This class implements a block that computes the PVT solution using the RTKLIB integrated library.

Definition at line 82 of file [rtklib\\_pvt\\_gs.h](#).

## 10.341.2 Constructor & Destructor Documentation

### 10.341.2.1 ~rtklib\_pvt\_gs()

```
rtklib_pvt_gs::~~rtklib_pvt_gs ()
```

Default destructor.

## 10.341.3 Member Function Documentation

### 10.341.3.1 clear\_ephemeris()

```
void rtklib_pvt_gs::clear_ephemeris ()
```

Clear all ephemeris information and the almanacs for GPS and Galileo.

### 10.341.3.2 get\_beidou\_dnav\_almanac\_map()

```
std::map< int, Beidou\_Dnav\_Almanac > rtklib_pvt_gs::get_beidou_dnav_almanac_map () const
```

Get latest set of BeiDou DNAV almanac from PVT block.

### 10.341.3.3 get\_beidou\_dnav\_ephemeris\_map()

```
std::map< int, Beidou\_Dnav\_Ephemeris > rtklib_pvt_gs::get_beidou_dnav_ephemeris_map () const
```

Get latest set of BeiDou DNAV ephemeris from PVT block.

### 10.341.3.4 get\_galileo\_almanac\_map()

```
std::map< int, Galileo\_Almanac > rtklib_pvt_gs::get_galileo_almanac_map () const
```

Get latest set of Galileo almanac from PVT block.

### 10.341.3.5 get\_galileo\_ephemeris\_map()

```
std::map< int, Galileo\_Ephemeris > rtklib_pvt_gs::get_galileo_ephemeris_map () const
```

Get latest set of Galileo ephemeris from PVT block.

### 10.341.3.6 get\_gps\_almanac\_map()

```
std::map< int, Gps\_Almanac > rtklib_pvt_gs::get_gps_almanac_map () const
```

Get latest set of GPS almanac from PVT block.

### 10.341.3.7 get\_gps\_ephemeris\_map()

```
std::map< int, Gps\_Ephemeris > rtklib_pvt_gs::get_gps_ephemeris_map () const
```

Get latest set of GPS ephemeris from PVT block.

#### 10.341.3.8 `get_latest_PVT()`

```
bool rtklib_pvt_gs::get_latest_PVT (  
    double * longitude_deg,  
    double * latitude_deg,  
    double * height_m,  
    double * ground_speed_kmh,  
    double * course_over_ground_deg,  
    time_t * UTC_time) const
```

Get the latest Position WGS84 [deg], Ground Velocity, Course over Ground, and UTC Time, if available.

#### 10.341.3.9 `work()`

```
int rtklib_pvt_gs::work (  
    int noutput_items,  
    gr_vector_const_void_star & input_items,  
    gr_vector_void_star & output_items)
```

PVT Signal Processing.

The documentation for this class was generated from the following file:

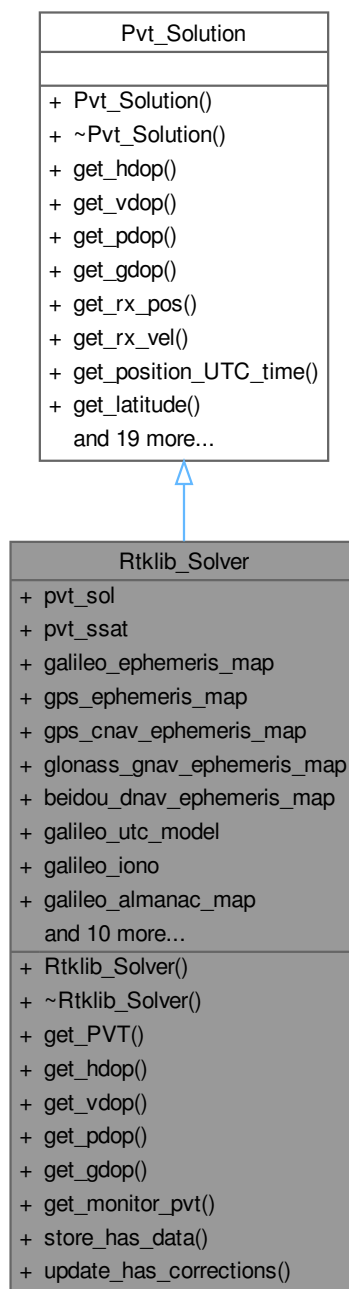
- [rtklib\\_pvt\\_gs.h](#)

### 10.342 Rtklib\_Solver Class Reference

This class implements a PVT solution based on RTKLIB.

```
#include <rtklib_solver.h>
```

Inheritance diagram for Rtklib\_Solver:







- double [get\\_speed\\_over\\_ground](#) () const  
*Get RX speed over ground [m/s].*
- double [get\\_course\\_over\\_ground](#) () const  
*Get RX course over ground [deg].*
- int [get\\_num\\_valid\\_observations](#) () const  
*Get the number of valid pseudorange observations (valid satellites)*
- bool [is\\_pre\\_2009](#) () const
- bool [is\\_valid\\_position](#) () const
- void [set\\_rx\\_pos](#) (const std::array< double, 3 > &pos)  
*Set position: X, Y, Z in Cartesian ECEF coordinates [m].*
- void [set\\_rx\\_vel](#) (const std::array< double, 3 > &vel)  
*Set velocity: East [m/s], North [m/s], Up [m/s].*
- void [set\\_position\\_UTC\\_time](#) (const boost::posix\_time::ptime &pt)
- void [set\\_time\\_offset\\_s](#) (double offset)  
*Set RX time offset [s].*
- void [set\\_clock\\_drift\\_ppm](#) (double clock\_drift\_ppm)  
*Set the Rx clock drift [ppm].*
- void [set\\_speed\\_over\\_ground](#) (double speed\_m\_s)  
*Set RX speed over ground [m/s].*
- void [set\\_course\\_over\\_ground](#) (double cog\_deg)  
*Set RX course over ground [deg].*
- void [set\\_valid\\_position](#) (bool is\_valid)
- void [set\\_num\\_valid\\_observations](#) (int num)  
*Set the number of valid pseudorange observations (valid satellites)*
- void [set\\_pre\\_2009\\_file](#) (bool pre\_2009\_file)  
*Flag for the week rollover computation in post processing mode for signals older than 2009.*

### Public Attributes

- [sol\\_t](#) pvt\_sol {}
- std::array< [ssat\\_t](#), MAXSAT > pvt\_ssat {}
- std::map< int, [Galileo\\_Ephemeris](#) > [galileo\\_ephemeris\\_map](#)  
*Map storing new Galileo\_Ephemeris.*
- std::map< int, [Gps\\_Ephemeris](#) > [gps\\_ephemeris\\_map](#)  
*Map storing new GPS\_Ephemeris.*
- std::map< int, [Gps\\_CNAV\\_Ephemeris](#) > [gps\\_cnav\\_ephemeris\\_map](#)  
*Map storing new GPS\_CNAV\_Ephemeris.*
- std::map< int, [Glonass\\_Gnav\\_Ephemeris](#) > [glonass\\_gnav\\_ephemeris\\_map](#)  
*Map storing new GLONASS GNAV Ephemeris.*
- std::map< int, [Beidou\\_Dnav\\_Ephemeris](#) > [beidou\\_dnav\\_ephemeris\\_map](#)  
*Map storing new BeiDou DNAV Ephemeris.*
- [Galileo\\_Utc\\_Model](#) [galileo\\_utc\\_model](#)
- [Galileo\\_Iono](#) [galileo\\_iono](#)
- std::map< int, [Galileo\\_Almanac](#) > [galileo\\_almanac\\_map](#)
- [Gps\\_Utc\\_Model](#) [gps\\_utc\\_model](#)
- [Gps\\_Iono](#) [gps\\_iono](#)
- std::map< int, [Gps\\_Almanac](#) > [gps\\_almanac\\_map](#)
- [Gps\\_CNAV\\_Iono](#) [gps\\_cnav\\_iono](#)
- [Gps\\_CNAV\\_Utc\\_Model](#) [gps\\_cnav\\_utc\\_model](#)
- [Glonass\\_Gnav\\_Utc\\_Model](#) [glonass\\_gnav\\_utc\\_model](#)  
*Map storing GLONASS GNAV UTC Model.*
- [Glonass\\_Gnav\\_Almanac](#) [glonass\\_gnav\\_almanac](#)

*Map storing GLONASS GNAV Almanac Model.*

- [Beidou\\_Dnav\\_Utc\\_Model](#) `beidou_dnav_utc_model`
- [Beidou\\_Dnav\\_Iono](#) `beidou_dnav_iono`
- `std::map< int, Beidou\_Dnav\_Almanac >` `beidou_dnav_almanac_map`

### 10.342.1 Detailed Description

This class implements a PVT solution based on RTKLIB.

Definition at line 81 of file [rtklib\\_solver.h](#).

### 10.342.2 Member Function Documentation

#### 10.342.2.1 `get_gdop()`

```
double Rtklib_Solver::get_gdop () const [override], [virtual]
```

Implements [Pvt\\_Solution](#).

#### 10.342.2.2 `get_hdop()`

```
double Rtklib_Solver::get_hdop () const [override], [virtual]
```

Implements [Pvt\\_Solution](#).

#### 10.342.2.3 `get_pdop()`

```
double Rtklib_Solver::get_pdop () const [override], [virtual]
```

Implements [Pvt\\_Solution](#).

#### 10.342.2.4 `get_vdop()`

```
double Rtklib_Solver::get_vdop () const [override], [virtual]
```

Implements [Pvt\\_Solution](#).

### 10.342.3 Member Data Documentation

#### 10.342.3.1 `beidou_dnav_almanac_map`

```
std::map<int, Beidou\_Dnav\_Almanac> Rtklib_Solver::beidou_dnav_almanac_map
```

Definition at line 128 of file [rtklib\\_solver.h](#).

#### 10.342.3.2 `beidou_dnav_ephemeris_map`

```
std::map<int, Beidou\_Dnav\_Ephemeris> Rtklib_Solver::beidou_dnav_ephemeris_map
```

Map storing new BeiDou DNAV Ephemeris.

Definition at line 110 of file [rtklib\\_solver.h](#).

#### 10.342.3.3 `beidou_dnav_iono`

```
Beidou\_Dnav\_Iono Rtklib_Solver::beidou_dnav_iono
```

Definition at line 127 of file [rtklib\\_solver.h](#).

#### 10.342.3.4 `beidou_dnav_utc_model`

```
Beidou\_Dnav\_Utc\_Model Rtklib_Solver::beidou_dnav_utc_model
```

Definition at line 126 of file [rtklib\\_solver.h](#).

#### 10.342.3.5 `galileo_almanac_map`

```
std::map<int, Galileo\_Almanac> Rtklib_Solver::galileo_almanac_map
```

Definition at line 114 of file [rtklib\\_solver.h](#).

### 10.342.3.6 galileo\_ephemeris\_map

`std::map<int, Galileo\_Ephemeris> Rtklib_Solver::galileo_ephemeris_map`

Map storing new [Galileo\\_Ephemeris](#).

Definition at line 106 of file [rtklib\\_solver.h](#).

### 10.342.3.7 galileo\_iono

`Galileo\_Iono Rtklib_Solver::galileo_iono`

Definition at line 113 of file [rtklib\\_solver.h](#).

### 10.342.3.8 galileo\_utc\_model

`Galileo\_Utc\_Model Rtklib_Solver::galileo_utc_model`

Definition at line 112 of file [rtklib\\_solver.h](#).

### 10.342.3.9 glonass\_gnav\_almanac

`Glonass\_Gnav\_Almanac Rtklib_Solver::glonass_gnav_almanac`

Map storing GLONASS GNAV Almanac Model.

Definition at line 124 of file [rtklib\\_solver.h](#).

### 10.342.3.10 glonass\_gnav\_ephemeris\_map

`std::map<int, Glonass\_Gnav\_Ephemeris> Rtklib_Solver::glonass_gnav_ephemeris_map`

Map storing new GLONASS GNAV Ephemeris.

Definition at line 109 of file [rtklib\\_solver.h](#).

### 10.342.3.11 glonass\_gnav\_utc\_model

`Glonass\_Gnav\_Utc\_Model Rtklib_Solver::glonass_gnav_utc_model`

Map storing GLONASS GNAV UTC Model.

Definition at line 123 of file [rtklib\\_solver.h](#).

### 10.342.3.12 gps\_almanac\_map

`std::map<int, Gps\_Almanac> Rtklib_Solver::gps_almanac_map`

Definition at line 118 of file [rtklib\\_solver.h](#).

### 10.342.3.13 gps\_cnav\_ephemeris\_map

`std::map<int, Gps\_CNAV\_Ephemeris> Rtklib_Solver::gps_cnav_ephemeris_map`

Map storing new GPS\_CNAV\_Ephemeris.

Definition at line 108 of file [rtklib\\_solver.h](#).

### 10.342.3.14 gps\_cnav\_iono

`Gps\_CNAV\_Iono Rtklib_Solver::gps_cnav_iono`

Definition at line 120 of file [rtklib\\_solver.h](#).

### 10.342.3.15 gps\_cnav\_utc\_model

`Gps\_CNAV\_Utc\_Model Rtklib_Solver::gps_cnav_utc_model`

Definition at line 121 of file [rtklib\\_solver.h](#).

### 10.342.3.16 gps\_ephemeris\_map

`std::map<int, Gps\_Ephemeris> Rtklib_Solver::gps_ephemeris_map`

Map storing new GPS\_Ephemeris.

Definition at line 107 of file [rtklib\\_solver.h](#).

**10.342.3.17   gps\_iono**

`Gps_Iono` `Rtklib_Solver::gps_iono`  
Definition at line 117 of file [rtklib\\_solver.h](#).

**10.342.3.18   gps\_utc\_model**

`Gps_Utc_Model` `Rtklib_Solver::gps_utc_model`  
Definition at line 116 of file [rtklib\\_solver.h](#).

**10.342.3.19   pvt\_sol**

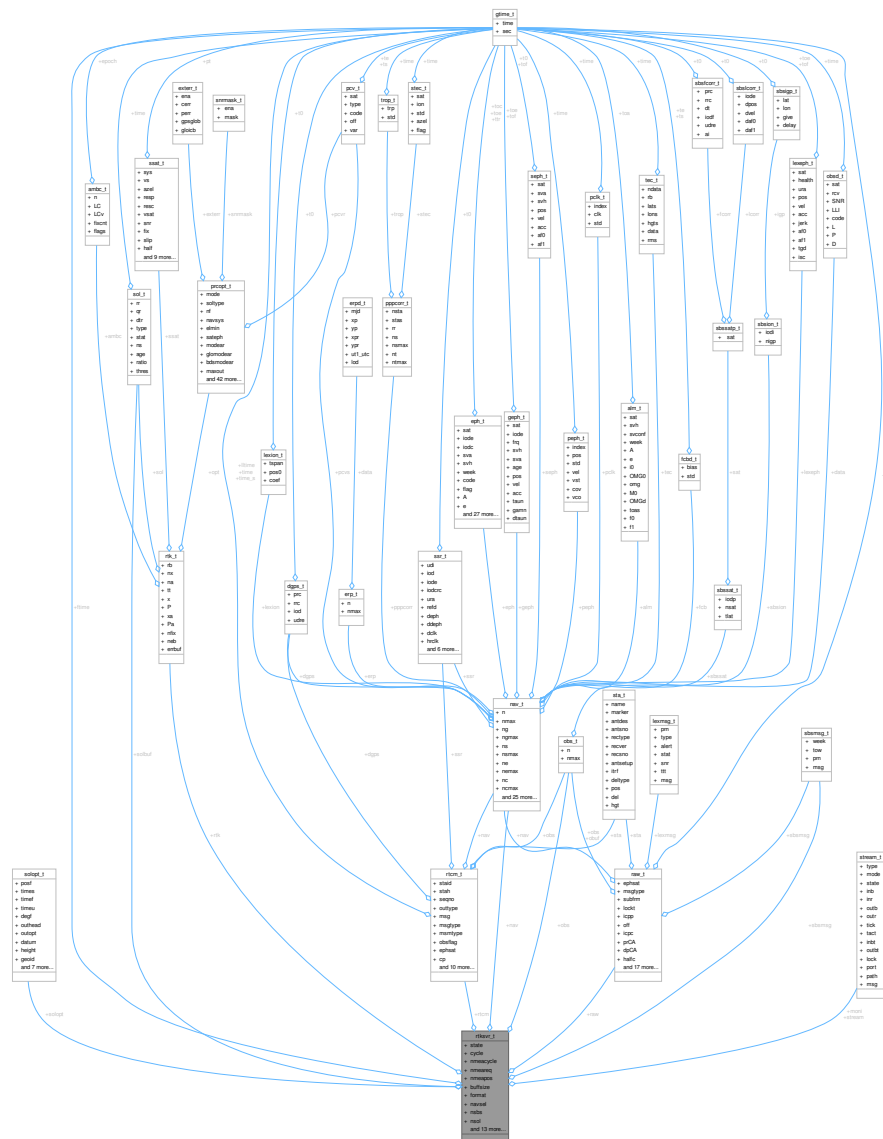
`sol_t` `Rtklib_Solver::pvt_sol` {}  
Definition at line 103 of file [rtklib\\_solver.h](#).

**10.342.3.20   pvt\_ssat**

`std::array<ssat_t, MAXSAT>` `Rtklib_Solver::pvt_ssat` {}  
Definition at line 104 of file [rtklib\\_solver.h](#).  
The documentation for this class was generated from the following file:

- [rtklib\\_solver.h](#)

Collaboration diagram for rtksvr\_t:



- int `state`
- int `cycle`
- int `nmeacycle`
- int `nmeareq`
- double `nmeapos` [3]
- int `buffsize`
- int `format` [3]
- `solo_t` `solo_t` [2]
- int `navsel`
- int `nsbs`
- int `nsol`
- `rtk_t` `rtk`
- int `nb` [3]

- int [nsb](#) [2]
- int [npb](#) [3]
- unsigned char \* [buff](#) [3]
- unsigned char \* [sbuf](#) [2]
- unsigned char \* [pbuf](#) [3]
- [sol\\_t](#) [solbuf](#) [MAXSOLBUF]
- unsigned int [nmsg](#) [3][10]
- [raw\\_t](#) [raw](#) [3]
- [rtcm\\_t](#) [rtcm](#) [3]
- [gtime\\_t](#) [ftime](#) [3]
- char [files](#) [3][MAXSTRPATH]
- [obs\\_t](#) [obs](#) [3][MAXOBSBUF]
- [nav\\_t](#) [nav](#)
- [sbsmsg\\_t](#) [sbsmsg](#) [MAXSBSMSG]
- [stream\\_t](#) [stream](#) [8]
- [stream\\_t](#) \* [moni](#)
- unsigned int [tick](#)
- pthread\_t [thread](#)
- int [cputime](#)
- int [prcout](#)
- lock\_t [lock](#)

### 10.343.1 Detailed Description

Definition at line [1238](#) of file [rtklib.h](#).

### 10.343.2 Member Data Documentation

#### 10.343.2.1 buff

```
unsigned char* rtksvr_t::buff[3]
```

Definition at line [1255](#) of file [rtklib.h](#).

#### 10.343.2.2 buffsize

```
int rtksvr_t::buffsize
```

Definition at line [1245](#) of file [rtklib.h](#).

#### 10.343.2.3 cputime

```
int rtksvr_t::cputime
```

Definition at line [1271](#) of file [rtklib.h](#).

#### 10.343.2.4 cycle

```
int rtksvr_t::cycle
```

Definition at line [1241](#) of file [rtklib.h](#).

#### 10.343.2.5 files

```
char rtksvr_t::files[3][MAXSTRPATH]
```

Definition at line [1263](#) of file [rtklib.h](#).

#### 10.343.2.6 format

```
int rtksvr_t::format[3]
```

Definition at line [1246](#) of file [rtklib.h](#).

### 10.343.2.7 ftime

`ptime_t rtksvr_t::ftime[3]`

Definition at line 1262 of file [rtklib.h](#).

### 10.343.2.8 lock

`lock_t rtksvr_t::lock`

Definition at line 1273 of file [rtklib.h](#).

### 10.343.2.9 moni

`stream_t* rtksvr_t::moni`

Definition at line 1268 of file [rtklib.h](#).

### 10.343.2.10 nav

`nav_t rtksvr_t::nav`

Definition at line 1265 of file [rtklib.h](#).

### 10.343.2.11 navsel

`int rtksvr_t::navsel`

Definition at line 1248 of file [rtklib.h](#).

### 10.343.2.12 nb

`int rtksvr_t::nb[3]`

Definition at line 1252 of file [rtklib.h](#).

### 10.343.2.13 nmeacycle

`int rtksvr_t::nmeacycle`

Definition at line 1242 of file [rtklib.h](#).

### 10.343.2.14 nmeapos

`double rtksvr_t::nmeapos[3]`

Definition at line 1244 of file [rtklib.h](#).

### 10.343.2.15 nmeareq

`int rtksvr_t::nmeareq`

Definition at line 1243 of file [rtklib.h](#).

### 10.343.2.16 nmsg

`unsigned int rtksvr_t::nmsg[3][10]`

Definition at line 1259 of file [rtklib.h](#).

### 10.343.2.17 npb

`int rtksvr_t::npb[3]`

Definition at line 1254 of file [rtklib.h](#).

### 10.343.2.18 nsb

`int rtksvr_t::nsb[2]`

Definition at line 1253 of file [rtklib.h](#).

**10.343.2.19 nsbs**

```
int rtksvr_t::nsbs
```

Definition at line 1249 of file [rtklib.h](#).

**10.343.2.20 nsol**

```
int rtksvr_t::nsol
```

Definition at line 1250 of file [rtklib.h](#).

**10.343.2.21 obs**

```
obs_t rtksvr_t::obs[3][MAXOBSBUF]
```

Definition at line 1264 of file [rtklib.h](#).

**10.343.2.22 pbuf**

```
unsigned char* rtksvr_t::pbuf[3]
```

Definition at line 1257 of file [rtklib.h](#).

**10.343.2.23 prcout**

```
int rtksvr_t::prcout
```

Definition at line 1272 of file [rtklib.h](#).

**10.343.2.24 raw**

```
raw_t rtksvr_t::raw[3]
```

Definition at line 1260 of file [rtklib.h](#).

**10.343.2.25 rtcn**

```
rtcn_t rtksvr_t::rtcn[3]
```

Definition at line 1261 of file [rtklib.h](#).

**10.343.2.26 rtk**

```
rtk_t rtksvr_t::rtk
```

Definition at line 1251 of file [rtklib.h](#).

**10.343.2.27 sbsmsg**

```
sbsmsg_t rtksvr_t::sbsmsg[MAXSBSMSG]
```

Definition at line 1266 of file [rtklib.h](#).

**10.343.2.28 sbuf**

```
unsigned char* rtksvr_t::sbuf[2]
```

Definition at line 1256 of file [rtklib.h](#).

**10.343.2.29 solbuf**

```
sol_t rtksvr_t::solbuf[MAXSOLBUF]
```

Definition at line 1258 of file [rtklib.h](#).

**10.343.2.30 solopt**

```
solopt_t rtksvr_t::solopt[2]
```

Definition at line 1247 of file [rtklib.h](#).



**10.343.2.31 state**

```
int rtksvr_t::state
```

Definition at line 1240 of file [rtklib.h](#).

**10.343.2.32 stream**

```
stream_t rtksvr_t::stream[8]
```

Definition at line 1267 of file [rtklib.h](#).

**10.343.2.33 thread**

```
pthread_t rtksvr_t::thread
```

Definition at line 1270 of file [rtklib.h](#).

**10.343.2.34 tick**

```
unsigned int rtksvr_t::tick
```

Definition at line 1269 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

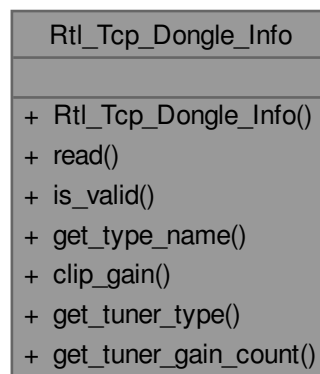
- [rtklib.h](#)

**10.344 Rtl\_Tcp\_Dongle\_Info Class Reference**

This class represents the dongle information which is sent by rtl\_tcp.

```
#include <rtl_tcp_dongle_info.h>
```

Collaboration diagram for Rtl\_Tcp\_Dongle\_Info:

**Public Types**

- enum {  
 TUNER\_UNKNOWN = 0 , TUNER\_E4000 , TUNER\_FC0012 , TUNER\_FC0013 ,  
 TUNER\_FC2580 , TUNER\_R820T , TUNER\_R828D }

**Public Member Functions**

- boost::system::error\_code **read** (boost::asio::ip::tcp::socket &socket)
- bool **is\_valid** () const

- const char \* **get\_type\_name** () const
- double **clip\_gain** (int gain) const
- uint32\_t **get\_tuner\_type** () const
- uint32\_t **get\_tuner\_gain\_count** () const

### 10.344.1 Detailed Description

This class represents the dongle information which is sent by rtl\_tcp.  
Definition at line 35 of file [rtl\\_tcp\\_dongle\\_info.h](#).

### 10.344.2 Member Enumeration Documentation

#### 10.344.2.1 anonymous enum

anonymous enum

Definition at line 38 of file [rtl\\_tcp\\_dongle\\_info.h](#).

### 10.344.3 Member Function Documentation

#### 10.344.3.1 get\_tuner\_gain\_count()

uint32\_t Rtl\_Tcp\_Dongle\_Info::get\_tuner\_gain\_count () const [inline]

Definition at line 64 of file [rtl\\_tcp\\_dongle\\_info.h](#).

#### 10.344.3.2 get\_tuner\_type()

uint32\_t Rtl\_Tcp\_Dongle\_Info::get\_tuner\_type () const [inline]

Definition at line 59 of file [rtl\\_tcp\\_dongle\\_info.h](#).

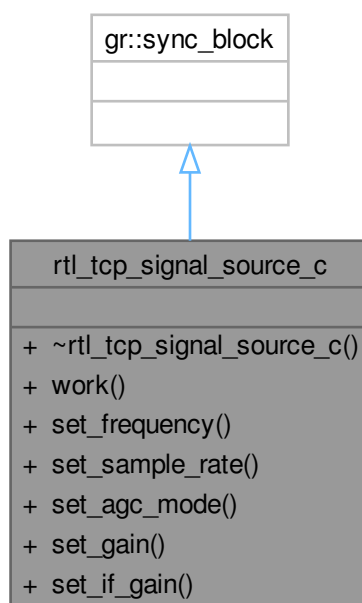
The documentation for this class was generated from the following file:

- [rtl\\_tcp\\_dongle\\_info.h](#)

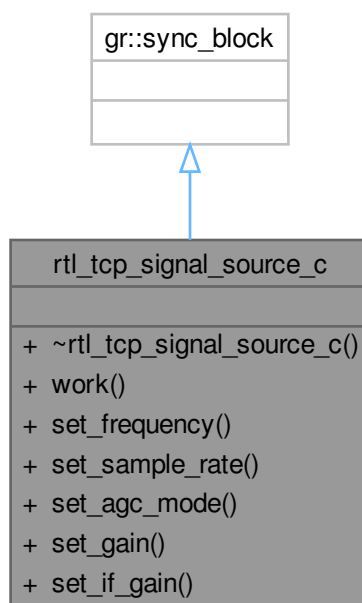
## 10.345 rtl\_tcp\_signal\_source\_c Class Reference

This class reads interleaved I/Q samples from an rtl\_tcp server and outputs complex types.  
`#include <rtl_tcp_signal_source_c.h>`

Inheritance diagram for rtl\_tcp\_signal\_source\_c:



Collaboration diagram for rtl\_tcp\_signal\_source\_c:



### Public Member Functions

- int **work** (int noutput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)
- void **set\_frequency** (int frequency)
- void **set\_sample\_rate** (int sample\_rate)
- void **set\_agc\_mode** (bool agc)
- void **set\_gain** (int gain)
- void **set\_if\_gain** (int gain)

### Friends

- rtl\_tcp\_signal\_source\_c\_sptr **rtl\_tcp\_make\_signal\_source\_c** (const std::string &address, int16\_t port, bool flip\_iq=false)

### 10.345.1 Detailed Description

This class reads interleaved I/Q samples from an rtl\_tcp server and outputs complex types.

Definition at line 65 of file [rtl\\_tcp\\_signal\\_source\\_c.h](#).

The documentation for this class was generated from the following file:

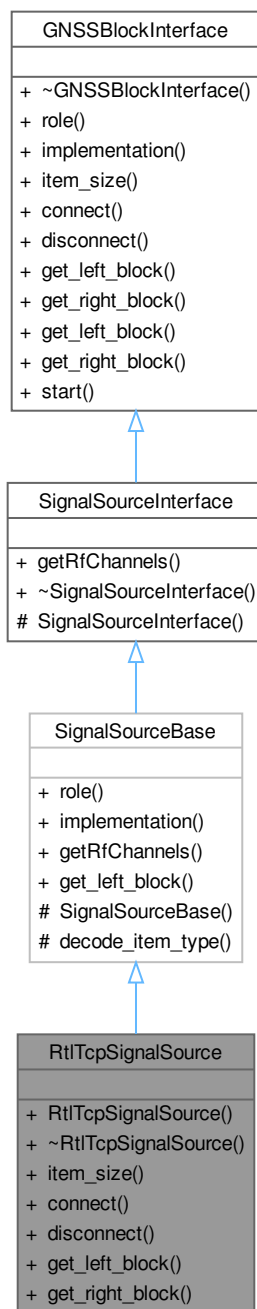
- [rtl\\_tcp\\_signal\\_source\\_c.h](#)

## 10.346 RtlTcpSignalSource Class Reference

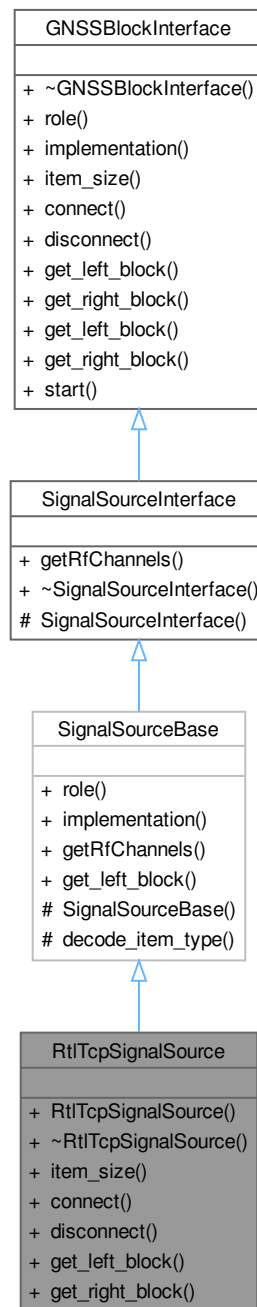
This class reads from rtl\_tcp, which streams interleaved I/Q samples over TCP. (see <https://osmocom.org/projects/rtl-sdr/wiki>)

```
#include <rtl_tcp_signal_source.h>
```

Inheritance diagram for RtlTcpSignalSource:



Collaboration diagram for RtlTcpSignalSource:



## Public Member Functions

- **RtlTcpSignalSource** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_↔ stream, unsigned int out\_stream, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override

## Public Member Functions inherited from [SignalSourceBase](#)

- `std::string` [role](#) () final
- `std::string` [implementation](#) () final
- `size_t` [getRfChannels](#) () const override
- `gr::basic_block_sptr` [get\\_left\\_block](#) () override

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual `gr::basic_block_sptr` [get\\_left\\_block](#) (int RF\_channel)
- virtual `gr::basic_block_sptr` [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

## Additional Inherited Members

## Protected Member Functions inherited from [SignalSourceBase](#)

- [SignalSourceBase](#) ([ConfigurationInterface](#) const \*configuration, `std::string` role, `std::string` impl)  
*Constructor.*
- `size_t` [decode\\_item\\_type](#) (`std::string` const &item\_type, `bool` \*is\_interleaved=nullptr, `bool` throw\_on\_↔ error=false)  
*utility for decoding passed ".item\_type" values*

### 10.346.1 Detailed Description

This class reads from `rtl_tcp`, which streams interleaved I/Q samples over TCP. (see <https://osmocom.org/projects/rtl-sdr/wiki>)

Definition at line 45 of file `rtl_tcp_signal_source.h`.

### 10.346.2 Member Function Documentation

#### 10.346.2.1 `connect()`

```
void RtlTcpSignalSource::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.346.2.2 `disconnect()`

```
void RtlTcpSignalSource::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.346.2.3 `get_left_block()`

```
gr::basic_block_sptr RtlTcpSignalSource::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.346.2.4 `get_right_block()`

```
gr::basic_block_sptr RtlTcpSignalSource::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.346.2.5 item\_size()

size\_t RtlTcpSignalSource::item\_size () [inline], [override], [virtual]

Implements [GNSSBlockInterface](#).

Definition at line 56 of file [rtl\\_tcp\\_signal\\_source.h](#).

The documentation for this class was generated from the following file:

- [rtl\\_tcp\\_signal\\_source.h](#)

## 10.347 Sbas\_Ephemeris Class Reference

This class stores SBAS SV ephemeris data.

#include <sbas\_ephemeris.h>

Collaboration diagram for Sbas\_Ephemeris:

Sbas_Ephemeris
+ i_prn
+ i_t0
+ d_tof
+ i_sv_ura
+ b_sv_do_not_use
+ d_pos
+ d_vel
+ d_acc
+ d_af0
+ d_af1
+ Sbas_Ephemeris()
+ print()

### Public Member Functions

- void **print** (std::ostream &out)

### Public Attributes

- int [i\\_prn](#) {}  
*PRN number.*
- int [i\\_t0](#) {}  
*Reference epoch time (GPST)*
- double [d\\_tof](#) {}  
*Time of message frame (GPST)*
- int [i\\_sv\\_ura](#) {}  
*SV accuracy (URA index), not standardized.*
- bool [b\\_sv\\_do\\_not\\_use](#) {}  
*Health status (false:do not use / true:usable)*
- double [d\\_pos](#) [3] {}



- Satellite position (m) (ECEF)*
  - double `d_vel` [3] {}
- Satellite velocity (m/s) (ECEF)*
  - double `d_acc` [3] {}
- Satellite acceleration (m/s<sup>2</sup>) (ECEF)*
  - double `d_af0` {}
- Satellite clock-offset (s)*
  - double `d_af1` {}
- Satellite drift (s/s)*

### 10.347.1 Detailed Description

This class stores SBAS SV ephemeris data.  
Definition at line 33 of file [sbas\\_ephemeris.h](#).

### 10.347.2 Member Data Documentation

#### 10.347.2.1 `b_sv_do_not_use`

```
bool Sbas_Ephemeris::b_sv_do_not_use {}
```

Health status (false:do not use / true:usable)  
Definition at line 42 of file [sbas\\_ephemeris.h](#).

#### 10.347.2.2 `d_acc`

```
double Sbas_Ephemeris::d_acc[3] {}
```

Satellite acceleration (m/s<sup>2</sup>) (ECEF)  
Definition at line 45 of file [sbas\\_ephemeris.h](#).

#### 10.347.2.3 `d_af0`

```
double Sbas_Ephemeris::d_af0 {}
```

Satellite clock-offset (s)  
Definition at line 46 of file [sbas\\_ephemeris.h](#).

#### 10.347.2.4 `d_af1`

```
double Sbas_Ephemeris::d_af1 {}
```

Satellite drift (s/s)  
Definition at line 47 of file [sbas\\_ephemeris.h](#).

#### 10.347.2.5 `d_pos`

```
double Sbas_Ephemeris::d_pos[3] {}
```

Satellite position (m) (ECEF)  
Definition at line 43 of file [sbas\\_ephemeris.h](#).

#### 10.347.2.6 `d_tof`

```
double Sbas_Ephemeris::d_tof {}
```

Time of message frame (GPST)  
Definition at line 40 of file [sbas\\_ephemeris.h](#).

#### 10.347.2.7 `d_vel`

```
double Sbas_Ephemeris::d_vel[3] {}
```

Satellite velocity (m/s) (ECEF)  
Definition at line 44 of file [sbas\\_ephemeris.h](#).

**10.347.2.8 i\_prn**

```
int Sbas_Ephemeris::i_prn {}
```

PRN number.

Definition at line 38 of file [sbas\\_ephemeris.h](#).

**10.347.2.9 i\_sv\_ura**

```
int Sbas_Ephemeris::i_sv_ura {}
```

SV accuracy (URA index), not standardized.

Definition at line 41 of file [sbas\\_ephemeris.h](#).

**10.347.2.10 i\_t0**

```
int Sbas_Ephemeris::i_t0 {}
```

Reference epoch time (GPST)

Definition at line 39 of file [sbas\\_ephemeris.h](#).

The documentation for this class was generated from the following file:

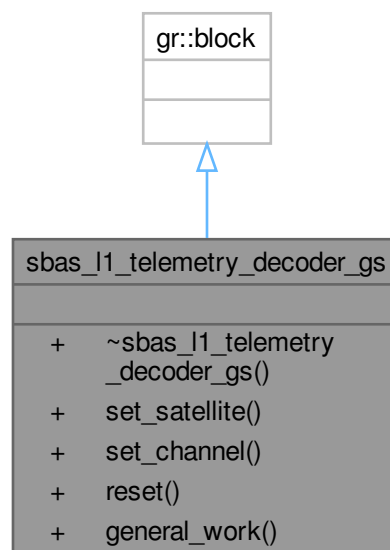
- [sbas\\_ephemeris.h](#)

**10.348 sbas\_l1\_telemetry\_decoder\_gs Class Reference**

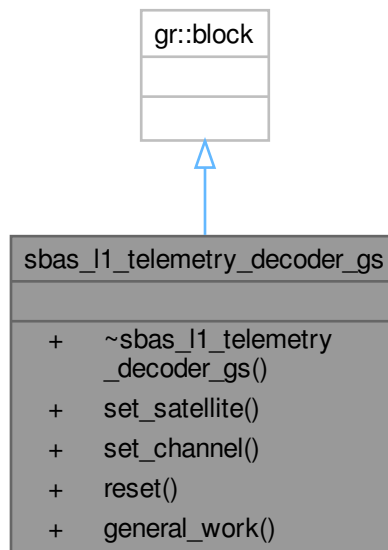
This class implements a block that decodes the SBAS integrity and corrections data defined in RTCA MOPS DO-229.

```
#include <sbas_l1_telemetry_decoder_gs.h>
```

Inheritance diagram for `sbas_l1_telemetry_decoder_gs`:



Collaboration diagram for sbas\_l1\_telemetry\_decoder\_gs:



### Public Member Functions

- void `set_satellite` (const [Gnss\\_Satellite](#) &satellite)  
*Set satellite PRN.*
- void `set_channel` (int32\_t channel)  
*Set receiver's channel.*
- void `reset` ()
- int `general_work` (int noutput\_items, gr\_vector\_int &ninput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items) override  
*This is where all signal processing takes place.*

### Friends

- sbas\_l1\_telemetry\_decoder\_gs\_sptr `sbas_l1_make_telemetry_decoder_gs` (const [Gnss\\_Satellite](#) &satellite, bool dump)

## 10.348.1 Detailed Description

This class implements a block that decodes the SBAS integrity and corrections data defined in RTCA MOPS DO-229.

Definition at line 54 of file [sbas\\_l1\\_telemetry\\_decoder\\_gs.h](#).

## 10.348.2 Member Function Documentation

### 10.348.2.1 `general_work()`

```

int sbas_l1_telemetry_decoder_gs::general_work (
    int noutput_items,
    gr_vector_int & ninput_items,
    gr_vector_const_void_star & input_items,
    gr_vector_void_star & output_items) [override]
  
```

This is where all signal processing takes place.

#### 10.348.2.2 `reset()`

```
void sbas_l1_telemetry_decoder_gs::reset () [inline]
```

Definition at line 60 of file [sbas\\_l1\\_telemetry\\_decoder\\_gs.h](#).

#### 10.348.2.3 `set_channel()`

```
void sbas_l1_telemetry_decoder_gs::set_channel (  
    int32_t channel)
```

Set receiver's channel.

#### 10.348.2.4 `set_satellite()`

```
void sbas_l1_telemetry_decoder_gs::set_satellite (  
    const Gnss_Satellite & satellite)
```

Set satellite PRN.

The documentation for this class was generated from the following file:

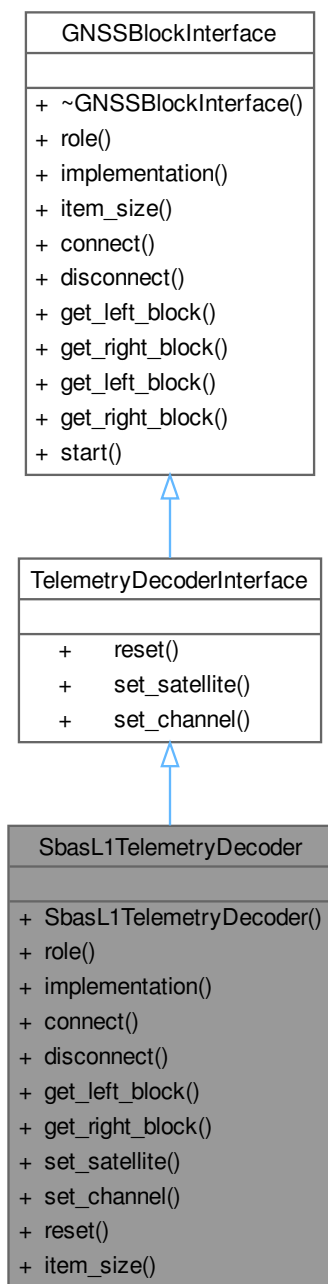
- [sbas\\_l1\\_telemetry\\_decoder\\_gs.h](#)

## 10.349 SbasL1TelemetryDecoder Class Reference

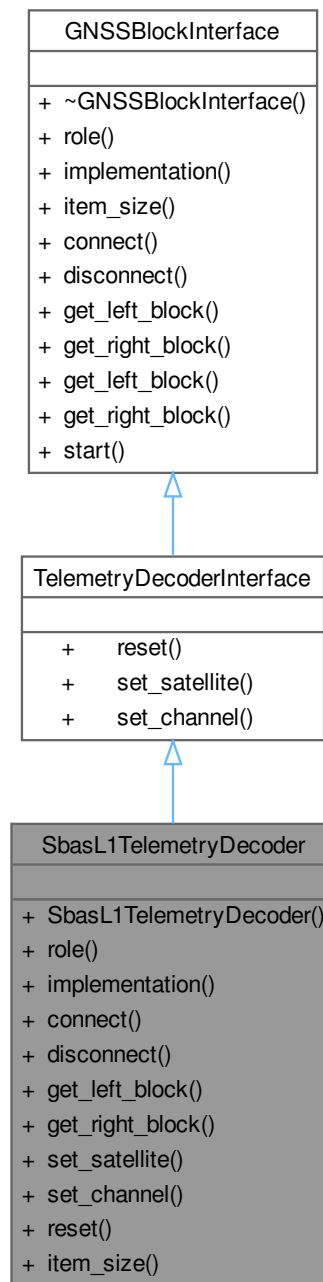
This class implements a NAV data decoder for SBAS frames in L1 radio link.

```
#include <sbas_l1_telemetry_decoder.h>
```

Inheritance diagram for SbasL1TelemetryDecoder:



Collaboration diagram for SbasL1TelemetryDecoder:



### Public Member Functions

- **SbasL1TelemetryDecoder** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams)
- std::string [role](#) () override
- std::string [implementation](#) () override  
*Returns "SBAS\_L1\_Telemetry\_Decoder".*
- void [connect](#) (gr::top\_block\_sptr top\_block) override

- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- void [set\\_satellite](#) (const [Gnss\\_Satellite](#) &satellite) override
- void [set\\_channel](#) (int channel) override
- void [reset](#) () override
- size\_t [item\\_size](#) () override

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

## 10.349.1 Detailed Description

This class implements a NAV data decoder for SBAS frames in L1 radio link.  
Definition at line 41 of file [sbas\\_l1\\_telemetry\\_decoder.h](#).

## 10.349.2 Member Function Documentation

### 10.349.2.1 connect()

```
void SbasL1TelemetryDecoder::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.349.2.2 disconnect()

```
void SbasL1TelemetryDecoder::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.349.2.3 get\_left\_block()

```
gr::basic_block_sptr SbasL1TelemetryDecoder::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.349.2.4 get\_right\_block()

```
gr::basic_block_sptr SbasL1TelemetryDecoder::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.349.2.5 implementation()

```
std::string SbasL1TelemetryDecoder::implementation () [inline], [override], [virtual]
```

Returns "SBAS\_L1\_Telemetry\_Decoder".

Implements [GNSSBlockInterface](#).

Definition at line 58 of file [sbas\\_l1\\_telemetry\\_decoder.h](#).

### 10.349.2.6 item\_size()

```
size_t SbasL1TelemetryDecoder::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 77 of file [sbas\\_l1\\_telemetry\\_decoder.h](#).

**10.349.2.7 reset()**

```
void SbasL1TelemetryDecoder::reset () [inline], [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

Definition at line 72 of file [sbas\\_l1\\_telemetry\\_decoder.h](#).

**10.349.2.8 role()**

```
std::string SbasL1TelemetryDecoder::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 50 of file [sbas\\_l1\\_telemetry\\_decoder.h](#).

**10.349.2.9 set\_channel()**

```
void SbasL1TelemetryDecoder::set_channel (
    int channel) [inline], [override], [virtual]
```

Implements [TelemetryDecoderInterface](#).

Definition at line 70 of file [sbas\\_l1\\_telemetry\\_decoder.h](#).

**10.349.2.10 set\_satellite()**

```
void SbasL1TelemetryDecoder::set_satellite (
    const Gnss_Satellite & satellite) [override], [virtual]
```

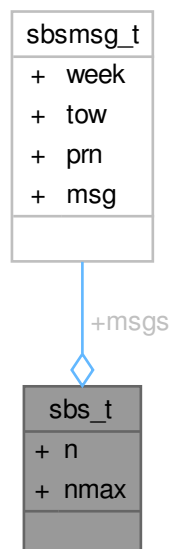
Implements [TelemetryDecoderInterface](#).

The documentation for this class was generated from the following file:

- [sbas\\_l1\\_telemetry\\_decoder.h](#)

**10.350 sbs\_t Struct Reference**

Collaboration diagram for sbs\_t:





## Public Attributes

- `int n`
- `int nmax`
- `sbsmsg_t * msgs`

## 10.350.1 Detailed Description

Definition at line 573 of file [rtklib.h](#).

## 10.350.2 Member Data Documentation

### 10.350.2.1 msgs

`sbsmsg_t* sbs_t::msgs`

Definition at line 576 of file [rtklib.h](#).

### 10.350.2.2 n

`int sbs_t::n`

Definition at line 575 of file [rtklib.h](#).

### 10.350.2.3 nmax

`int sbs_t::nmax`

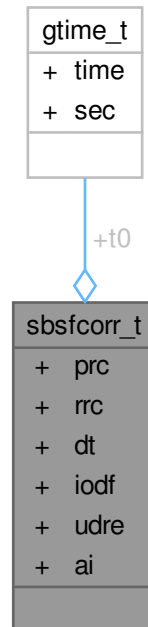
Definition at line 575 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.351 sbsfcorr\_t Struct Reference

Collaboration diagram for sbsfcorr\_t:



### Public Attributes

- [gtime\\_t t0](#)
- double [prc](#)
- double [rrc](#)
- double [dt](#)
- int [iodf](#)
- short [udre](#)
- short [ai](#)

### 10.351.1 Detailed Description

Definition at line 580 of file [rtklib.h](#).

### 10.351.2 Member Data Documentation

#### 10.351.2.1 ai

```
short sbsfcorr_t::ai
```

Definition at line 588 of file [rtklib.h](#).

#### 10.351.2.2 dt

```
double sbsfcorr_t::dt
```

Definition at line 585 of file [rtklib.h](#).

### 10.351.2.3 iodef

```
int sbsfcorr_t::iodf
```

Definition at line 586 of file [rtklib.h](#).

### 10.351.2.4 prc

```
double sbsfcorr_t::prc
```

Definition at line 583 of file [rtklib.h](#).

### 10.351.2.5 rrc

```
double sbsfcorr_t::rrc
```

Definition at line 584 of file [rtklib.h](#).

### 10.351.2.6 t0

```
gtime_t sbsfcorr_t::t0
```

Definition at line 582 of file [rtklib.h](#).

### 10.351.2.7 udre

```
short sbsfcorr_t::udre
```

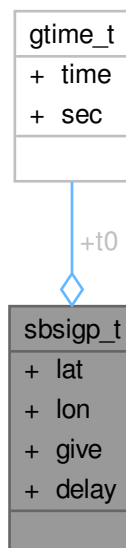
Definition at line 587 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.352 sbsigp\_t Struct Reference

Collaboration diagram for sbsigp\_t:



## Public Attributes

- [gtime\\_t t0](#)
- short [lat](#)
- short [lon](#)
- short [give](#)
- float [delay](#)

### 10.352.1 Detailed Description

Definition at line [619](#) of file [rtklib.h](#).

### 10.352.2 Member Data Documentation

#### 10.352.2.1 delay

```
float sbsigp_t::delay
```

Definition at line [624](#) of file [rtklib.h](#).

#### 10.352.2.2 give

```
short sbsigp_t::give
```

Definition at line [623](#) of file [rtklib.h](#).

#### 10.352.2.3 lat

```
short sbsigp_t::lat
```

Definition at line [622](#) of file [rtklib.h](#).

#### 10.352.2.4 lon

```
short sbsigp_t::lon
```

Definition at line [622](#) of file [rtklib.h](#).

#### 10.352.2.5 t0

```
gtime\_t sbsigp_t::t0
```

Definition at line [621](#) of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.353 sbsigpband\_t Struct Reference

Collaboration diagram for sbsigpband\_t:

sbsigpband_t	
+	x
+	y
+	bits
+	bite

### Public Attributes

- short [x](#)
- const short \* [y](#)
- unsigned char [bits](#)
- unsigned char [bite](#)

### 10.353.1 Detailed Description

Definition at line [628](#) of file [rtklib.h](#).

### 10.353.2 Member Data Documentation

#### 10.353.2.1 bite

```
unsigned char sbsigpband_t::bite
```

Definition at line [633](#) of file [rtklib.h](#).

#### 10.353.2.2 bits

```
unsigned char sbsigpband_t::bits
```

Definition at line [632](#) of file [rtklib.h](#).

#### 10.353.2.3 x

```
short sbsigpband_t::x
```

Definition at line [630](#) of file [rtklib.h](#).

#### 10.353.2.4 y

```
const short* sbsigpband_t::y
```

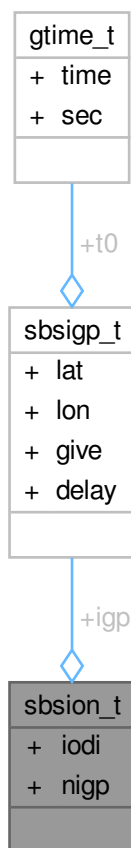
Definition at line [631](#) of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.354 sbsion\_t Struct Reference

Collaboration diagram for sbsion\_t:



### Public Attributes

- int [iodi](#)
- int [nigp](#)
- [sbsigp\\_t igp](#) [[MAXNIGP](#)]

### 10.354.1 Detailed Description

Definition at line [637](#) of file [rtklib.h](#).

### 10.354.2 Member Data Documentation

#### 10.354.2.1 igp

[sbsigp\\_t](#) [sbsion\\_t::igp](#) [[MAXNIGP](#)]

Definition at line [641](#) of file [rtklib.h](#).

#### 10.354.2.2 iodi

`int sbsion_t::iodi`

Definition at line 639 of file [rtklib.h](#).

### 10.354.2.3 nignp

```
int sbsion_t::nignp
```

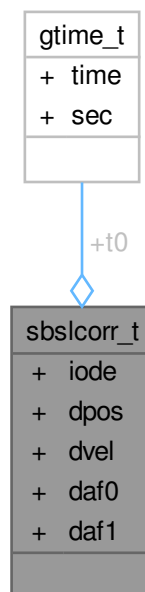
Definition at line 640 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.355 sbslcorr\_t Struct Reference

Collaboration diagram for sbslcorr\_t:



### Public Attributes

- [gtime\\_t t0](#)
- int [iode](#)
- double [dpos](#) [3]
- double [dvel](#) [3]
- double [daf0](#)
- double [daf1](#)

### 10.355.1 Detailed Description

Definition at line 592 of file [rtklib.h](#).

### 10.355.2 Member Data Documentation

#### 10.355.2.1 daf0

```
double sbslcorr_t::daf0
```

Definition at line 598 of file [rtklib.h](#).

#### 10.355.2.2 daf1

```
double sbslcorr_t::daf1
```

Definition at line 598 of file [rtklib.h](#).

#### 10.355.2.3 dpos

```
double sbslcorr_t::dpos[3]
```

Definition at line 596 of file [rtklib.h](#).

#### 10.355.2.4 dvel

```
double sbslcorr_t::dvel[3]
```

Definition at line 597 of file [rtklib.h](#).

#### 10.355.2.5 iode

```
int sbslcorr_t::iode
```

Definition at line 595 of file [rtklib.h](#).

#### 10.355.2.6 t0

```
ptime_t sbslcorr_t::t0
```

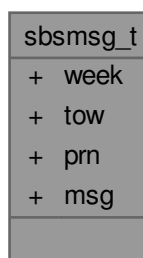
Definition at line 594 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.356 sbsmsg\_t Struct Reference

Collaboration diagram for sbsmsg\_t:



### Public Attributes

- int [week](#)
- int [tow](#)
- int [prn](#)
- unsigned char [msg](#) [29]



### 10.356.1 Detailed Description

Definition at line 565 of file [rtklib.h](#).

### 10.356.2 Member Data Documentation

#### 10.356.2.1 msg

```
unsigned char sbsmsg_t::msg[29]
```

Definition at line 569 of file [rtklib.h](#).

#### 10.356.2.2 prn

```
int sbsmsg_t::prn
```

Definition at line 568 of file [rtklib.h](#).

#### 10.356.2.3 tow

```
int sbsmsg_t::tow
```

Definition at line 567 of file [rtklib.h](#).

#### 10.356.2.4 week

```
int sbsmsg_t::week
```

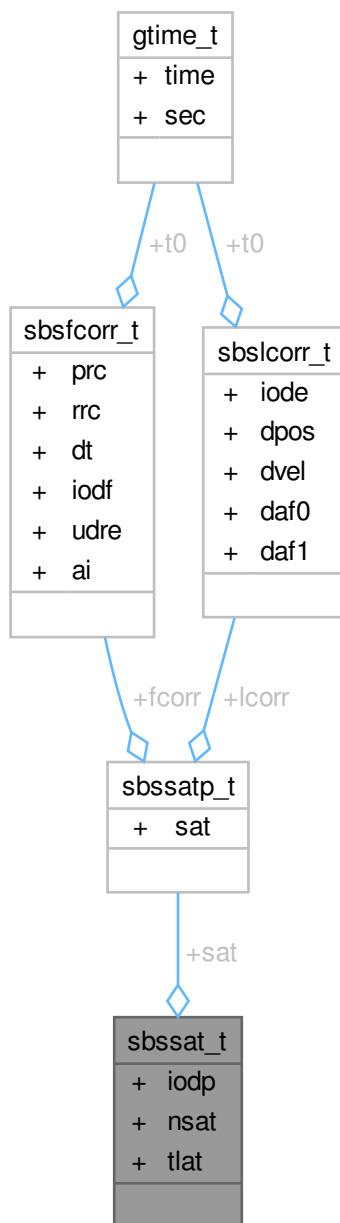
Definition at line 567 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.357 sbssat\_t Struct Reference

Collaboration diagram for sbssat\_t:



### Public Attributes

- int `iodp`
- int `nsat`
- int `tlat`
- `sbssatp_t sat` [MAXSAT]

### 10.357.1 Detailed Description

Definition at line 610 of file [rtklib.h](#).

### 10.357.2 Member Data Documentation

#### 10.357.2.1 iodp

```
int sbssat_t::iodp
```

Definition at line 612 of file [rtklib.h](#).

#### 10.357.2.2 nsat

```
int sbssat_t::nsat
```

Definition at line 613 of file [rtklib.h](#).

#### 10.357.2.3 sat

```
sbssatp_t sbssat_t::sat [MAXSAT]
```

Definition at line 615 of file [rtklib.h](#).

#### 10.357.2.4 tlat

```
int sbssat_t::tlat
```

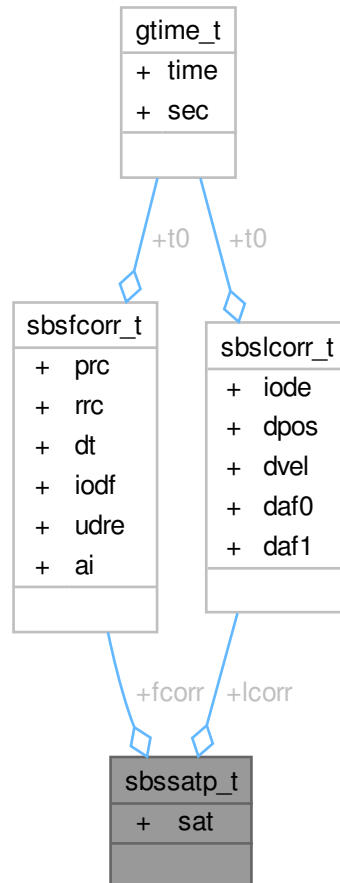
Definition at line 614 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.358 sbssatp\_t Struct Reference

Collaboration diagram for sbssatp\_t:



### Public Attributes

- int `sat`
- `sbsfcorr_t` `fcorr`
- `sbslcorr_t` `lcorr`

### 10.358.1 Detailed Description

Definition at line 602 of file `rtklib.h`.

### 10.358.2 Member Data Documentation

#### 10.358.2.1 fcorr

`sbsfcorr_t` `sbssatp_t::fcorr`

Definition at line 605 of file `rtklib.h`.

### 10.358.2.2 lcorr

`sbslcorr_t` `sbssatp_t::lcorr`

Definition at line 606 of file [rtklib.h](#).

### 10.358.2.3 sat

`int` `sbssatp_t::sat`

Definition at line 604 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.359 sdr\_gnss\_packet\_t Struct Reference

Collaboration diagram for `sdr_gnss_packet_t`:

sdr_gnss_packet_t
+ nsuffix
+ gps_satellites
+ galileo_satellites
+ microseconds
+ latitude
+ longitude
+ height
+ velocity
+ prn
+ snr
+ doppler
+ sats
+ reserved
+ status

### Public Attributes

- `uint8_t` [nsuffix](#)
- `uint8_t` [gps\\_satellites](#)
- `uint8_t` [galileo\\_satellites](#)
- `uint32_t` [microseconds](#)
- `double` [latitude](#)
- `double` [longitude](#)
- `double` [height](#)
- `float` [velocity](#) [3]

- struct {
  - uint8\_t [prn](#)
  - uint8\_t [snr](#)
  - int16\_t [doppler](#)
- } [sats](#) [6]
- uint32\_t [reserved](#)
- uint16\_t [status](#)

### 10.359.1 Detailed Description

Definition at line 39 of file [an\\_packet\\_printer.h](#).

### 10.359.2 Member Data Documentation

#### 10.359.2.1 doppler

int16\_t sdr\_gnss\_packet\_t::doppler

Definition at line 54 of file [an\\_packet\\_printer.h](#).

#### 10.359.2.2 galileo\_satellites

uint8\_t sdr\_gnss\_packet\_t::galileo\_satellites

Definition at line 43 of file [an\\_packet\\_printer.h](#).

#### 10.359.2.3 gps\_satellites

uint8\_t sdr\_gnss\_packet\_t::gps\_satellites

Definition at line 42 of file [an\\_packet\\_printer.h](#).

#### 10.359.2.4 height

double sdr\_gnss\_packet\_t::height

Definition at line 47 of file [an\\_packet\\_printer.h](#).

#### 10.359.2.5 latitude

double sdr\_gnss\_packet\_t::latitude

Definition at line 45 of file [an\\_packet\\_printer.h](#).

#### 10.359.2.6 longitude

double sdr\_gnss\_packet\_t::longitude

Definition at line 46 of file [an\\_packet\\_printer.h](#).

#### 10.359.2.7 microseconds

uint32\_t sdr\_gnss\_packet\_t::microseconds

Definition at line 44 of file [an\\_packet\\_printer.h](#).

#### 10.359.2.8 nsfix

uint8\_t sdr\_gnss\_packet\_t::nsfix

Definition at line 41 of file [an\\_packet\\_printer.h](#).

#### 10.359.2.9 prn

uint8\_t sdr\_gnss\_packet\_t::prn

Definition at line 52 of file [an\\_packet\\_printer.h](#).

**10.359.2.10 reserved**

`uint32_t sdr_gnss_packet_t::reserved`  
Definition at line 57 of file [an\\_packet\\_printer.h](#).

**10.359.2.11 snr**

`uint8_t sdr_gnss_packet_t::snr`  
Definition at line 53 of file [an\\_packet\\_printer.h](#).

**10.359.2.12 status**

`uint16_t sdr_gnss_packet_t::status`  
Definition at line 58 of file [an\\_packet\\_printer.h](#).

**10.359.2.13 velocity**

`float sdr_gnss_packet_t::velocity[3]`  
Definition at line 48 of file [an\\_packet\\_printer.h](#).  
The documentation for this struct was generated from the following file:

- [an\\_packet\\_printer.h](#)

**10.360 seph\_t Struct Reference**

Collaboration diagram for `seph_t`:



## Public Attributes

- int [sat](#)
- [gtime\\_t](#) [t0](#)
- [gtime\\_t](#) [tof](#)
- int [sva](#)
- int [svh](#)
- double [pos](#) [3]
- double [vel](#) [3]
- double [acc](#) [3]
- double [af0](#)
- double [af1](#)

### 10.360.1 Detailed Description

Definition at line 500 of file [rtklib.h](#).

### 10.360.2 Member Data Documentation

#### 10.360.2.1 [acc](#)

```
double seph_t::acc[3]
```

Definition at line 509 of file [rtklib.h](#).

#### 10.360.2.2 [af0](#)

```
double seph_t::af0
```

Definition at line 510 of file [rtklib.h](#).

#### 10.360.2.3 [af1](#)

```
double seph_t::af1
```

Definition at line 510 of file [rtklib.h](#).

#### 10.360.2.4 [pos](#)

```
double seph_t::pos[3]
```

Definition at line 507 of file [rtklib.h](#).

#### 10.360.2.5 [sat](#)

```
int seph_t::sat
```

Definition at line 502 of file [rtklib.h](#).

#### 10.360.2.6 [sva](#)

```
int seph_t::sva
```

Definition at line 505 of file [rtklib.h](#).

#### 10.360.2.7 [svh](#)

```
int seph_t::svh
```

Definition at line 506 of file [rtklib.h](#).

#### 10.360.2.8 [t0](#)

```
gtime\_t seph_t::t0
```

Definition at line 503 of file [rtklib.h](#).



### 10.360.2.9 tof

`gtime_t seph_t::tof`

Definition at line 504 of file [rtklib.h](#).

### 10.360.2.10 vel

`double seph_t::vel[3]`

Definition at line 508 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

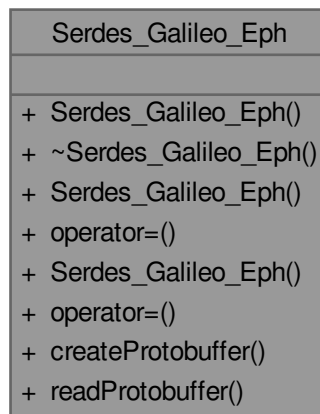
- [rtklib.h](#)

## 10.361 Serdes\_Galileo\_Eph Class Reference

This class implements serialization and deserialization of [Galileo\\_Ephemeris](#) using Protocol Buffers.

`#include <serdes_galileo_eph.h>`

Collaboration diagram for Serdes\_Galileo\_Eph:



### Public Member Functions

- [Serdes\\_Galileo\\_Eph](#) (const [Serdes\\_Galileo\\_Eph](#) &other) noexcept
- [Serdes\\_Galileo\\_Eph](#) & operator= (const [Serdes\\_Galileo\\_Eph](#) &rhs) noexcept  
*< Copy assignment operator*
- [Serdes\\_Galileo\\_Eph](#) ([Serdes\\_Galileo\\_Eph](#) &&other) noexcept
- [Serdes\\_Galileo\\_Eph](#) & operator= ([Serdes\\_Galileo\\_Eph](#) &&other) noexcept  
*< Move assignment operator*
- `std::string` [createProtobuffer](#) (const `std::shared_ptr< Galileo\_Ephemeris >` monitor)
- [Galileo\\_Ephemeris](#) [readProtobuffer](#) (const `gnss_sdr::GalileoEphemeris` &mon) const  
*< Deserialization*

### 10.361.1 Detailed Description

This class implements serialization and deserialization of [Galileo\\_Ephemeris](#) using Protocol Buffers.

Definition at line 37 of file [serdes\\_galileo\\_eph.h](#).

## 10.361.2 Constructor & Destructor Documentation

### 10.361.2.1 Serdes\_Galileo\_Eph() [1/3]

`Serdes_Galileo_Eph::Serdes_Galileo_Eph () [inline]`

Definition at line 40 of file [serdes\\_galileo\\_eph.h](#).

### 10.361.2.2 ~Serdes\_Galileo\_Eph()

`Serdes_Galileo_Eph::~~Serdes_Galileo_Eph () [inline]`

Definition at line 47 of file [serdes\\_galileo\\_eph.h](#).

### 10.361.2.3 Serdes\_Galileo\_Eph() [2/3]

`Serdes_Galileo_Eph::Serdes_Galileo_Eph (const Serdes_Galileo_Eph & other) [inline], [noexcept]`

Definition at line 52 of file [serdes\\_galileo\\_eph.h](#).

### 10.361.2.4 Serdes\_Galileo\_Eph() [3/3]

`Serdes_Galileo_Eph::Serdes_Galileo_Eph (Serdes_Galileo_Eph && other) [inline], [noexcept]`

Definition at line 63 of file [serdes\\_galileo\\_eph.h](#).

## 10.361.3 Member Function Documentation

### 10.361.3.1 createProtobuffer()

`std::string Serdes_Galileo_Eph::createProtobuffer (const std::shared_ptr< Galileo_Ephemeris > monitor) [inline]`

#### Parameters

<i>monitor</i>	Serialization into a string
----------------	-----------------------------

Definition at line 73 of file [serdes\\_galileo\\_eph.h](#).

### 10.361.3.2 operator=() [1/2]

`Serdes_Galileo_Eph & Serdes_Galileo_Eph::operator= (const Serdes_Galileo_Eph & rhs) [inline], [noexcept]`

< Copy assignment operator

Definition at line 56 of file [serdes\\_galileo\\_eph.h](#).

### 10.361.3.3 operator=() [2/2]

`Serdes_Galileo_Eph & Serdes_Galileo_Eph::operator= (Serdes_Galileo_Eph && other) [inline], [noexcept]`

< Move assignment operator

Definition at line 67 of file [serdes\\_galileo\\_eph.h](#).

### 10.361.3.4 readProtobuffer()

`Galileo_Ephemeris Serdes_Galileo_Eph::readProtobuffer (const gnss_sdr::GalileoEphemeris & mon) const [inline]`

< Deserialization

Definition at line 122 of file [serdes\\_galileo\\_eph.h](#).

References [Gnss\\_Ephemeris::af0](#), [Gnss\\_Ephemeris::af1](#), [Gnss\\_Ephemeris::af2](#), [Galileo\\_Ephemeris::BGD\\_E1E5a](#), [Galileo\\_Ephemeris::BGD\\_E1E5b](#), [Gnss\\_Ephemeris::Cic](#), [Gnss\\_Ephemeris::Cis](#), [Gnss\\_Ephemeris::Crc](#), [Gnss\\_Ephemeris::Crs](#), [Gnss\\_Ephemeris::Cuc](#), [Gnss\\_Ephemeris::Cus](#), [Gnss\\_Ephemeris::delta\\_n](#), [Gnss\\_Ephemeris::dtr](#), [Galileo\\_Ephemeris::E1B\\_DVS](#),

Galileo\_Ephemeris::E1B\_HS, Galileo\_Ephemeris::E5a\_DVS, Galileo\_Ephemeris::E5a\_HS, Galileo\_Ephemeris::E5b\_DVS, Galileo\_Ephemeris::E5b\_HS, Gnss\_Ephemeris::ecc, Gnss\_Ephemeris::i\_0, Gnss\_Ephemeris::idot, Gnss\_Ephemeris::M\_0, Gnss\_Ephemeris::omega, Gnss\_Ephemeris::OMEGA\_0, Gnss\_Ephemeris::OMEGAdot, Gnss\_Ephemeris::PRN, Gnss\_Ephemeris::satClkDrift, Galileo\_Ephemeris::SISA, Gnss\_Ephemeris::sqrtA, Gnss\_Ephemeris::toc, Gnss\_Ephemeris::toe, Gnss\_Ephemeris::tow, and Gnss\_Ephemeris::WN.

The documentation for this class was generated from the following file:

- [serdes\\_galileo\\_eph.h](#)

## 10.362 Serdes\_Gnss\_Synchro Class Reference

This class implements serialization and deserialization of [Gnss\\_Synchro](#) objects using Protocol Buffers.

```
#include <serdes_gnss_synchro.h>
```

Collaboration diagram for Serdes\_Gnss\_Synchro:

Serdes_Gnss_Synchro
<ul style="list-style-type: none"> <li>+ Serdes_Gnss_Synchro()</li> <li>+ ~Serdes_Gnss_Synchro()</li> <li>+ Serdes_Gnss_Synchro()</li> <li>+ operator=()</li> <li>+ Serdes_Gnss_Synchro()</li> <li>+ operator=()</li> <li>+ createProtobuffer()</li> <li>+ readProtobuffer()</li> </ul>

### Public Member Functions

- [Serdes\\_Gnss\\_Synchro](#) (const [Serdes\\_Gnss\\_Synchro](#) &other) noexcept  
*< Copy constructor*
- [Serdes\\_Gnss\\_Synchro](#) & operator= (const [Serdes\\_Gnss\\_Synchro](#) &rhs) noexcept  
*< Copy assignment operator*
- [Serdes\\_Gnss\\_Synchro](#) ([Serdes\\_Gnss\\_Synchro](#) &&other) noexcept  
*< Move constructor*
- [Serdes\\_Gnss\\_Synchro](#) & operator= ([Serdes\\_Gnss\\_Synchro](#) &&other) noexcept  
*< Move assignment operator*
- std::string [createProtobuffer](#) (const std::vector< [Gnss\\_Synchro](#) > &vgs)
- std::vector< [Gnss\\_Synchro](#) > [readProtobuffer](#) (const gnss\_sdr::Observables &obs) const  
*< Deserialization*

### 10.362.1 Detailed Description

This class implements serialization and deserialization of [Gnss\\_Synchro](#) objects using Protocol Buffers.

Definition at line 32 of file [serdes\\_gnss\\_synchro.h](#).

## 10.362.2 Constructor & Destructor Documentation

### 10.362.2.1 Serdes\_Gnss\_Synchro() [1/3]

`Serdes_Gnss_Synchro::Serdes_Gnss_Synchro () [inline]`

Definition at line 35 of file [serdes\\_gnss\\_synchro.h](#).

### 10.362.2.2 ~Serdes\_Gnss\_Synchro()

`Serdes_Gnss_Synchro::~~Serdes_Gnss_Synchro () [inline]`

Definition at line 42 of file [serdes\\_gnss\\_synchro.h](#).

### 10.362.2.3 Serdes\_Gnss\_Synchro() [2/3]

`Serdes_Gnss_Synchro::Serdes_Gnss_Synchro (const Serdes_Gnss_Synchro & other) [inline], [noexcept]`

< Copy constructor

Definition at line 47 of file [serdes\\_gnss\\_synchro.h](#).

### 10.362.2.4 Serdes\_Gnss\_Synchro() [3/3]

`Serdes_Gnss_Synchro::Serdes_Gnss_Synchro (Serdes_Gnss_Synchro && other) [inline], [noexcept]`

< Move constructor

Definition at line 58 of file [serdes\\_gnss\\_synchro.h](#).

## 10.362.3 Member Function Documentation

### 10.362.3.1 createProtobuffer()

`std::string Serdes_Gnss_Synchro::createProtobuffer (const std::vector< Gnss_Synchro > & vgs) [inline]`

Parameters

<code>vgs</code>	Serialization into a string
------------------	-----------------------------

Definition at line 72 of file [serdes\\_gnss\\_synchro.h](#).

### 10.362.3.2 operator=() [1/2]

`Serdes_Gnss_Synchro & Serdes_Gnss_Synchro::operator= (const Serdes_Gnss_Synchro & rhs) [inline], [noexcept]`

< Copy assignment operator

Definition at line 52 of file [serdes\\_gnss\\_synchro.h](#).

### 10.362.3.3 operator=() [2/2]

`Serdes_Gnss_Synchro & Serdes_Gnss_Synchro::operator= (Serdes_Gnss_Synchro && other) [inline], [noexcept]`

< Move assignment operator

Definition at line 63 of file [serdes\\_gnss\\_synchro.h](#).

### 10.362.3.4 readProtobuffer()

`std::vector< Gnss_Synchro > Serdes_Gnss_Synchro::readProtobuffer (const gnss_sdr::Observables & obs) const [inline]`

< Deserialization

Definition at line 122 of file [serdes\\_gnss\\_synchro.h](#).

References [Gnss\\_Synchro::Acq\\_delay\\_samples](#), [Gnss\\_Synchro::Acq\\_doppler\\_hz](#), [Gnss\\_Synchro::Acq\\_doppler\\_step](#), [Gnss\\_Synchro::Acq\\_samplestamp\\_samples](#), [Gnss\\_Synchro::Carrier\\_Doppler\\_hz](#), [Gnss\\_Synchro::Carrier\\_phase\\_rads](#), [Gnss\\_Synchro::Channel\\_ID](#), [Gnss\\_Synchro::CN0\\_dB\\_hz](#), [Gnss\\_Synchro::Code\\_phase\\_samples](#), [Gnss\\_Synchro::correlation\\_length](#), [Gnss\\_Synchro::Flag\\_PLL\\_180\\_deg\\_phase\\_locked](#), [Gnss\\_Synchro::Flag\\_valid\\_acquisition](#), [Gnss\\_Synchro::Flag\\_valid\\_pseudorange](#), [Gnss\\_Synchro::Flag\\_valid\\_symbol\\_output](#), [Gnss\\_Synchro::Flag\\_valid\\_word](#), [Gnss\\_Synchro::fs](#), [Gnss\\_Synchro::interp\\_TOW\\_ms](#), [Gnss\\_Synchro::PRN](#), [Gnss\\_Synchro::Prompt\\_I](#), [Gnss\\_Synchro::Prompt\\_Q](#), [Gnss\\_Synchro::Pseudorange\\_m](#), [Gnss\\_Synchro::RX\\_time](#), [Gnss\\_Synchro::Signal](#), [Gnss\\_Synchro::System](#), [Gnss\\_Synchro::TOW\\_at\\_current\\_symbol\\_ms](#), and [Gnss\\_Synchro::Tracking\\_sample\\_counter](#).

The documentation for this class was generated from the following file:

- [serdes\\_gnss\\_synchro.h](#)

## 10.363 Serdes\_Gps\_Eph Class Reference

This class implements serialization and deserialization of [Gps\\_Ephemeris](#) objects using Protocol Buffers.

```
#include <serdes_gps_eph.h>
```

Collaboration diagram for Serdes\_Gps\_Eph:

Serdes_Gps_Eph
<ul style="list-style-type: none"> <li>+ Serdes_Gps_Eph()</li> <li>+ ~Serdes_Gps_Eph()</li> <li>+ Serdes_Gps_Eph()</li> <li>+ operator=()</li> <li>+ Serdes_Gps_Eph()</li> <li>+ operator=()</li> <li>+ createProtobuffer()</li> <li>+ readProtobuffer()</li> </ul>

### Public Member Functions

- [Serdes\\_Gps\\_Eph](#) (const [Serdes\\_Gps\\_Eph](#) &other) noexcept
- [Serdes\\_Gps\\_Eph](#) & operator= (const [Serdes\\_Gps\\_Eph](#) &rhs) noexcept  
*< Copy assignment operator*
- [Serdes\\_Gps\\_Eph](#) ([Serdes\\_Gps\\_Eph](#) &&other) noexcept
- [Serdes\\_Gps\\_Eph](#) & operator= ([Serdes\\_Gps\\_Eph](#) &&other) noexcept  
*< Move assignment operator*
- std::string [createProtobuffer](#) (const std::shared\_ptr< [Gps\\_Ephemeris](#) > monitor)
- [Gps\\_Ephemeris](#) [readProtobuffer](#) (const gnss\_sdr::GpsEphemeris &mon) const  
*< Deserialization*

### 10.363.1 Detailed Description

This class implements serialization and deserialization of [Gps\\_Ephemeris](#) objects using Protocol Buffers.

Definition at line 36 of file [serdes\\_gps\\_eph.h](#).

## 10.363.2 Constructor & Destructor Documentation

### 10.363.2.1 Serdes\_Gps\_Eph() [1/3]

`Serdes_Gps_Eph::Serdes_Gps_Eph () [inline]`

Definition at line 39 of file [serdes\\_gps\\_eph.h](#).

### 10.363.2.2 ~Serdes\_Gps\_Eph()

`Serdes_Gps_Eph::~~Serdes_Gps_Eph () [inline]`

Definition at line 46 of file [serdes\\_gps\\_eph.h](#).

### 10.363.2.3 Serdes\_Gps\_Eph() [2/3]

`Serdes_Gps_Eph::Serdes_Gps_Eph (const Serdes_Gps_Eph & other) [inline], [noexcept]`

Definition at line 51 of file [serdes\\_gps\\_eph.h](#).

### 10.363.2.4 Serdes\_Gps\_Eph() [3/3]

`Serdes_Gps_Eph::Serdes_Gps_Eph (Serdes_Gps_Eph && other) [inline], [noexcept]`

Definition at line 62 of file [serdes\\_gps\\_eph.h](#).

## 10.363.3 Member Function Documentation

### 10.363.3.1 createProtobuffer()

`std::string Serdes_Gps_Eph::createProtobuffer (const std::shared_ptr< Gps_Ephemeris > monitor) [inline]`

#### Parameters

<i>monitor</i>	Serialization into a string
----------------	-----------------------------

Definition at line 72 of file [serdes\\_gps\\_eph.h](#).

### 10.363.3.2 operator=() [1/2]

`Serdes_Gps_Eph & Serdes_Gps_Eph::operator= (const Serdes_Gps_Eph & rhs) [inline], [noexcept]`

< Copy assignment operator

Definition at line 55 of file [serdes\\_gps\\_eph.h](#).

### 10.363.3.3 operator=() [2/2]

`Serdes_Gps_Eph & Serdes_Gps_Eph::operator= (Serdes_Gps_Eph && other) [inline], [noexcept]`

< Move assignment operator

Definition at line 66 of file [serdes\\_gps\\_eph.h](#).

### 10.363.3.4 readProtobuffer()

`Gps_Ephemeris Serdes_Gps_Eph::readProtobuffer (const gnss_sdr::GpsEphemeris & mon) const [inline]`

< Deserialization

Definition at line 124 of file [serdes\\_gps\\_eph.h](#).

References [Gnss\\_Ephemeris::af0](#), [Gnss\\_Ephemeris::af1](#), [Gnss\\_Ephemeris::af2](#), [Gps\\_Ephemeris::alert\\_flag](#), [Gps\\_Ephemeris::antispoofing\\_flag](#), [Gps\\_Ephemeris::AODO](#), [Gnss\\_Ephemeris::Cic](#), [Gnss\\_Ephemeris::Cis](#), [Gps\\_Ephemeris::code\\_on\\_L2](#), [Gnss\\_Ephemeris::Crc](#), [Gnss\\_Ephemeris::Crs](#), [Gnss\\_Ephemeris::Cuc](#), [Gnss\\_Ephemeris::Cus](#),

Gnss\_Ephemeris::delta\_n, Gnss\_Ephemeris::dtr, Gnss\_Ephemeris::ecc, Gps\_Ephemeris::fit\_interval\_flag, Gnss\_Ephemeris::i\_0, Gnss\_Ephemeris::idot, Gps\_Ephemeris::integrity\_status\_flag, Gps\_Ephemeris::IODC, Gps\_Ephemeris::IODE\_SF2, Gps\_Ephemeris::IODE\_SF3, Gps\_Ephemeris::L2\_P\_data\_flag, Gnss\_Ephemeris::M\_0, Gnss\_Ephemeris::omega, Gnss\_Ephemeris::OMEGA\_0, Gnss\_Ephemeris::OMEGAdot, Gnss\_Ephemeris::PRN, Gnss\_Ephemeris::satClkDrift, Gnss\_Ephemeris::sqrtA, Gps\_Ephemeris::SV\_accuracy, Gps\_Ephemeris::SV\_health, Gps\_Ephemeris::TGD, Gnss\_Ephemeris::toc, Gnss\_Ephemeris::toe, Gnss\_Ephemeris::tow, and Gnss\_Ephemeris::WN.

The documentation for this class was generated from the following file:

- [serdes\\_gps\\_eph.h](#)

## 10.364 Serdes\_Monitor\_Pvt Class Reference

This class implements serialization and deserialization of [Monitor\\_Pvt](#) objects using Protocol Buffers.

```
#include <serdes_monitor_pvt.h>
```

Collaboration diagram for Serdes\_Monitor\_Pvt:

Serdes_Monitor_Pvt
<ul style="list-style-type: none"> <li>+ Serdes_Monitor_Pvt()</li> <li>+ ~Serdes_Monitor_Pvt()</li> <li>+ Serdes_Monitor_Pvt()</li> <li>+ operator=()</li> <li>+ Serdes_Monitor_Pvt()</li> <li>+ operator=()</li> <li>+ createProtobuffer()</li> <li>+ readProtobuffer()</li> </ul>

### Public Member Functions

- [Serdes\\_Monitor\\_Pvt](#) (const [Serdes\\_Monitor\\_Pvt](#) &other) noexcept
- [Serdes\\_Monitor\\_Pvt](#) & operator= (const [Serdes\\_Monitor\\_Pvt](#) &rhs) noexcept  
*< Copy assignment operator*
- [Serdes\\_Monitor\\_Pvt](#) ([Serdes\\_Monitor\\_Pvt](#) &&other) noexcept
- [Serdes\\_Monitor\\_Pvt](#) & operator= ([Serdes\\_Monitor\\_Pvt](#) &&other) noexcept  
*< Move assignment operator*
- std::string [createProtobuffer](#) (const [Monitor\\_Pvt](#) \*const monitor)
- [Monitor\\_Pvt](#) [readProtobuffer](#) (const gnss\_sdr::MonitorPvt &mon) const  
*< Deserialization*

### 10.364.1 Detailed Description

This class implements serialization and deserialization of [Monitor\\_Pvt](#) objects using Protocol Buffers.

Definition at line 37 of file [serdes\\_monitor\\_pvt.h](#).

## 10.364.2 Constructor & Destructor Documentation

### 10.364.2.1 Serdes\_Monitor\_Pvt() [1/3]

`Serdes_Monitor_Pvt::Serdes_Monitor_Pvt () [inline]`

Definition at line 40 of file [serdes\\_monitor\\_pvt.h](#).

### 10.364.2.2 ~Serdes\_Monitor\_Pvt()

`Serdes_Monitor_Pvt::~~Serdes_Monitor_Pvt () [inline]`

Definition at line 47 of file [serdes\\_monitor\\_pvt.h](#).

### 10.364.2.3 Serdes\_Monitor\_Pvt() [2/3]

`Serdes_Monitor_Pvt::Serdes_Monitor_Pvt (const Serdes_Monitor_Pvt & other) [inline], [noexcept]`

Definition at line 52 of file [serdes\\_monitor\\_pvt.h](#).

### 10.364.2.4 Serdes\_Monitor\_Pvt() [3/3]

`Serdes_Monitor_Pvt::Serdes_Monitor_Pvt (Serdes_Monitor_Pvt && other) [inline], [noexcept]`

Definition at line 65 of file [serdes\\_monitor\\_pvt.h](#).

## 10.364.3 Member Function Documentation

### 10.364.3.1 createProtobuffer()

`std::string Serdes_Monitor_Pvt::createProtobuffer (const Monitor_Pvt *const monitor) [inline]`

#### Parameters

<i>monitor</i>	Serialization into a string
----------------	-----------------------------

Definition at line 80 of file [serdes\\_monitor\\_pvt.h](#).

### 10.364.3.2 operator=() [1/2]

`Serdes_Monitor_Pvt & Serdes_Monitor_Pvt::operator= (const Serdes_Monitor_Pvt & rhs) [inline], [noexcept]`

< Copy assignment operator

Definition at line 56 of file [serdes\\_monitor\\_pvt.h](#).

### 10.364.3.3 operator=() [2/2]

`Serdes_Monitor_Pvt & Serdes_Monitor_Pvt::operator= (Serdes_Monitor_Pvt && other) [inline], [noexcept]`

< Move assignment operator

Definition at line 71 of file [serdes\\_monitor\\_pvt.h](#).

### 10.364.3.4 readProtobuffer()

`Monitor_Pvt Serdes_Monitor_Pvt::readProtobuffer (const gnss_sdr::MonitorPvt & mon) const [inline]`

< Deserialization

Definition at line 127 of file [serdes\\_monitor\\_pvt.h](#).

The documentation for this class was generated from the following file:

- [serdes\\_monitor\\_pvt.h](#)

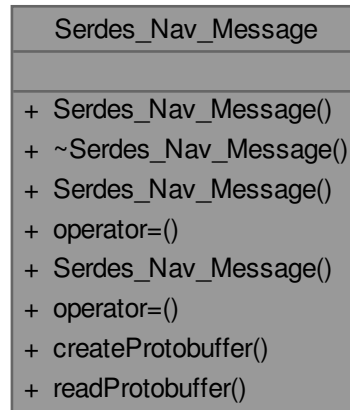


## 10.365 Serdes\_Nav\_Message Class Reference

This class implements serialization and deserialization of [Nav\\_Message\\_Packet](#) objects using Protocol Buffers.

#include <serdes\_nav\_message.h>

Collaboration diagram for Serdes\_Nav\_Message:



### Public Member Functions

- [Serdes\\_Nav\\_Message](#) (const [Serdes\\_Nav\\_Message](#) &other) noexcept
- [Serdes\\_Nav\\_Message](#) & operator= (const [Serdes\\_Nav\\_Message](#) &rhs) noexcept  
 < Copy assignment operator
- [Serdes\\_Nav\\_Message](#) ([Serdes\\_Nav\\_Message](#) &&other) noexcept
- [Serdes\\_Nav\\_Message](#) & operator= ([Serdes\\_Nav\\_Message](#) &&other) noexcept  
 < Move assignment operator
- std::string [createProtobuffer](#) (const std::shared\_ptr< [Nav\\_Message\\_Packet](#) > nav\_msg\_packet)
- [Nav\\_Message\\_Packet](#) [readProtobuffer](#) (const gnss\_sdr::navMsg &msg) const  
 < Deserialization

### 10.365.1 Detailed Description

This class implements serialization and deserialization of [Nav\\_Message\\_Packet](#) objects using Protocol Buffers. Definition at line 38 of file [serdes\\_nav\\_message.h](#).

### 10.365.2 Constructor & Destructor Documentation

#### 10.365.2.1 Serdes\_Nav\_Message() [1/3]

`Serdes_Nav_Message::Serdes_Nav_Message () [inline]`

Definition at line 41 of file [serdes\\_nav\\_message.h](#).

#### 10.365.2.2 ~Serdes\_Nav\_Message()

`Serdes_Nav_Message::~~Serdes_Nav_Message () [inline]`

Definition at line 48 of file [serdes\\_nav\\_message.h](#).

**10.365.2.3 Serdes\_Nav\_Message() [2/3]**

```
Serdes_Nav_Message::Serdes_Nav_Message (
    const Serdes\_Nav\_Message & other) [inline], [noexcept]
```

Definition at line 53 of file [serdes\\_nav\\_message.h](#).

**10.365.2.4 Serdes\_Nav\_Message() [3/3]**

```
Serdes_Nav_Message::Serdes_Nav_Message (
    Serdes\_Nav\_Message && other) [inline], [noexcept]
```

Definition at line 66 of file [serdes\\_nav\\_message.h](#).

**10.365.3 Member Function Documentation****10.365.3.1 createProtobuffer()**

```
std::string Serdes_Nav_Message::createProtobuffer (
    const std::shared_ptr< Nav\_Message\_Packet > nav_msg_packet) [inline]
```

**Parameters**

<a href="#">nav_msg_packet</a>	Serialization into a string
--------------------------------	-----------------------------

Definition at line 82 of file [serdes\\_nav\\_message.h](#).

**10.365.3.2 operator=() [1/2]**

```
Serdes\_Nav\_Message & Serdes_Nav_Message::operator= (
    const Serdes\_Nav\_Message & rhs) [inline], [noexcept]
```

< Copy assignment operator

Definition at line 57 of file [serdes\\_nav\\_message.h](#).

**10.365.3.3 operator=() [2/2]**

```
Serdes\_Nav\_Message & Serdes_Nav_Message::operator= (
    Serdes\_Nav\_Message && other) [inline], [noexcept]
```

< Move assignment operator

Definition at line 72 of file [serdes\\_nav\\_message.h](#).

**10.365.3.4 readProtobuffer()**

```
Nav\_Message\_Packet Serdes_Nav_Message::readProtobuffer (
    const gnss_sdr::navMsg & msg) const [inline]
```

< Deserialization

Definition at line 98 of file [serdes\\_nav\\_message.h](#).

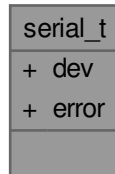
References [Nav\\_Message\\_Packet::nav\\_message](#), [Nav\\_Message\\_Packet::prn](#), [Nav\\_Message\\_Packet::signal](#), [Nav\\_Message\\_Packet::system](#), and [Nav\\_Message\\_Packet::tow\\_at\\_current\\_symbol\\_ms](#).

The documentation for this class was generated from the following file:

- [serdes\\_nav\\_message.h](#)

## 10.366 serial\_t Struct Reference

Collaboration diagram for serial\_t:



### Public Attributes

- dev\_t [dev](#)
- int [error](#)

### 10.366.1 Detailed Description

Definition at line [1110](#) of file [rtklib.h](#).

### 10.366.2 Member Data Documentation

#### 10.366.2.1 dev

```
dev_t serial_t::dev
```

Definition at line [1112](#) of file [rtklib.h](#).

#### 10.366.2.2 error

```
int serial_t::error
```

Definition at line [1113](#) of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

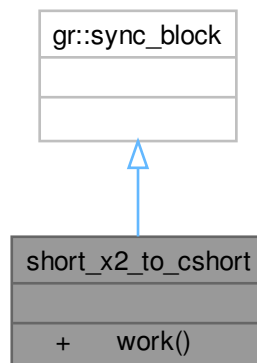
- [rtklib.h](#)

## 10.367 short\_x2\_to\_cshort Class Reference

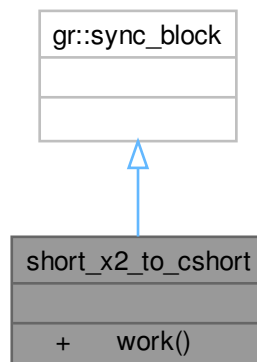
This class adapts two short streams into a `std::complex<short>` stream.

```
#include <short_x2_to_cshort.h>
```

Inheritance diagram for `short_x2_to_cshort`:



Collaboration diagram for `short_x2_to_cshort`:



### Public Member Functions

- `int` **work** (`int` noutput\_items, `gr_vector_const_void_star` &input\_items, `gr_vector_void_star` &output\_items)

### Friends

- `short_x2_to_cshort_sptr` **make\_short\_x2\_to\_cshort** ()

### 10.367.1 Detailed Description

This class adapts two short streams into a `std::complex<short>` stream.

Definition at line 40 of file [short\\_x2\\_to\\_cshort.h](#).

The documentation for this class was generated from the following file:

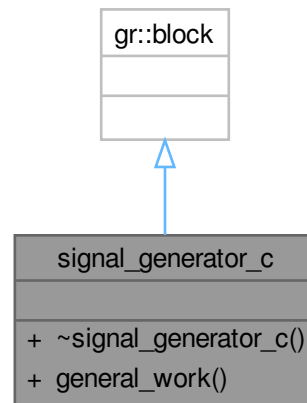
- [short\\_x2\\_to\\_cshort.h](#)

## 10.368 signal\_generator\_c Class Reference

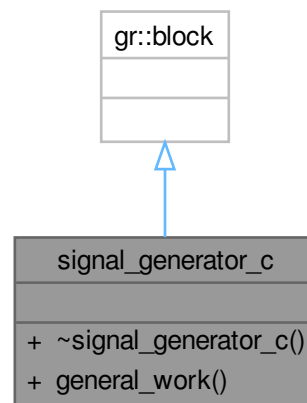
This class generates synthesized GNSS signal.

```
#include <signal_generator_c.h>
```

Inheritance diagram for signal\_generator\_c:



Collaboration diagram for signal\_generator\_c:



### Public Member Functions

- `int` **general\_work** (`int` noutput\_items, `gr_vector_int` &ninput\_items, `gr_vector_const_void_star` &input\_items, `gr_vector_void_star` &output\_items)

### Friends

- `signal_generator_c_sptr` [signal\\_make\\_generator\\_c](#) (`const` `std::vector`< `std::string` > &signal1, `const` `std::vector`< `std::string` > &system, `const` `std::vector`< `unsigned int` > &PRN, `const` `std::vector`< `float` > &CN0↵

```
_dB, const std::vector< float > &doppler_Hz, const std::vector< unsigned int > &delay_chips, const std::vector< unsigned int > &delay_sec, bool data_flag, bool noise_flag, unsigned int fs_in, unsigned int vector_length, float BW_BB)
```

*Return a shared\_ptr to a new instance of gen\_source.*

### 10.368.1 Detailed Description

This class generates synthesized GNSS signal.

See also

[gen\\_source](#) for a version that subclasses gr\_block.

Definition at line 58 of file [signal\\_generator\\_c.h](#).

### 10.368.2 Friends And Related Symbol Documentation

#### 10.368.2.1 signal\_make\_generator\_c

```
signal_generator_c_sptr signal_make_generator_c (
    const std::vector< std::string > & signal1,
    const std::vector< std::string > & system,
    const std::vector< unsigned int > & PRN,
    const std::vector< float > & CNO_dB,
    const std::vector< float > & doppler_Hz,
    const std::vector< unsigned int > & delay_chips,
    const std::vector< unsigned int > & delay_sec,
    bool data_flag,
    bool noise_flag,
    unsigned int fs_in,
    unsigned int vector_length,
    float BW_BB) [friend]
```

Return a shared\_ptr to a new instance of gen\_source.

To avoid accidental use of raw pointers, gen\_source's constructor is private. signal\_make\_generator\_c is the public interface for creating new instances.

The documentation for this class was generated from the following file:

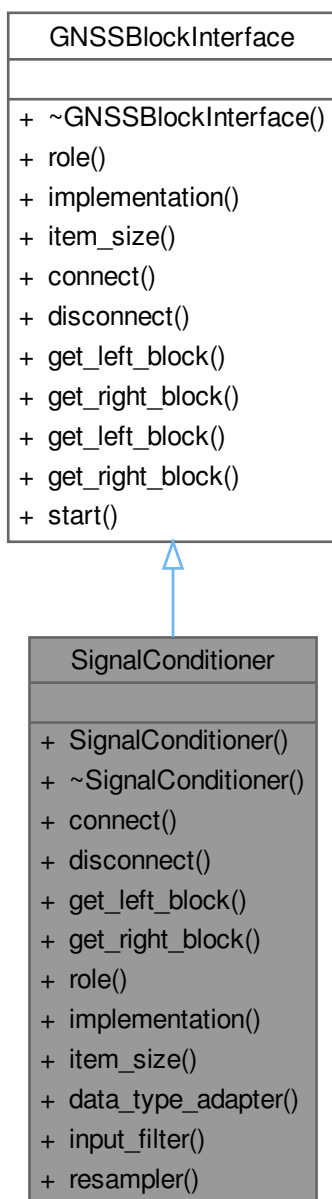
- [signal\\_generator\\_c.h](#)

## 10.369 SignalConditioner Class Reference

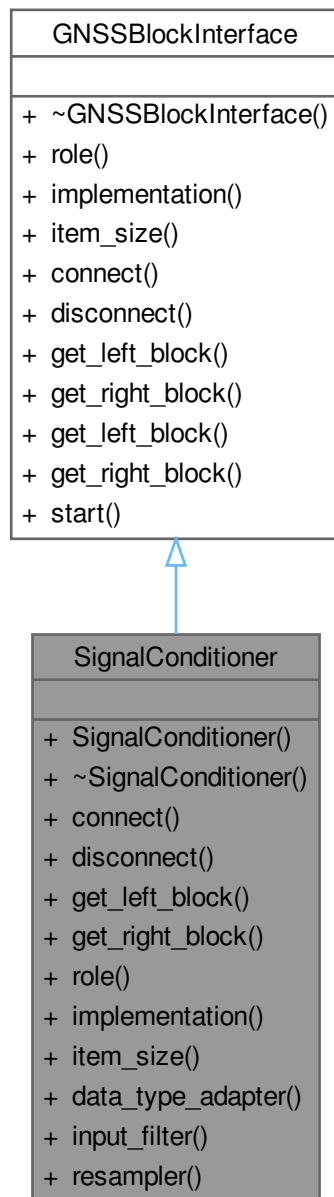
This class wraps blocks to change data\_type\_adapter, input\_filter and resampler to be applied to the input flow of sampled signal.

```
#include <signal_conditioner.h>
```

Inheritance diagram for SignalConditioner:



Collaboration diagram for SignalConditioner:



### Public Member Functions

- [SignalConditioner](#) (std::shared\_ptr< [GNSSBlockInterface](#) > data\_type\_adapt, std::shared\_ptr< [GNSSBlockInterface](#) > in\_filt, std::shared\_ptr< [GNSSBlockInterface](#) > res, std::string role)  
*Constructor.*
- [~SignalConditioner](#) ()=default  
*Destructor.*
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override



- `gr::basic_block_sptr` [get\\_right\\_block](#) () override
- `std::string` [role](#) () override
- `std::string` [implementation](#) () override
 

*Returns "Signal\_Conditioner".*
- `size_t` [item\\_size](#) () override
- `std::shared_ptr< GNSSBlockInterface >` [data\\_type\\_adapter](#) ()
- `std::shared_ptr< GNSSBlockInterface >` [input\\_filter](#) ()
- `std::shared_ptr< GNSSBlockInterface >` [resampler](#) ()

## Public Member Functions inherited from GNSSBlockInterface

- virtual `gr::basic_block_sptr` [get\\_left\\_block](#) (int RF\_channel)
- virtual `gr::basic_block_sptr` [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.369.1 Detailed Description

This class wraps blocks to change `data_type_adapter`, `input_filter` and `resampler` to be applied to the input flow of sampled signal.

Definition at line 39 of file [signal\\_conditioner.h](#).

### 10.369.2 Constructor & Destructor Documentation

#### 10.369.2.1 SignalConditioner()

```
SignalConditioner::SignalConditioner (
    std::shared_ptr< GNSSBlockInterface > data_type_adapt,
    std::shared_ptr< GNSSBlockInterface > in_filt,
    std::shared_ptr< GNSSBlockInterface > res,
    std::string role)
```

Constructor.

#### 10.369.2.2 ~SignalConditioner()

```
SignalConditioner::~SignalConditioner () [default]
```

Destructor.

### 10.369.3 Member Function Documentation

#### 10.369.3.1 connect()

```
void SignalConditioner::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.369.3.2 data\_type\_adapter()

```
std::shared_ptr< GNSSBlockInterface > SignalConditioner::data_type_adapter () [inline]
```

Definition at line 62 of file [signal\\_conditioner.h](#).

#### 10.369.3.3 disconnect()

```
void SignalConditioner::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

**10.369.3.4 get\_left\_block()**

gr::basic\_block\_sptr SignalConditioner::get\_left\_block () [override], [virtual]  
 Implements [GNSSBlockInterface](#).

**10.369.3.5 get\_right\_block()**

gr::basic\_block\_sptr SignalConditioner::get\_right\_block () [override], [virtual]  
 Implements [GNSSBlockInterface](#).

**10.369.3.6 implementation()**

std::string SignalConditioner::implementation () [inline], [override], [virtual]  
 Returns "Signal\_Conditioner".  
 Implements [GNSSBlockInterface](#).  
 Definition at line 58 of file [signal\\_conditioner.h](#).

**10.369.3.7 input\_filter()**

std::shared\_ptr< [GNSSBlockInterface](#) > SignalConditioner::input\_filter () [inline]  
 Definition at line 63 of file [signal\\_conditioner.h](#).

**10.369.3.8 item\_size()**

size\_t SignalConditioner::item\_size () [inline], [override], [virtual]  
 Implements [GNSSBlockInterface](#).  
 Definition at line 60 of file [signal\\_conditioner.h](#).

**10.369.3.9 resampler()**

std::shared\_ptr< [GNSSBlockInterface](#) > SignalConditioner::resampler () [inline]  
 Definition at line 64 of file [signal\\_conditioner.h](#).

**10.369.3.10 role()**

std::string SignalConditioner::role () [inline], [override], [virtual]  
 Implements [GNSSBlockInterface](#).  
 Definition at line 56 of file [signal\\_conditioner.h](#).

The documentation for this class was generated from the following file:

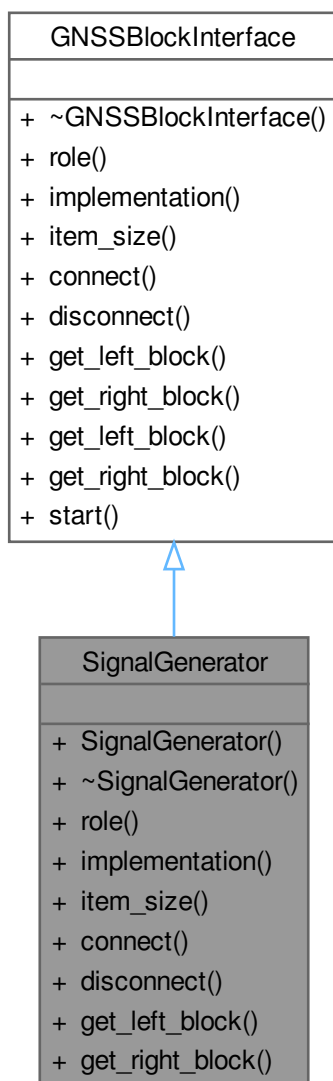
- [signal\\_conditioner.h](#)

**10.370 SignalGenerator Class Reference**

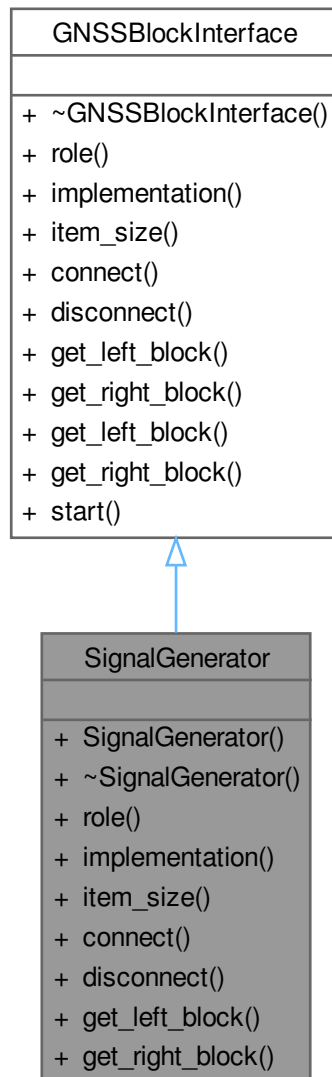
This class generates synthesized GNSS signal.

```
#include <signal_generator.h>
```

Inheritance diagram for SignalGenerator:



Collaboration diagram for SignalGenerator:



### Public Member Functions

- **SignalGenerator** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_stream, unsigned int out\_stream, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)
- std::string [role](#) () override
- std::string [implementation](#) () override  
Returns "GNSSSignalGenerator".
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override

## Public Member Functions inherited from GNSSBlockInterface

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.370.1 Detailed Description

This class generates synthesized GNSS signal.  
Definition at line 39 of file [signal\\_generator.h](#).

### 10.370.2 Member Function Documentation

#### 10.370.2.1 connect()

```
void SignalGenerator::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.370.2.2 disconnect()

```
void SignalGenerator::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.370.2.3 get\_left\_block()

```
gr::basic_block_sptr SignalGenerator::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.370.2.4 get\_right\_block()

```
gr::basic_block_sptr SignalGenerator::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.370.2.5 implementation()

```
std::string SignalGenerator::implementation () [inline], [override], [virtual]
```

Returns "GNSSSignalGenerator".

Implements [GNSSBlockInterface](#).

Definition at line 56 of file [signal\\_generator.h](#).

#### 10.370.2.6 item\_size()

```
size_t SignalGenerator::item_size () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

Definition at line 61 of file [signal\\_generator.h](#).

#### 10.370.2.7 role()

```
std::string SignalGenerator::role () [inline], [override], [virtual]
```

Implements [GNSSBlockInterface](#).

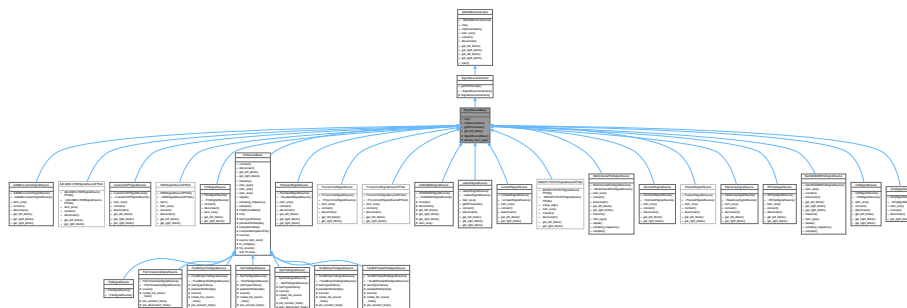
Definition at line 48 of file [signal\\_generator.h](#).

The documentation for this class was generated from the following file:

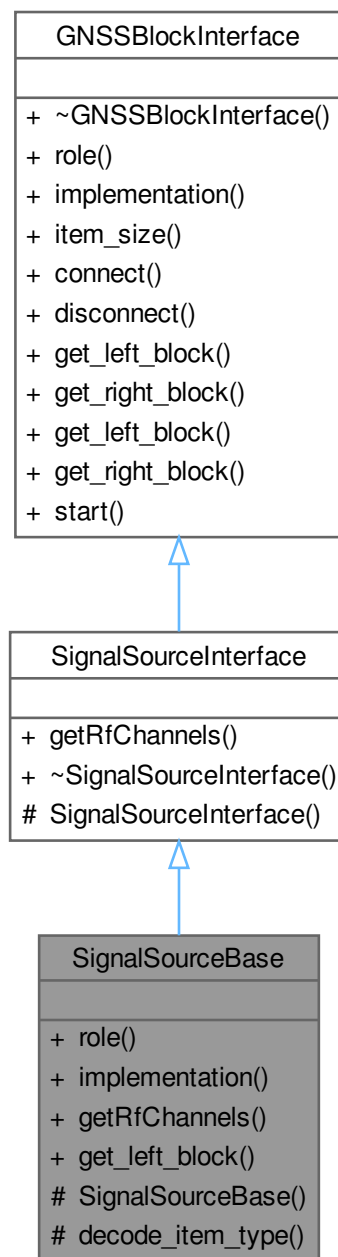
- [signal\\_generator.h](#)

## 10.371 SignalSourceBase Class Reference

Inheritance diagram for SignalSourceBase:



Collaboration diagram for SignalSourceBase:



### Public Member Functions

- `std::string role ()` final
- `std::string implementation ()` final
- `size_t getRfChannels ()` const override
- `gr::basic_block_sptr get_left_block ()` override

## Public Member Functions inherited from GNSSBlockInterface

- virtual `size_t item_size ()`=0
- virtual void `connect (gr::top_block_sptr top_block)`=0
- virtual void `disconnect (gr::top_block_sptr top_block)`=0
- virtual `gr::basic_block_sptr get_right_block ()`=0
- virtual `gr::basic_block_sptr get_left_block (int RF_channel)`
- virtual `gr::basic_block_sptr get_right_block (int RF_channel)`
- virtual void `start ()`

*Start the flow of samples if needed.*

## Protected Member Functions

- `SignalSourceBase (ConfigurationInterface const *configuration, std::string role, std::string impl)`  
*Constructor.*
- `size_t decode_item_type (std::string const &item_type, bool *is_interleaved=nullptr, bool throw_on_error=false)`  
*utility for decoding passed ".item\_type" values*

### 10.371.1 Detailed Description

Definition at line 28 of file [signal\\_source\\_base.h](#).

### 10.371.2 Constructor & Destructor Documentation

#### 10.371.2.1 SignalSourceBase()

```
SignalSourceBase::SignalSourceBase (
    ConfigurationInterface const * configuration,
    std::string role,
    std::string impl) [protected]
```

Constructor.

### 10.371.3 Member Function Documentation

#### 10.371.3.1 decode\_item\_type()

```
size_t SignalSourceBase::decode_item_type (
    std::string const & item_type,
    bool * is_interleaved = nullptr,
    bool throw_on_error = false) [protected]
```

utility for decoding passed ".item\_type" values

#### Parameters

in	<i>item_type</i>	- user provided string, should be one of the known types
out	<i>is_interleaved</i>	- if non-null, the pointed to memory is updated with whether the data is interleaved I/Q (e.g., ishort)
in	<i>throw_on_error</i>	- if true, throw an exception if the string does not represent a known type

#### Returns

the size in bytes of the passed type

#### 10.371.3.2 get\_left\_block()

```
gr::basic_block_sptr SignalSourceBase::get_left_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).



### 10.371.3.3 getRfChannels()

```
size_t SignalSourceBase::getRfChannels () const [override], [virtual]
```

Implements [SignalSourceInterface](#).

#### 10.371.3.4 implementation()

```
std::string SignalSourceBase::implementation () [final], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.371.3.5 role()

```
std::string SignalSourceBase::role () [final], [virtual]
```

Implements [GNSSBlockInterface](#).

The documentation for this class was generated from the following file:

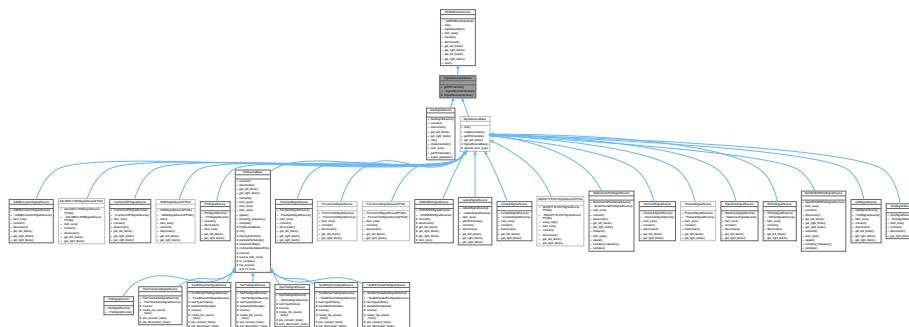
- `signal_source_base.h`

## 10.372 SignalSourceInterface Class Reference

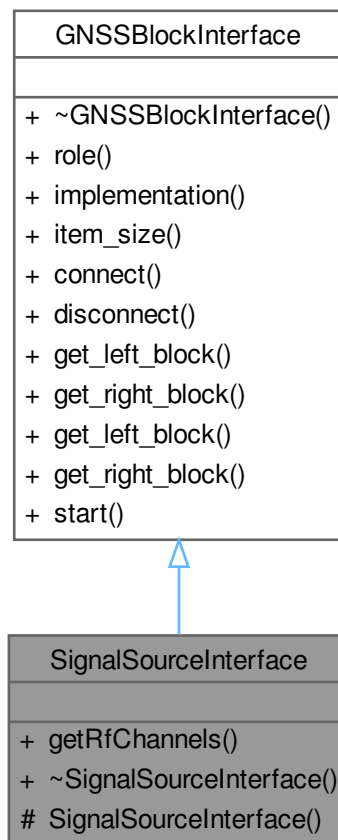
This abstract class represents an interface to signal\_source GNSS block.

```
#include <signal_source_interface.h>
```

Inheritance diagram for SignalSourceInterface:



Collaboration diagram for SignalSourceInterface:



### Public Member Functions

- virtual `size_t` **getRfChannels** () const =0

### Public Member Functions inherited from **GNSSBlockInterface**

- virtual `std::string` **role** ()=0
- virtual `std::string` **implementation** ()=0
- virtual `size_t` **item\_size** ()=0
- virtual void **connect** (gr::top\_block\_sptr top\_block)=0
- virtual void **disconnect** (gr::top\_block\_sptr top\_block)=0
- virtual gr::basic\_block\_sptr **get\_left\_block** ()=0
- virtual gr::basic\_block\_sptr **get\_right\_block** ()=0
- virtual gr::basic\_block\_sptr **get\_left\_block** (int RF\_channel)
- virtual gr::basic\_block\_sptr **get\_right\_block** (int RF\_channel)
- virtual void **start** ()

*Start the flow of samples if needed.*

### 10.372.1 Detailed Description

This abstract class represents an interface to signal\_source GNSS block.

Abstract class for signal sources. Since all its methods are virtual, this class cannot be instantiated directly, and a subclass can only be instantiated directly if all inherited pure virtual methods have been implemented by that class or a parent class.

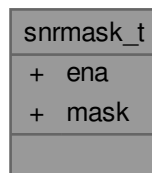
Definition at line 51 of file [signal\\_source\\_interface.h](#).

The documentation for this class was generated from the following file:

- [signal\\_source\\_interface.h](#)

## 10.373 snrmask\_t Struct Reference

Collaboration diagram for snrmask\_t:



### Public Attributes

- int [ena](#) [2]
- double [mask](#) [[NFREQ](#)][9]

### 10.373.1 Detailed Description

Definition at line 935 of file [rtklib.h](#).

### 10.373.2 Member Data Documentation

#### 10.373.2.1 ena

int snrmask\_t::ena[2]

Definition at line 937 of file [rtklib.h](#).

#### 10.373.2.2 mask

double snrmask\_t::mask[[NFREQ](#)][9]

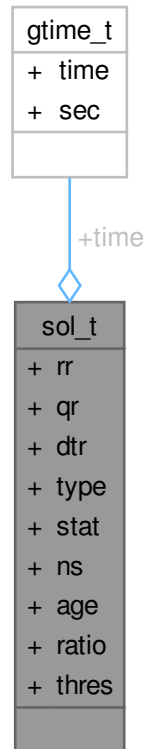
Definition at line 938 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.374 sol\_t Struct Reference

Collaboration diagram for sol\_t:



### Public Attributes

- [gtime\\_t time](#)
- double [rr](#) [6]
- float [qr](#) [6]
- double [dtr](#) [6]
- unsigned char [type](#)
- unsigned char [stat](#)
- unsigned char [ns](#)
- float [age](#)
- float [ratio](#)
- float [thres](#)

### 10.374.1 Detailed Description

Definition at line [819](#) of file [rtklib.h](#).

### 10.374.2 Member Data Documentation

#### 10.374.2.1 age

`float sol_t::age`

Definition at line [831](#) of file [rtklib.h](#).

#### 10.374.2.2 dtr

```
double sol_t::dtr[6]
```

Definition at line 827 of file [rtklib.h](#).

#### 10.374.2.3 ns

```
unsigned char sol_t::ns
```

Definition at line 830 of file [rtklib.h](#).

#### 10.374.2.4 qr

```
float sol_t::qr[6]
```

Definition at line 824 of file [rtklib.h](#).

#### 10.374.2.5 ratio

```
float sol_t::ratio
```

Definition at line 832 of file [rtklib.h](#).

#### 10.374.2.6 rr

```
double sol_t::rr[6]
```

Definition at line 822 of file [rtklib.h](#).

#### 10.374.2.7 stat

```
unsigned char sol_t::stat
```

Definition at line 829 of file [rtklib.h](#).

#### 10.374.2.8 thres

```
float sol_t::thres
```

Definition at line 833 of file [rtklib.h](#).

#### 10.374.2.9 time

```
gtime_t sol_t::time
```

Definition at line 821 of file [rtklib.h](#).

#### 10.374.2.10 type

```
unsigned char sol_t::type
```

Definition at line 828 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.375 solbuf\_t Struct Reference

Collaboration diagram for solbuf\_t:



### Public Attributes

- int `n`
- int `nmax`
- int `cyclic`
- int `start`
- int `end`

- [gtime\\_t](#) time
- [sol\\_t](#) \* data
- double [rb](#) [3]
- unsigned char [buff](#) [MAXSOLMSG+1]
- int [nb](#)

### 10.375.1 Detailed Description

Definition at line 837 of file [rtklib.h](#).

### 10.375.2 Member Data Documentation

#### 10.375.2.1 buff

```
unsigned char solbuf_t::buff[MAXSOLMSG+1]
```

Definition at line 845 of file [rtklib.h](#).

#### 10.375.2.2 cyclic

```
int solbuf_t::cyclic
```

Definition at line 840 of file [rtklib.h](#).

#### 10.375.2.3 data

```
sol\_t* solbuf_t::data
```

Definition at line 843 of file [rtklib.h](#).

#### 10.375.2.4 end

```
int solbuf_t::end
```

Definition at line 841 of file [rtklib.h](#).

#### 10.375.2.5 n

```
int solbuf_t::n
```

Definition at line 839 of file [rtklib.h](#).

#### 10.375.2.6 nb

```
int solbuf_t::nb
```

Definition at line 846 of file [rtklib.h](#).

#### 10.375.2.7 nmax

```
int solbuf_t::nmax
```

Definition at line 839 of file [rtklib.h](#).

#### 10.375.2.8 rb

```
double solbuf_t::rb[3]
```

Definition at line 844 of file [rtklib.h](#).

#### 10.375.2.9 start

```
int solbuf_t::start
```

Definition at line 841 of file [rtklib.h](#).

### 10.375.2.10 time

`gtime_t solbuf_t::time`

Definition at line 842 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.376 solopt\_t Struct Reference

Collaboration diagram for `solo_t`:

solo_t
+ posf
+ times
+ timef
+ timeu
+ degf
+ outhead
+ outopt
+ datum
+ height
+ geoid
and 7 more...

### Public Attributes

- int [posf](#)
- int [times](#)
- int [timef](#)
- int [timeu](#)
- int [degf](#)
- int [outhead](#)
- int [outopt](#)
- int [datum](#)
- int [height](#)
- int [geoid](#)
- int [solstatic](#)
- int [sstat](#)
- int [trace](#)
- double [nmeaintv](#) [2]
- char [sep](#) [64]
- char [prog](#) [64]
- double [maxsolstd](#)



## 10.376.1 Detailed Description

Definition at line 1007 of file [rtklib.h](#).

## 10.376.2 Member Data Documentation

### 10.376.2.1 datum

```
int solopt_t::datum
```

Definition at line 1016 of file [rtklib.h](#).

### 10.376.2.2 degf

```
int solopt_t::degf
```

Definition at line 1013 of file [rtklib.h](#).

### 10.376.2.3 geoid

```
int solopt_t::geoid
```

Definition at line 1018 of file [rtklib.h](#).

### 10.376.2.4 height

```
int solopt_t::height
```

Definition at line 1017 of file [rtklib.h](#).

### 10.376.2.5 maxsolstd

```
double solopt_t::maxsolstd
```

Definition at line 1026 of file [rtklib.h](#).

### 10.376.2.6 nmeaintv

```
double solopt_t::nmeaintv[2]
```

Definition at line 1022 of file [rtklib.h](#).

### 10.376.2.7 outhead

```
int solopt_t::outhead
```

Definition at line 1014 of file [rtklib.h](#).

### 10.376.2.8 outopt

```
int solopt_t::outopt
```

Definition at line 1015 of file [rtklib.h](#).

### 10.376.2.9 posf

```
int solopt_t::posf
```

Definition at line 1009 of file [rtklib.h](#).

### 10.376.2.10 prog

```
char solopt_t::prog[64]
```

Definition at line 1025 of file [rtklib.h](#).

### 10.376.2.11 sep

```
char solopt_t::sep[64]
```

Definition at line 1024 of file [rtklib.h](#).

**10.376.2.12 solstatic**

```
int solopt_t::solstatic
```

Definition at line 1019 of file [rtklib.h](#).

**10.376.2.13 sstat**

```
int solopt_t::sstat
```

Definition at line 1020 of file [rtklib.h](#).

**10.376.2.14 timef**

```
int solopt_t::timef
```

Definition at line 1011 of file [rtklib.h](#).

**10.376.2.15 times**

```
int solopt_t::times
```

Definition at line 1010 of file [rtklib.h](#).

**10.376.2.16 timeu**

```
int solopt_t::timeu
```

Definition at line 1012 of file [rtklib.h](#).

**10.376.2.17 trace**

```
int solopt_t::trace
```

Definition at line 1021 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.377 solstat\_t Struct Reference

Collaboration diagram for solstat\_t:



### Public Attributes

- [gtime\\_t time](#)
- unsigned char [sat](#)
- unsigned char [frq](#)
- float [az](#)
- float [el](#)
- float [resp](#)
- float [resc](#)
- unsigned char [flag](#)
- unsigned char [snr](#)
- unsigned short [lock](#)
- unsigned short [outc](#)
- unsigned short [slipc](#)
- unsigned short [rejc](#)

### 10.377.1 Detailed Description

Definition at line 850 of file [rtklib.h](#).

## 10.377.2 Member Data Documentation

### 10.377.2.1 az

`float solstat_t::az`

Definition at line 855 of file [rtklib.h](#).

### 10.377.2.2 el

`float solstat_t::el`

Definition at line 855 of file [rtklib.h](#).

### 10.377.2.3 flag

`unsigned char solstat_t::flag`

Definition at line 858 of file [rtklib.h](#).

### 10.377.2.4 frq

`unsigned char solstat_t::frq`

Definition at line 854 of file [rtklib.h](#).

### 10.377.2.5 lock

`unsigned short solstat_t::lock`

Definition at line 860 of file [rtklib.h](#).

### 10.377.2.6 outc

`unsigned short solstat_t::outc`

Definition at line 861 of file [rtklib.h](#).

### 10.377.2.7 rejc

`unsigned short solstat_t::rejc`

Definition at line 863 of file [rtklib.h](#).

### 10.377.2.8 resc

`float solstat_t::resc`

Definition at line 857 of file [rtklib.h](#).

### 10.377.2.9 resp

`float solstat_t::resp`

Definition at line 856 of file [rtklib.h](#).

### 10.377.2.10 sat

`unsigned char solstat_t::sat`

Definition at line 853 of file [rtklib.h](#).

### 10.377.2.11 slipc

`unsigned short solstat_t::slipc`

Definition at line 862 of file [rtklib.h](#).

### 10.377.2.12 snr

`unsigned char solstat_t::snr`

Definition at line 859 of file [rtklib.h](#).

### 10.377.2.13 time

`gtime_t solstat_t::time`

Definition at line 852 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.378 solstatbuf\_t Struct Reference

Collaboration diagram for solstatbuf\_t:



## Public Attributes

- int [n](#)
- int [nmax](#)
- [solstat\\_t](#) \* [data](#)

### 10.378.1 Detailed Description

Definition at line [867](#) of file [rtklib.h](#).

### 10.378.2 Member Data Documentation

#### 10.378.2.1 data

```
solstat\_t* solstatbuf_t::data
```

Definition at line [870](#) of file [rtklib.h](#).

#### 10.378.2.2 n

```
int solstatbuf_t::n
```

Definition at line [869](#) of file [rtklib.h](#).

#### 10.378.2.3 nmax

```
int solstatbuf_t::nmax
```

Definition at line [869](#) of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

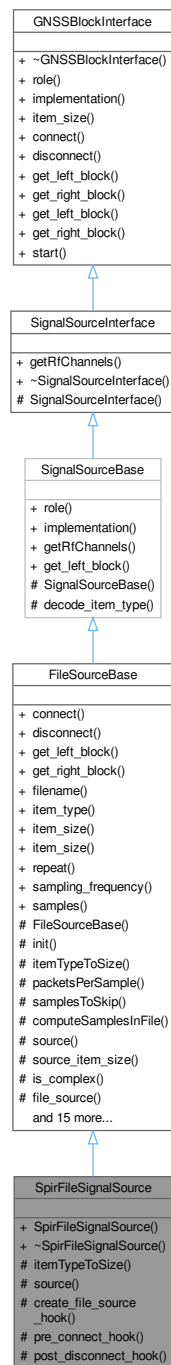
- [rtklib.h](#)

## 10.379 SpirFileSignalSource Class Reference

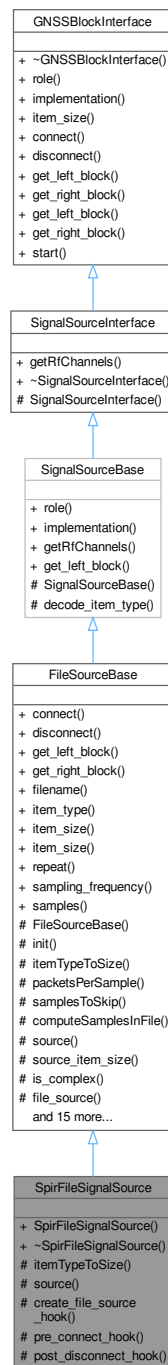
Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).

```
#include <spir_file_signal_source.h>
```

Inheritance diagram for SpirFileSignalSource:



Collaboration diagram for SpirFileSignalSource:



## Public Member Functions

- **SpirFileSignalSource** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)

## Public Member Functions inherited from [FileSourceBase](#)

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override



- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- std::string [filename](#) () const  
*The file to read.*
- std::string [item\\_type](#) () const  
*The item type.*
- size\_t [item\\_size](#) () override  
*The configured size of each item.*
- virtual size\_t [item\\_size](#) () const
- bool [repeat](#) () const  
*Whether to repeat reading after end-of-file.*
- int64\_t [sampling\\_frequency](#) () const  
*The sampling frequency of the source file.*
- uint64\_t [samples](#) () const  
*The number of samples in the file.*

### Public Member Functions inherited from [SignalSourceBase](#)

- std::string [role](#) () final
- std::string [implementation](#) () final
- size\_t [getRfChannels](#) () const override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()  
*Start the flow of samples if needed.*

### Protected Member Functions

- std::tuple< size\_t, bool > [itemTypeToSize](#) () override  
*Compute the item size, from the [item\\_type](#)(). Subclasses may constrain types that don't make.*
- gnss\_shared\_ptr< gr::block > [source](#) () const override  
*Abstracted front-end source. Sub-classes may override if they create specialized chains to decode source files into a usable format.*
- void [create\\_file\\_source\\_hook](#) () override
- void [pre\\_connect\\_hook](#) (gr::top\_block\_sptr top\_block) override
- void [post\\_disconnect\\_hook](#) (gr::top\_block\_sptr top\_block) override

### Protected Member Functions inherited from [FileSourceBase](#)

- [FileSourceBase](#) ([ConfigurationInterface](#) const \*configuration, std::string const &role, std::string impl, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue, std::string default\_item\_type="short")  
*Constructor.*
- void [init](#) ()  
*Perform post-construction initialization.*
- virtual double [packetsPerSample](#) () const  
*The number of (possibly unpacked) samples in a (raw) file sample (default=1)*
- virtual size\_t [samplesToSkip](#) () const  
*Compute the number of samples to skip.*
- size\_t [computeSamplesInFile](#) () const  
*Compute the number of samples in the file.*

- virtual size\_t [source\\_item\\_size](#) () const

*For complex source chains, the size of the file item may not be the same as the size of the.*

- bool **is\_complex** () const
- gnss\_shared\_ptr< gr::block > **file\_source** () const
- gnss\_shared\_ptr< gr::block > **valve** () const
- gnss\_shared\_ptr< gr::block > **throttle** () const
- gnss\_shared\_ptr< gr::block > **sink** () const
- gr::blocks::file\_source::sptr **create\_file\_source** ()
- gr::blocks::throttle::sptr **create\_throttle** ()
- gnss\_shared\_ptr< gr::block > **create\_valve** ()
- gr::blocks::file\_sink::sptr **create\_sink** ()
- virtual void **create\_throttle\_hook** ()
- virtual void **create\_valve\_hook** ()
- virtual void **create\_sink\_hook** ()
- virtual void **post\_connect\_hook** (gr::top\_block\_sptr top\_block)
- virtual void **pre\_disconnect\_hook** (gr::top\_block\_sptr top\_block)

## Protected Member Functions inherited from [SignalSourceBase](#)

- [SignalSourceBase](#) ([ConfigurationInterface](#) const \*configuration, std::string role, std::string impl)  
*Constructor.*
- size\_t [decode\\_item\\_type](#) (std::string const &item\_type, bool \*is\_interleaved=nullptr, bool throw\_on\_↵ error=false)  
*utility for decoding passed ".item\_type" values*

## 10.379.1 Detailed Description

Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).

Definition at line 40 of file [spir\\_file\\_signal\\_source.h](#).

## 10.379.2 Member Function Documentation

### 10.379.2.1 [create\\_file\\_source\\_hook\(\)](#)

void [SpirFileSignalSource::create\\_file\\_source\\_hook](#) () [override], [protected], [virtual]  
Reimplemented from [FileSourceBase](#).

### 10.379.2.2 [itemTypeToSize\(\)](#)

std::tuple< size\_t, bool > [SpirFileSignalSource::itemTypeToSize](#) () [override], [protected], [virtual]

Compute the item size, from the [item\\_type\(\)](#). Subclasses may constrain types that don't make.

Reimplemented from [FileSourceBase](#).

### 10.379.2.3 [post\\_disconnect\\_hook\(\)](#)

void [SpirFileSignalSource::post\\_disconnect\\_hook](#) (  
gr::top\_block\_sptr top\_block) [override], [protected], [virtual]

Reimplemented from [FileSourceBase](#).

### 10.379.2.4 [pre\\_connect\\_hook\(\)](#)

void [SpirFileSignalSource::pre\\_connect\\_hook](#) (  
gr::top\_block\_sptr top\_block) [override], [protected], [virtual]

Reimplemented from [FileSourceBase](#).

#### 10.379.2.5 source()

```
gnss_shared_ptr< gr::block > SpirFileSignalSource::source () const [override], [protected],  
[virtual]
```

Abstracted front-end source. Sub-classes may override if they create specialized chains to decode source files into a usable format.

Reimplemented from [FileSourceBase](#).

The documentation for this class was generated from the following file:

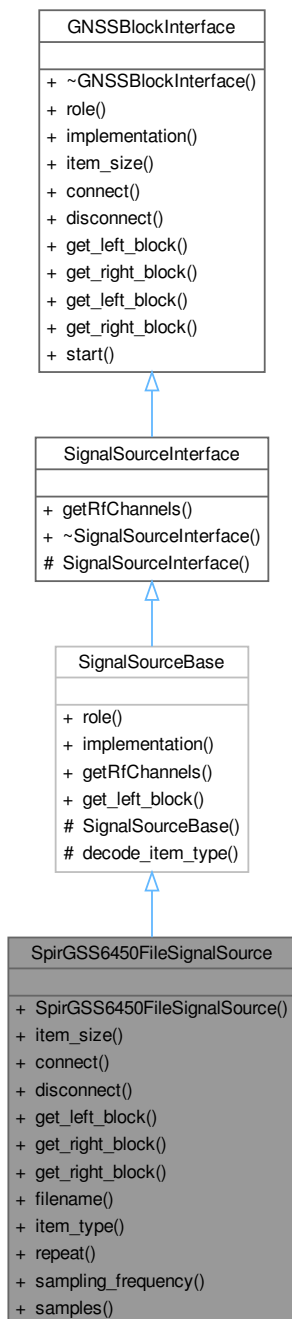
- [spir\\_file\\_signal\\_source.h](#)

## 10.380 SpirGSS6450FileSignalSource Class Reference

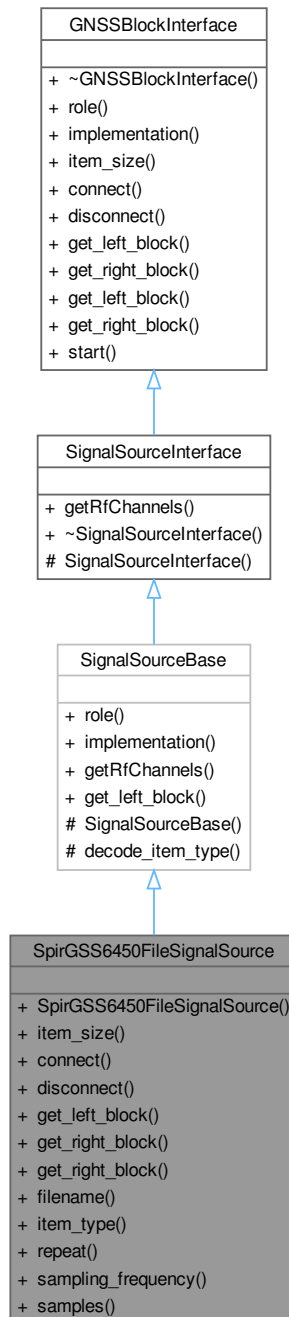
Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).

```
#include <spir_gss6450_file_signal_source.h>
```

Inheritance diagram for SpirGSS6450FileSignalSource:



Collaboration diagram for SpirGSS6450FileSignalSource:



### Public Member Functions

- **SpirGSS6450FileSignalSource** (const [ConfigurationInterface](#) \*configuration, const std::string &role, uint32\_t in\_streams, uint32\_t out\_streams, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override

- gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel) override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- std::string [filename](#) () const
- std::string [item\\_type](#) () const
- bool [repeat](#) () const
- int64\_t [sampling\\_frequency](#) () const
- uint64\_t [samples](#) () const

### Public Member Functions inherited from [SignalSourceBase](#)

- std::string [role](#) () final
- std::string [implementation](#) () final
- size\_t [getRfChannels](#) () const override

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### Additional Inherited Members

### Protected Member Functions inherited from [SignalSourceBase](#)

- [SignalSourceBase](#) ([ConfigurationInterface](#) const \*configuration, std::string role, std::string impl)  
*Constructor.*
- size\_t [decode\\_item\\_type](#) (std::string const &item\_type, bool \*is\_interleaved=nullptr, bool throw\_on\_↔ error=false)  
*utility for decoding passed ".item\_type" values*

## 10.380.1 Detailed Description

Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).  
Definition at line 53 of file [spir\\_gss6450\\_file\\_signal\\_source.h](#).

## 10.380.2 Member Function Documentation

### 10.380.2.1 connect()

```
void SpirGSS6450FileSignalSource::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.380.2.2 disconnect()

```
void SpirGSS6450FileSignalSource::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

### 10.380.2.3 filename()

```
std::string SpirGSS6450FileSignalSource::filename () const [inline]
```

Definition at line 70 of file [spir\\_gss6450\\_file\\_signal\\_source.h](#).

### 10.380.2.4 get\_left\_block()

```
gr::basic_block_sptr SpirGSS6450FileSignalSource::get_left_block () [override], [virtual]
```

Reimplemented from [SignalSourceBase](#).

**10.380.2.5 get\_right\_block()** [1/2]

gr::basic\_block\_sptr SpirGSS6450FileSignalSource::get\_right\_block () [override], [virtual]  
Implements [GNSSBlockInterface](#).

**10.380.2.6 get\_right\_block()** [2/2]

gr::basic\_block\_sptr SpirGSS6450FileSignalSource::get\_right\_block (  
    int *RF\_channel*) [override], [virtual]  
Reimplemented from [GNSSBlockInterface](#).

**10.380.2.7 item\_size()**

size\_t SpirGSS6450FileSignalSource::item\_size () [inline], [override], [virtual]  
Implements [GNSSBlockInterface](#).  
Definition at line 59 of file [spir\\_gss6450\\_file\\_signal\\_source.h](#).

**10.380.2.8 item\_type()**

std::string SpirGSS6450FileSignalSource::item\_type () const [inline]  
Definition at line 75 of file [spir\\_gss6450\\_file\\_signal\\_source.h](#).

**10.380.2.9 repeat()**

bool SpirGSS6450FileSignalSource::repeat () const [inline]  
Definition at line 80 of file [spir\\_gss6450\\_file\\_signal\\_source.h](#).

**10.380.2.10 samples()**

uint64\_t SpirGSS6450FileSignalSource::samples () const [inline]  
Definition at line 90 of file [spir\\_gss6450\\_file\\_signal\\_source.h](#).

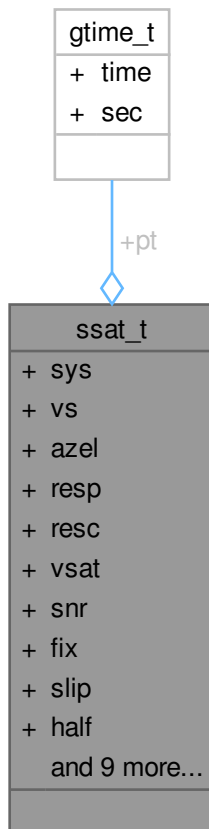
**10.380.2.11 sampling\_frequency()**

int64\_t SpirGSS6450FileSignalSource::sampling\_frequency () const [inline]  
Definition at line 85 of file [spir\\_gss6450\\_file\\_signal\\_source.h](#).  
The documentation for this class was generated from the following file:

- [spir\\_gss6450\\_file\\_signal\\_source.h](#)

## 10.381 ssat\_t Struct Reference

Collaboration diagram for ssat\_t:



### Public Attributes

- unsigned char [sys](#)
- unsigned char [vs](#)
- double [azel](#) [2]
- double [resp](#) [NFREQ]
- double [resc](#) [NFREQ]
- unsigned char [vsat](#) [NFREQ]
- unsigned char [snr](#) [NFREQ]
- unsigned char [fix](#) [NFREQ]
- unsigned char [slip](#) [NFREQ]
- unsigned char [half](#) [NFREQ]
- int [lock](#) [NFREQ]
- unsigned int [outc](#) [NFREQ]
- unsigned int [slipc](#) [NFREQ]
- unsigned int [rejc](#) [NFREQ]
- double [gf](#)
- double [gf2](#)
- double [mw](#)



- double [phw](#)
- [gtime\\_t](#) pt [2][[NFREQ](#)]
- double [ph](#) [2][[NFREQ](#)]

### 10.381.1 Detailed Description

Definition at line [1030](#) of file [rtklib.h](#).

### 10.381.2 Member Data Documentation

#### 10.381.2.1 azel

```
double ssat_t::azel[2]
```

Definition at line [1034](#) of file [rtklib.h](#).

#### 10.381.2.2 fix

```
unsigned char ssat_t::fix[NFREQ]
```

Definition at line [1039](#) of file [rtklib.h](#).

#### 10.381.2.3 gf

```
double ssat_t::gf
```

Definition at line [1046](#) of file [rtklib.h](#).

#### 10.381.2.4 gf2

```
double ssat_t::gf2
```

Definition at line [1047](#) of file [rtklib.h](#).

#### 10.381.2.5 half

```
unsigned char ssat_t::half[NFREQ]
```

Definition at line [1041](#) of file [rtklib.h](#).

#### 10.381.2.6 lock

```
int ssat_t::lock[NFREQ]
```

Definition at line [1042](#) of file [rtklib.h](#).

#### 10.381.2.7 mw

```
double ssat_t::mw
```

Definition at line [1048](#) of file [rtklib.h](#).

#### 10.381.2.8 outc

```
unsigned int ssat_t::outc[NFREQ]
```

Definition at line [1043](#) of file [rtklib.h](#).

#### 10.381.2.9 ph

```
double ssat_t::ph[2][NFREQ]
```

Definition at line [1051](#) of file [rtklib.h](#).

#### 10.381.2.10 phw

```
double ssat_t::phw
```

Definition at line [1049](#) of file [rtklib.h](#).

**10.381.2.11 pt**

```
gtime_t ssat_t::pt[2][NFREQ]
```

Definition at line 1050 of file [rtklib.h](#).

**10.381.2.12 rejc**

```
unsigned int ssat_t::rejc[NFREQ]
```

Definition at line 1045 of file [rtklib.h](#).

**10.381.2.13 resc**

```
double ssat_t::resc[NFREQ]
```

Definition at line 1036 of file [rtklib.h](#).

**10.381.2.14 resp**

```
double ssat_t::resp[NFREQ]
```

Definition at line 1035 of file [rtklib.h](#).

**10.381.2.15 slip**

```
unsigned char ssat_t::slip[NFREQ]
```

Definition at line 1040 of file [rtklib.h](#).

**10.381.2.16 slipc**

```
unsigned int ssat_t::slipc[NFREQ]
```

Definition at line 1044 of file [rtklib.h](#).

**10.381.2.17 snr**

```
unsigned char ssat_t::snr[NFREQ]
```

Definition at line 1038 of file [rtklib.h](#).

**10.381.2.18 sys**

```
unsigned char ssat_t::sys
```

Definition at line 1032 of file [rtklib.h](#).

**10.381.2.19 vs**

```
unsigned char ssat_t::vs
```

Definition at line 1033 of file [rtklib.h](#).

**10.381.2.20 vsat**

```
unsigned char ssat_t::vsat[NFREQ]
```

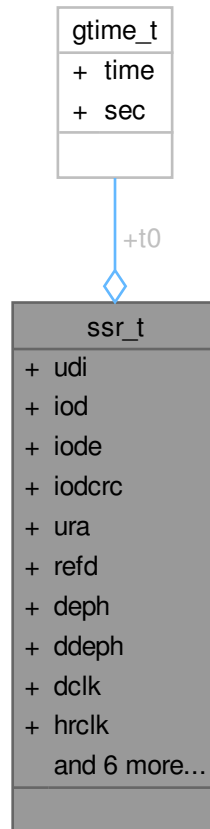
Definition at line 1037 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.382 `ssr_t` Struct Reference

Collaboration diagram for `ssr_t`:



### Public Attributes

- `gtime_t t0` [6]
- double `udi` [6]
- int `iod` [6]
- int `iode`
- int `iodcrc`
- int `ura`
- int `refd`
- double `deph` [3]
- double `ddeph` [3]
- double `dclk` [3]
- double `hrclk`
- float `cbias` [MAXCODE]
- double `pbias` [MAXCODE]
- float `stdpb` [MAXCODE]
- double `yaw_ang`
- double `yaw_rate`
- unsigned char `update`

### 10.382.1 Detailed Description

Definition at line 655 of file [rtklib.h](#).

### 10.382.2 Member Data Documentation

#### 10.382.2.1 cbias

```
float ssr_t::cbias[MAXCODE]
```

Definition at line 668 of file [rtklib.h](#).

#### 10.382.2.2 dclk

```
double ssr_t::dclk[3]
```

Definition at line 666 of file [rtklib.h](#).

#### 10.382.2.3 ddeph

```
double ssr_t::ddeph[3]
```

Definition at line 665 of file [rtklib.h](#).

#### 10.382.2.4 deph

```
double ssr_t::deph[3]
```

Definition at line 664 of file [rtklib.h](#).

#### 10.382.2.5 hrclk

```
double ssr_t::hrclk
```

Definition at line 667 of file [rtklib.h](#).

#### 10.382.2.6 iod

```
int ssr_t::iod[6]
```

Definition at line 659 of file [rtklib.h](#).

#### 10.382.2.7 iodcrc

```
int ssr_t::iodcrc
```

Definition at line 661 of file [rtklib.h](#).

#### 10.382.2.8 iode

```
int ssr_t::iode
```

Definition at line 660 of file [rtklib.h](#).

#### 10.382.2.9 pbias

```
double ssr_t::pbias[MAXCODE]
```

Definition at line 669 of file [rtklib.h](#).

#### 10.382.2.10 refd

```
int ssr_t::refd
```

Definition at line 663 of file [rtklib.h](#).

#### 10.382.2.11 stdpb

```
float ssr_t::stdpb[MAXCODE]
```

Definition at line 670 of file [rtklib.h](#).

**10.382.2.12 t0**

```
gtime_t ssr_t::t0[6]
```

Definition at line 657 of file [rtklib.h](#).

**10.382.2.13 udi**

```
double ssr_t::udi[6]
```

Definition at line 658 of file [rtklib.h](#).

**10.382.2.14 update**

```
unsigned char ssr_t::update
```

Definition at line 672 of file [rtklib.h](#).

**10.382.2.15 ura**

```
int ssr_t::ura
```

Definition at line 662 of file [rtklib.h](#).

**10.382.2.16 yaw\_ang**

```
double ssr_t::yaw_ang
```

Definition at line 671 of file [rtklib.h](#).

**10.382.2.17 yaw\_rate**

```
double ssr_t::yaw_rate
```

Definition at line 671 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.383 sta\_t Struct Reference

Collaboration diagram for sta\_t:

sta_t
+ name
+ marker
+ antdes
+ antsno
+ rectype
+ recver
+ recsno
+ antsetup
+ itrfr
+ deltype
+ pos
+ del
+ hgt

### Public Attributes

- char [name](#) [[MAXANT](#)]
- char [marker](#) [[MAXANT](#)]
- char [antdes](#) [[MAXANT](#)]
- char [antsno](#) [[MAXANT](#)]
- char [rectype](#) [[MAXANT](#)]
- char [recver](#) [[MAXANT](#)]
- char [recsno](#) [[MAXANT](#)]
- int [antsetup](#)
- int [itrfr](#)
- int [deltype](#)
- double [pos](#) [3]
- double [del](#) [3]
- double [hgt](#)

### 10.383.1 Detailed Description

Definition at line [801](#) of file [rtklib.h](#).

### 10.383.2 Member Data Documentation

#### 10.383.2.1 antdes

char [sta\\_t::antdes](#) [[MAXANT](#)]

Definition at line [805](#) of file [rtklib.h](#).

### 10.383.2.2 antsetup

```
int sta_t::antsetup
```

Definition at line 810 of file [rtklib.h](#).

### 10.383.2.3 antsno

```
char sta_t::antsno[MAXANT]
```

Definition at line 806 of file [rtklib.h](#).

### 10.383.2.4 del

```
double sta_t::del[3]
```

Definition at line 814 of file [rtklib.h](#).

### 10.383.2.5 deltype

```
int sta_t::deltype
```

Definition at line 812 of file [rtklib.h](#).

### 10.383.2.6 hgt

```
double sta_t::hgt
```

Definition at line 815 of file [rtklib.h](#).

### 10.383.2.7 itrfr

```
int sta_t::itrfr
```

Definition at line 811 of file [rtklib.h](#).

### 10.383.2.8 marker

```
char sta_t::marker[MAXANT]
```

Definition at line 804 of file [rtklib.h](#).

### 10.383.2.9 name

```
char sta_t::name[MAXANT]
```

Definition at line 803 of file [rtklib.h](#).

### 10.383.2.10 pos

```
double sta_t::pos[3]
```

Definition at line 813 of file [rtklib.h](#).

### 10.383.2.11 recsno

```
char sta_t::recsno[MAXANT]
```

Definition at line 809 of file [rtklib.h](#).

### 10.383.2.12 rectype

```
char sta_t::rectype[MAXANT]
```

Definition at line 807 of file [rtklib.h](#).

### 10.383.2.13 recver

```
char sta_t::recver[MAXANT]
```

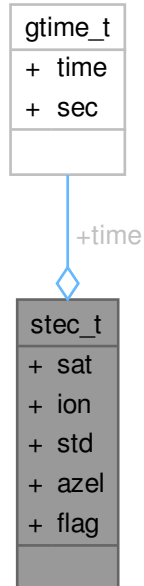
Definition at line 808 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.384 stec\_t Struct Reference

Collaboration diagram for stec\_t:



### Public Attributes

- [gtime\\_t time](#)
- unsigned char [sat](#)
- double [ion](#)
- float [std](#)
- float [azel](#) [2]
- unsigned char [flag](#)

### 10.384.1 Detailed Description

Definition at line 721 of file [rtklib.h](#).

### 10.384.2 Member Data Documentation

#### 10.384.2.1 azel

```
float stec_t::azel[2]
```

Definition at line 727 of file [rtklib.h](#).

#### 10.384.2.2 flag

```
unsigned char stec_t::flag
```

Definition at line 728 of file [rtklib.h](#).

#### 10.384.2.3 ion

```
double stec_t::ion
```

Definition at line 725 of file [rtklib.h](#).



**10.384.2.4 sat**

unsigned char stec\_t::sat

Definition at line 724 of file [rtklib.h](#).

**10.384.2.5 std**

float stec\_t::std

Definition at line 726 of file [rtklib.h](#).

**10.384.2.6 time**

gtime\_t stec\_t::time

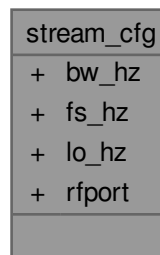
Definition at line 723 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

**10.385 stream\_cfg Struct Reference**

Collaboration diagram for stream\_cfg:

**Public Attributes**

- int64\_t [bw\\_hz](#)
- int64\_t [fs\\_hz](#)
- int64\_t [lo\\_hz](#)
- const char \* [rfport](#)

**10.385.1 Detailed Description**

Definition at line 50 of file [ad9361\\_manager.h](#).

**10.385.2 Member Data Documentation****10.385.2.1 bw\_hz**

int64\_t stream\_cfg::bw\_hz

Definition at line 52 of file [ad9361\\_manager.h](#).

**10.385.2.2 fs\_hz**

int64\_t stream\_cfg::fs\_hz

Definition at line 53 of file [ad9361\\_manager.h](#).

### 10.385.2.3 lo\_hz

```
int64_t stream_cfg::lo_hz
```

Definition at line 54 of file [ad9361\\_manager.h](#).

### 10.385.2.4 rfport

```
const char* stream_cfg::rfport
```

Definition at line 55 of file [ad9361\\_manager.h](#).

The documentation for this struct was generated from the following file:

- [ad9361\\_manager.h](#)

## 10.386 stream\_t Struct Reference

Collaboration diagram for stream\_t:

stream_t
+ type
+ mode
+ state
+ inb
+ inr
+ outb
+ outr
+ tick
+ tact
+ inbt
+ outbt
+ lock
+ port
+ path
+ msg

### Public Attributes

- int [type](#)
- int [mode](#)
- int [state](#)
- unsigned int [inb](#)
- unsigned int [inr](#)
- unsigned int [outb](#)
- unsigned int [outr](#)
- unsigned int [tick](#)
- unsigned int [tact](#)

- unsigned int [inbt](#)
- unsigned int [outbt](#)
- lock\_t [lock](#)
- void \* [port](#)
- char [path](#) [[MAXSTRPATH](#)]
- char [msg](#) [[MAXSTRMSG](#)]

### 10.386.1 Detailed Description

Definition at line [1094](#) of file [rtklib.h](#).

### 10.386.2 Member Data Documentation

#### 10.386.2.1 inb

`unsigned int stream_t::inb`

Definition at line [1099](#) of file [rtklib.h](#).

#### 10.386.2.2 inbt

`unsigned int stream_t::inbt`

Definition at line [1102](#) of file [rtklib.h](#).

#### 10.386.2.3 inr

`unsigned int stream_t::inr`

Definition at line [1099](#) of file [rtklib.h](#).

#### 10.386.2.4 lock

`lock_t stream_t::lock`

Definition at line [1103](#) of file [rtklib.h](#).

#### 10.386.2.5 mode

`int stream_t::mode`

Definition at line [1097](#) of file [rtklib.h](#).

#### 10.386.2.6 msg

`char stream_t::msg[MAXSTRMSG]`

Definition at line [1106](#) of file [rtklib.h](#).

#### 10.386.2.7 outb

`unsigned int stream_t::outb`

Definition at line [1100](#) of file [rtklib.h](#).

#### 10.386.2.8 outbt

`unsigned int stream_t::outbt`

Definition at line [1102](#) of file [rtklib.h](#).

#### 10.386.2.9 outr

`unsigned int stream_t::outr`

Definition at line [1100](#) of file [rtklib.h](#).

**10.386.2.10 path**

```
char stream_t::path[MAXSTRPATH]
```

Definition at line 1105 of file [rtklib.h](#).

**10.386.2.11 port**

```
void* stream_t::port
```

Definition at line 1104 of file [rtklib.h](#).

**10.386.2.12 state**

```
int stream_t::state
```

Definition at line 1098 of file [rtklib.h](#).

**10.386.2.13 tact**

```
unsigned int stream_t::tact
```

Definition at line 1101 of file [rtklib.h](#).

**10.386.2.14 tick**

```
unsigned int stream_t::tick
```

Definition at line 1101 of file [rtklib.h](#).

**10.386.2.15 type**

```
int stream_t::type
```

Definition at line 1096 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

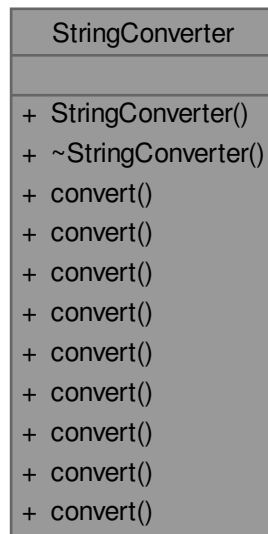
- [rtklib.h](#)

## 10.387 StringConverter Class Reference

Class that interprets the contents of a string and converts it into different types.

```
#include <string_converter.h>
```

Collaboration diagram for StringConverter:



### Public Member Functions

- `bool convert (const std::string &value, bool default_value)`
- `int64_t convert (const std::string &value, int64_t default_value)`
- `uint64_t convert (const std::string &value, uint64_t default_value)`
- `int32_t convert (const std::string &value, int32_t default_value)`
- `uint32_t convert (const std::string &value, uint32_t default_value)`
- `int16_t convert (const std::string &value, int16_t default_value)`
- `uint16_t convert (const std::string &value, uint16_t default_value)`
- `float convert (const std::string &value, float default_value)`
- `double convert (const std::string &value, double default_value)`

#### 10.387.1 Detailed Description

Class that interprets the contents of a string and converts it into different types.

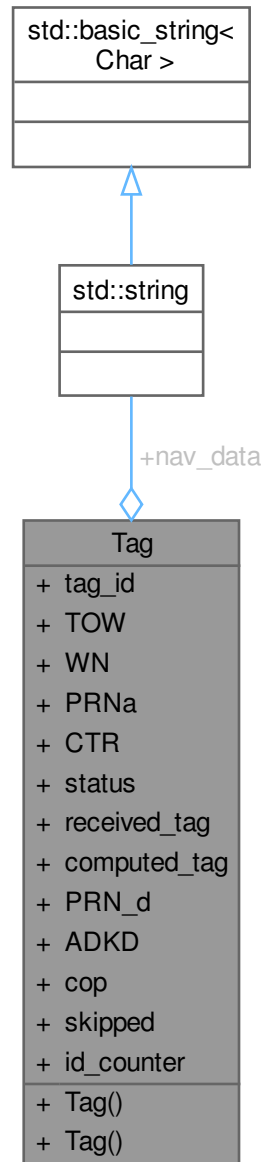
Definition at line 35 of file [string\\_converter.h](#).

The documentation for this class was generated from the following file:

- [string\\_converter.h](#)

## 10.388 Tag Class Reference

Collaboration diagram for Tag:



### Public Types

- enum `e_verification_status` { `SUCCESS`, `FAIL`, `UNVERIFIED` }

### Public Member Functions

- `Tag` (const `MACK_tag_and_info` &MTI, uint32\_t TOW, uint32\_t WN, uint32\_t PRNa, uint8\_t CTR)
- `Tag` (const `MACK_message` &mack)

**Public Attributes**

- const uint32\_t [tag\\_id](#)
- uint32\_t [TOW](#)
- uint32\_t [WN](#)
- uint32\_t [PRNa](#)
- uint8\_t [CTR](#)
- e\_verification\_status [status](#)
- uint64\_t [received\\_tag](#)
- uint64\_t [computed\\_tag](#)
- uint8\_t [PRN\\_d](#)
- uint8\_t [ADKD](#)
- uint8\_t [cop](#)
- uint32\_t [skipped](#)
- std::string [nav\\_data](#)

**Static Public Attributes**

- static uint32\_t [id\\_counter](#)

**10.388.1 Detailed Description**

Definition at line 192 of file [osnma\\_data.h](#).

**10.388.2 Member Enumeration Documentation****10.388.2.1 e\_verification\_status**

```
enum Tag::e_verification_status
```

Definition at line 195 of file [osnma\\_data.h](#).

**10.388.3 Constructor & Destructor Documentation****10.388.3.1 Tag() [1/2]**

```
Tag::Tag (
    const MACK_tag_and_info & MTI,
    uint32_t TOW,
    uint32_t WN,
    uint32_t PRNa,
    uint8_t CTR) [inline]
```

Definition at line 201 of file [osnma\\_data.h](#).

**10.388.3.2 Tag() [2/2]**

```
Tag::Tag (
    const MACK_message & mack) [inline], [explicit]
```

Definition at line 216 of file [osnma\\_data.h](#).

**10.388.4 Member Data Documentation****10.388.4.1 ADKD**

```
uint8_t Tag::ADKD
```

Definition at line 241 of file [osnma\\_data.h](#).

**10.388.4.2 computed\_tag**

```
uint64_t Tag::computed_tag
```

Definition at line 239 of file [osnma\\_data.h](#).

#### 10.388.4.3 cop

uint8\_t Tag::cop

Definition at line 242 of file [osnma\\_data.h](#).

#### 10.388.4.4 CTR

uint8\_t Tag::CTR

Definition at line 236 of file [osnma\\_data.h](#).

#### 10.388.4.5 id\_counter

uint32\_t Tag::id\_counter [static]

Definition at line 232 of file [osnma\\_data.h](#).

#### 10.388.4.6 nav\_data

std::string Tag::nav\_data

Definition at line 244 of file [osnma\\_data.h](#).

#### 10.388.4.7 PRN\_d

uint8\_t Tag::PRN\_d

Definition at line 240 of file [osnma\\_data.h](#).

#### 10.388.4.8 PRNa

uint32\_t Tag::PRNa

Definition at line 235 of file [osnma\\_data.h](#).

#### 10.388.4.9 received\_tag

uint64\_t Tag::received\_tag

Definition at line 238 of file [osnma\\_data.h](#).

#### 10.388.4.10 skipped

uint32\_t Tag::skipped

Definition at line 243 of file [osnma\\_data.h](#).

#### 10.388.4.11 status

e\_verification\_status Tag::status

Definition at line 237 of file [osnma\\_data.h](#).

#### 10.388.4.12 tag\_id

const uint32\_t Tag::tag\_id

Definition at line 231 of file [osnma\\_data.h](#).

#### 10.388.4.13 TOW

uint32\_t Tag::TOW

Definition at line 233 of file [osnma\\_data.h](#).

#### 10.388.4.14 WN

uint32\_t Tag::WN

Definition at line 234 of file [osnma\\_data.h](#).

The documentation for this class was generated from the following file:

- [osnma\\_data.h](#)



## 10.389 Tcp\_Communication Class Reference

TCP communication class.

```
#include <tcp_communication.h>
```

Collaboration diagram for Tcp\_Communication:

Tcp_Communication
<ul style="list-style-type: none"> <li>+ Tcp_Communication()</li> <li>+ ~Tcp_Communication()</li> <li>+ listen_tcp_connection()</li> <li>+ send_receive_tcp_packet_galileo_e1()</li> <li>+ send_receive_tcp_packet_gps_l1_ca()</li> <li>+ close_tcp_connection()</li> </ul>

### Public Member Functions

- int **listen\_tcp\_connection** (size\_t d\_port\_, size\_t d\_port\_ch0\_)
- void **send\_receive\_tcp\_packet\_galileo\_e1** (boost::array< float, NUM\_TX\_VARIABLES\_GALILEO\_E1 > buf, [Tcp\\_Packet\\_Data](#) \*tcp\_data\_)
- void **send\_receive\_tcp\_packet\_gps\_l1\_ca** (boost::array< float, NUM\_TX\_VARIABLES\_GPS\_L1\_CA > buf, [Tcp\\_Packet\\_Data](#) \*tcp\_data\_)
- void **close\_tcp\_connection** (size\_t d\_port\_)

### 10.389.1 Detailed Description

TCP communication class.

Definition at line 44 of file [tcp\\_communication.h](#).

The documentation for this class was generated from the following file:

- [tcp\\_communication.h](#)

## 10.390 Tcp\_Packet\_Data Class Reference

Class that implements a TCP data packet.

```
#include <tcp_packet_data.h>
```

Collaboration diagram for Tcp\_Packet\_Data:

Tcp_Packet_Data
+ proc_pack_code_error
+ proc_pack_carr_error
+ proc_pack_carrier_doppler_hz
+ Tcp_Packet_Data()
+ ~Tcp_Packet_Data()

### Public Attributes

- float [proc\\_pack\\_code\\_error](#) = 0.0
- float [proc\\_pack\\_carr\\_error](#) = 0.0
- float [proc\\_pack\\_carrier\\_doppler\\_hz](#) = 0.0

### 10.390.1 Detailed Description

Class that implements a TCP data packet.

Definition at line 30 of file [tcp\\_packet\\_data.h](#).

### 10.390.2 Member Data Documentation

#### 10.390.2.1 proc\_pack\_carr\_error

```
float Tcp_Packet_Data::proc_pack_carr_error = 0.0
```

Definition at line 36 of file [tcp\\_packet\\_data.h](#).

#### 10.390.2.2 proc\_pack\_carrier\_doppler\_hz

```
float Tcp_Packet_Data::proc_pack_carrier_doppler_hz = 0.0
```

Definition at line 37 of file [tcp\\_packet\\_data.h](#).

#### 10.390.2.3 proc\_pack\_code\_error

```
float Tcp_Packet_Data::proc_pack_code_error = 0.0
```

Definition at line 35 of file [tcp\\_packet\\_data.h](#).

The documentation for this class was generated from the following file:

- [tcp\\_packet\\_data.h](#)

## 10.391 tcp\_t Struct Reference

Collaboration diagram for tcp\_t:

tcp_t
+ state
+ saddr
+ port
+ addr
+ sock
+ tcon
+ tact
+ tdis

### Public Attributes

- int [state](#)
- char [saddr](#) [256]
- int [port](#)
- struct sockaddr\_in [addr](#)
- socket\_t [sock](#)
- int [tcon](#)
- unsigned int [tact](#)
- unsigned int [tdis](#)

### 10.391.1 Detailed Description

Definition at line [1141](#) of file [rtklib.h](#).

### 10.391.2 Member Data Documentation

#### 10.391.2.1 addr

```
struct sockaddr_in tcp_t::addr
```

Definition at line [1146](#) of file [rtklib.h](#).

#### 10.391.2.2 port

```
int tcp_t::port
```

Definition at line [1145](#) of file [rtklib.h](#).

#### 10.391.2.3 saddr

```
char tcp_t::saddr[256]
```

Definition at line [1144](#) of file [rtklib.h](#).

**10.391.2.4 sock**

```
socket_t tcp_t::sock
```

Definition at line 1147 of file [rtklib.h](#).

**10.391.2.5 state**

```
int tcp_t::state
```

Definition at line 1143 of file [rtklib.h](#).

**10.391.2.6 tact**

```
unsigned int tcp_t::tact
```

Definition at line 1149 of file [rtklib.h](#).

**10.391.2.7 tcon**

```
int tcp_t::tcon
```

Definition at line 1148 of file [rtklib.h](#).

**10.391.2.8 tdis**

```
unsigned int tcp_t::tdis
```

Definition at line 1150 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.392 tcpcli\_t Struct Reference

Collaboration diagram for tcpcli\_t:



### Public Attributes

- [tcp\\_t](#) [svr](#)
- int [toinact](#)
- int [tirecon](#)

### 10.392.1 Detailed Description

Definition at line [1161](#) of file [rtklib.h](#).

### 10.392.2 Member Data Documentation

#### 10.392.2.1 svr

[tcp\\_t](#) [tcpcli\\_t::svr](#)

Definition at line [1163](#) of file [rtklib.h](#).

#### 10.392.2.2 tirecon

int [tcpcli\\_t::tirecon](#)

Definition at line [1165](#) of file [rtklib.h](#).

### 10.392.2.3 toinact

```
int tcpcli_t::toinact
```

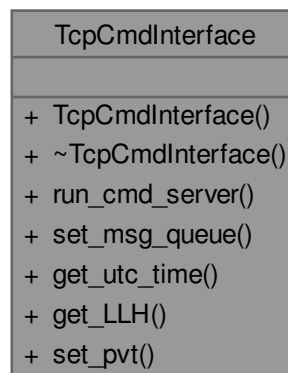
Definition at line 1164 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.393 TcpCmdInterface Class Reference

Collaboration diagram for TcpCmdInterface:



### Public Member Functions

- void **run\_cmd\_server** (int tcp\_port)
- void **set\_msg\_queue** (std::shared\_ptr< [Concurrent\\_Queue](#)< pmt::pmt\_t > > control\_queue)
- time\_t **get\_utc\_time** () const  
*gets the UTC time parsed from the last TC command issued*
- std::array< float, 3 > **get\_LLH** () const  
*gets the Latitude, Longitude and Altitude vector from the last TC command issued*
- void **set\_pvt** (std::shared\_ptr< [PvtInterface](#) > PVT\_sptr)

### 10.393.1 Detailed Description

Definition at line 41 of file [tcp\\_cmd\\_interface.h](#).

### 10.393.2 Member Function Documentation

#### 10.393.2.1 get\_LLH()

```
std::array< float, 3 > TcpCmdInterface::get_LLH () const
```

gets the Latitude, Longitude and Altitude vector from the last TC command issued

#### 10.393.2.2 get\_utc\_time()

```
time_t TcpCmdInterface::get_utc_time () const
```

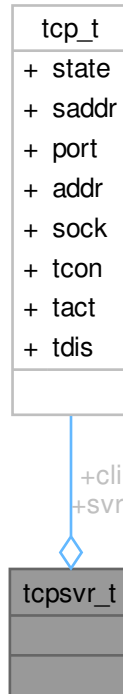
gets the UTC time parsed from the last TC command issued

The documentation for this class was generated from the following file:

- [tcp\\_cmd\\_interface.h](#)

## 10.394 tcpsvr\_t Struct Reference

Collaboration diagram for tcpsvr\_t:



### Public Attributes

- [tcp\\_t svr](#)
- [tcp\\_t cli](#) [[MAXCLI](#)]

### 10.394.1 Detailed Description

Definition at line [1154](#) of file [rtklib.h](#).

### 10.394.2 Member Data Documentation

#### 10.394.2.1 cli

[tcp\\_t](#) tcpsvr\_t::cli [[MAXCLI](#)]

Definition at line [1157](#) of file [rtklib.h](#).

#### 10.394.2.2 svr

[tcp\\_t](#) tcpsvr\_t::svr

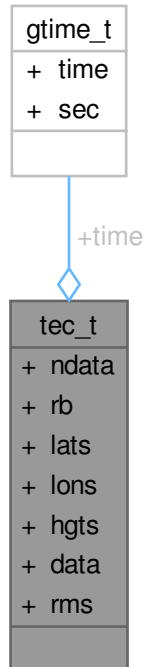
Definition at line [1156](#) of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.395 `tec_t` Struct Reference

Collaboration diagram for `tec_t`:



### Public Attributes

- [gtime\\_t time](#)
- int [ndata](#) [3]
- double [rb](#)
- double [lats](#) [3]
- double [lons](#) [3]
- double [hgts](#) [3]
- double \* [data](#)
- float \* [rms](#)

### 10.395.1 Detailed Description

Definition at line 544 of file [rtklib.h](#).

### 10.395.2 Member Data Documentation

#### 10.395.2.1 `data`

```
double* tec_t::data
```

Definition at line 552 of file [rtklib.h](#).

#### 10.395.2.2 `hgts`

```
double tec_t::hgts[3]
```

Definition at line 551 of file [rtklib.h](#).



double tec\_t::lats[3]  
Definition at line 549 of file rtklib.h.

```
double tec_t::lons[3]
```

Definition at line 550 of file [rtklib.h](#).

```
int tec_t::ndata[3]
```

Definition at line 547 of file [rtklib.h](#).

double tec\_t::rb

Definition at line 548 of file rtklib.h.

float\* tec\_t::rms  
Definition at line 553 of file rtklib.h.

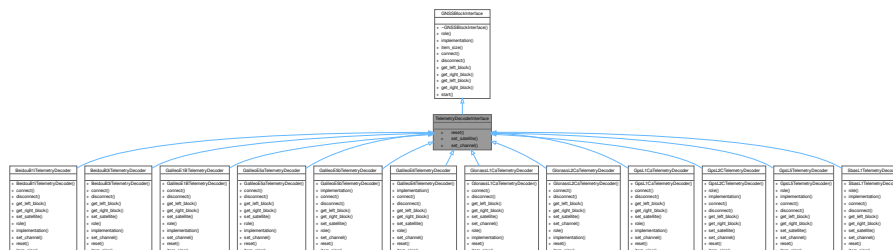
```
gtime_t tec_t::time
```

Definition at line 546 of file [rtklib.h](#).

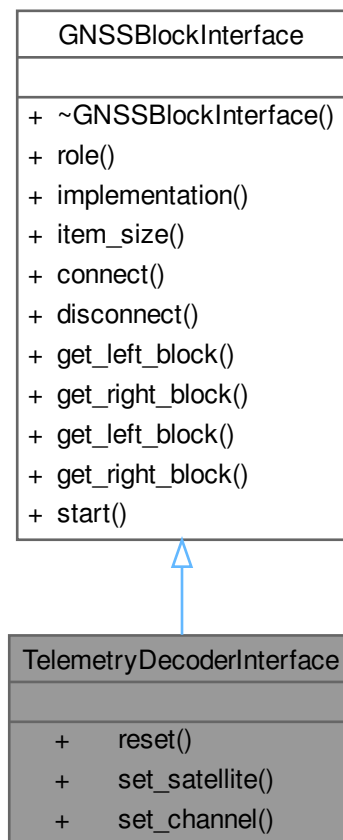
- `rtklib.h`

This abstract class represents an interface to a navigation GNSS block.

Inheritance diagram for TelemetryDecoderInterface:



Collaboration diagram for TelemetryDecoderInterface:



### Public Member Functions

- virtual void **reset** ()=0
- virtual void **set\_satellite** (const [Gnss\\_Satellite](#) &sat)=0
- virtual void **set\_channel** (int channel)=0

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual std::string [role](#) ()=0
- virtual std::string [implementation](#) ()=0
- virtual size\_t [item\\_size](#) ()=0
- virtual void [connect](#) (gr::top\_block\_sptr top\_block)=0
- virtual void [disconnect](#) (gr::top\_block\_sptr top\_block)=0
- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) ()=0
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) ()=0
- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.396.1 Detailed Description

This abstract class represents an interface to a navigation GNSS block.

Abstract class for navigation interfaces. Since all its methods are virtual, this class cannot be instantiated directly, and a subclass can only be instantiated directly if all inherited pure virtual methods have been implemented by that class or a parent class.

Definition at line 43 of file [telemetry\\_decoder\\_interface.h](#).

The documentation for this class was generated from the following file:

- [telemetry\\_decoder\\_interface.h](#)

## 10.397 tle\_t Struct Reference

Collaboration diagram for tle\_t:



**Public Attributes**

- int [n](#)
- int [nmax](#)
- [tled\\_t](#) \* [data](#)

**10.397.1 Detailed Description**

Definition at line [537](#) of file [rtklib.h](#).

**10.397.2 Member Data Documentation****10.397.2.1 data**

```
tled\_t* tle_t::data
```

Definition at line [540](#) of file [rtklib.h](#).

**10.397.2.2 n**

```
int tle_t::n
```

Definition at line [539](#) of file [rtklib.h](#).

**10.397.2.3 nmax**

```
int tle_t::nmax
```

Definition at line [539](#) of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.398 tled\_t Struct Reference

Collaboration diagram for tled\_t:



### Public Attributes

- char [name](#) [32]
- char [alias](#) [32]
- char [satno](#) [16]
- char [satclass](#)
- char [desig](#) [16]
- [gtime\\_t](#) [epoch](#)
- double [ndot](#)
- double [nddot](#)
- double [bstar](#)
- int [etype](#)
- int [eleno](#)
- double [inc](#)
- double [OMG](#)
- double [ecc](#)
- double [omg](#)
- double [M](#)
- double [n](#)
- int [rev](#)

### 10.398.1 Detailed Description

Definition at line 514 of file [rtklib.h](#).

### 10.398.2 Member Data Documentation

#### 10.398.2.1 alias

```
char tled_t::alias[32]
```

Definition at line 517 of file [rtklib.h](#).

#### 10.398.2.2 bstar

```
double tled_t::bstar
```

Definition at line 524 of file [rtklib.h](#).

#### 10.398.2.3 desig

```
char tled_t::desig[16]
```

Definition at line 520 of file [rtklib.h](#).

#### 10.398.2.4 ecc

```
double tled_t::ecc
```

Definition at line 529 of file [rtklib.h](#).

#### 10.398.2.5 eleno

```
int tled_t::eleno
```

Definition at line 526 of file [rtklib.h](#).

#### 10.398.2.6 epoch

```
gtime_t tled_t::epoch
```

Definition at line 521 of file [rtklib.h](#).

#### 10.398.2.7 etype

```
int tled_t::etype
```

Definition at line 525 of file [rtklib.h](#).

#### 10.398.2.8 inc

```
double tled_t::inc
```

Definition at line 527 of file [rtklib.h](#).

#### 10.398.2.9 M

```
double tled_t::M
```

Definition at line 531 of file [rtklib.h](#).

#### 10.398.2.10 n

```
double tled_t::n
```

Definition at line 532 of file [rtklib.h](#).

#### 10.398.2.11 name

```
char tled_t::name[32]
```

Definition at line 516 of file [rtklib.h](#).

**10.398.2.12 nddot**

```
double tled_t::nddot
```

Definition at line 523 of file [rtklib.h](#).

**10.398.2.13 ndot**

```
double tled_t::ndot
```

Definition at line 522 of file [rtklib.h](#).

**10.398.2.14 OMG**

```
double tled_t::OMG
```

Definition at line 528 of file [rtklib.h](#).

**10.398.2.15 omg**

```
double tled_t::omg
```

Definition at line 530 of file [rtklib.h](#).

**10.398.2.16 rev**

```
int tled_t::rev
```

Definition at line 533 of file [rtklib.h](#).

**10.398.2.17 satclass**

```
char tled_t::satclass
```

Definition at line 519 of file [rtklib.h](#).

**10.398.2.18 satno**

```
char tled_t::satno[16]
```

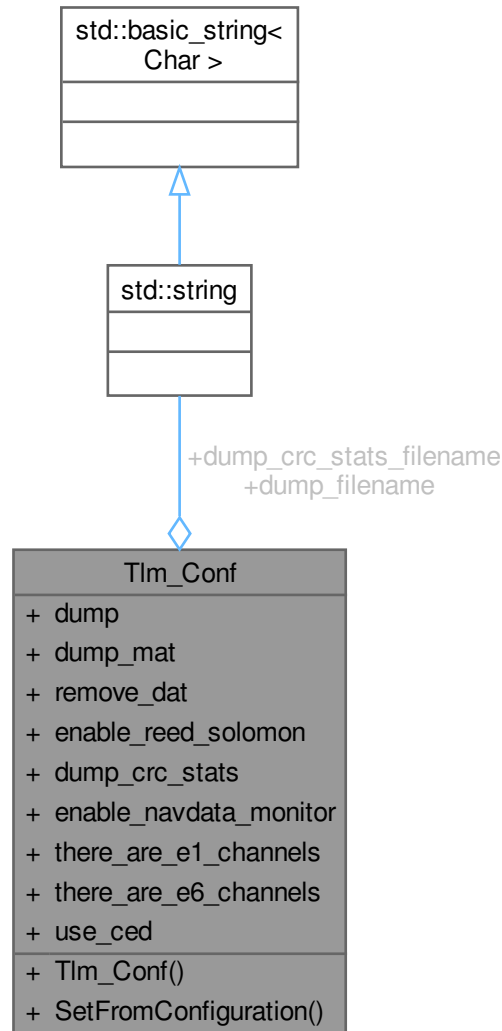
Definition at line 518 of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.399 TIm\_Conf Class Reference

Collaboration diagram for TIm\_Conf:



### Public Member Functions

- void **SetFromConfiguration** (const [ConfigurationInterface](#) \*configuration, const std::string &role)

### Public Attributes

- std::string [dump\\_filename](#)
- std::string [dump\\_crc\\_stats\\_filename](#)
- bool [dump](#) {false}
- bool [dump\\_mat](#) {false}
- bool [remove\\_dat](#) {false}
- bool [enable\\_reed\\_solomon](#) {false}
- bool [dump\\_crc\\_stats](#) {false}



- bool [enable\\_navdata\\_monitor](#) {false}
- bool [there\\_are\\_e1\\_channels](#) {false}
- bool [there\\_are\\_e6\\_channels](#) {false}
- bool [use\\_ced](#) {false}

### 10.399.1 Detailed Description

Definition at line 30 of file [tlm\\_conf.h](#).

### 10.399.2 Member Data Documentation

#### 10.399.2.1 dump

```
bool Tlm_Conf::dump {false}
```

Definition at line 39 of file [tlm\\_conf.h](#).

#### 10.399.2.2 dump\_crc\_stats

```
bool Tlm_Conf::dump_crc_stats {false}
```

Definition at line 43 of file [tlm\\_conf.h](#).

#### 10.399.2.3 dump\_crc\_stats\_filename

```
std::string Tlm_Conf::dump_crc_stats_filename
```

Definition at line 38 of file [tlm\\_conf.h](#).

#### 10.399.2.4 dump\_filename

```
std::string Tlm_Conf::dump_filename
```

Definition at line 37 of file [tlm\\_conf.h](#).

#### 10.399.2.5 dump\_mat

```
bool Tlm_Conf::dump_mat {false}
```

Definition at line 40 of file [tlm\\_conf.h](#).

#### 10.399.2.6 enable\_navdata\_monitor

```
bool Tlm_Conf::enable_navdata_monitor {false}
```

Definition at line 44 of file [tlm\\_conf.h](#).

#### 10.399.2.7 enable\_reed\_solomon

```
bool Tlm_Conf::enable_reed_solomon {false}
```

Definition at line 42 of file [tlm\\_conf.h](#).

#### 10.399.2.8 remove\_dat

```
bool Tlm_Conf::remove_dat {false}
```

Definition at line 41 of file [tlm\\_conf.h](#).

#### 10.399.2.9 there\_are\_e1\_channels

```
bool Tlm_Conf::there_are_e1_channels {false}
```

Definition at line 45 of file [tlm\\_conf.h](#).

#### 10.399.2.10 there\_are\_e6\_channels

```
bool Tlm_Conf::there_are_e6_channels {false}
```

Definition at line 46 of file [tlm\\_conf.h](#).

### 10.399.2.11 use\_ced

```
bool Tlm_Conf::use_ced {false}
```

Definition at line 47 of file [tlm\\_conf.h](#).

The documentation for this class was generated from the following file:

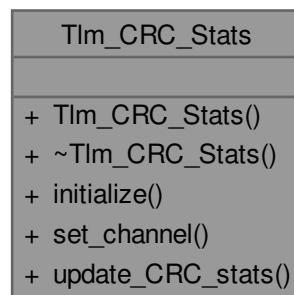
- [tlm\\_conf.h](#)

## 10.400 Tlm\_CRC\_Stats Class Reference

Class that computes the telemetry CRC statistics.

```
#include <tlm_crc_stats.h>
```

Collaboration diagram for Tlm\_CRC\_Stats:



### Public Member Functions

- void [initialize](#) (std::string dump\_crc\_stats\_filename)  
*Initialize the telemetry CRC statistics.*
- bool [set\\_channel](#) (int32\_t channel)  
*Initialize the channel number and output file.*
- void [update\\_CRC\\_stats](#) (bool CRC)  
*Update the CRC statistics.*

### 10.400.1 Detailed Description

Class that computes the telemetry CRC statistics.

Definition at line 32 of file [tlm\\_crc\\_stats.h](#).

### 10.400.2 Member Function Documentation

#### 10.400.2.1 initialize()

```
void Tlm_CRC_Stats::initialize (
    std::string dump_crc_stats_filename)
```

Initialize the telemetry CRC statistics.

#### 10.400.2.2 set\_channel()

```
bool Tlm_CRC_Stats::set_channel (
    int32_t channel)
```

Initialize the channel number and output file.

### 10.400.2.3 update\_CRC\_stats()

```
void Tlm_CRC_Stats::update_CRC_stats (
    bool CRC)
```

Update the CRC statistics.

The documentation for this class was generated from the following file:

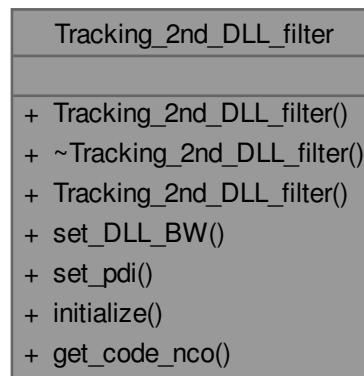
- [tlm\\_crc\\_stats.h](#)

## 10.401 Tracking\_2nd\_DLL\_filter Class Reference

This class implements a 2nd order DLL filter for code tracking loop.

```
#include <tracking_2nd_DLL_filter.h>
```

Collaboration diagram for Tracking\_2nd\_DLL\_filter:



### Public Member Functions

- **Tracking\_2nd\_DLL\_filter** (float pdi\_code)
- void **set\_DLL\_BW** (float dll\_bw\_hz)  
*Set DLL filter bandwidth [Hz].*
- void **set\_pdi** (float pdi\_code)  
*Set Summation interval for code [s].*
- void **initialize** ()  
*Start tracking with acquisition information.*
- float **get\_code\_nco** (float DLL\_discriminator)  
*Numerically controlled oscillator.*

### 10.401.1 Detailed Description

This class implements a 2nd order DLL filter for code tracking loop.

The algorithm is described in: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S. H. Jensen, A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach, Birkhauser, 2007, Applied and Numerical Harmonic Analysis.

Definition at line 40 of file [tracking\\_2nd\\_DLL\\_filter.h](#).

## 10.401.2 Member Function Documentation

### 10.401.2.1 `get_code_nco()`

```
float Tracking_2nd_DLL_filter::get_code_nco (
    float DLL_discriminator)
```

Numerically controlled oscillator.

### 10.401.2.2 `initialize()`

```
void Tracking_2nd_DLL_filter::initialize ()
```

Start tracking with acquisition information.

### 10.401.2.3 `set_DLL_BW()`

```
void Tracking_2nd_DLL_filter::set_DLL_BW (
    float dll_bw_hz)
```

Set DLL filter bandwidth [Hz].

### 10.401.2.4 `set_pdi()`

```
void Tracking_2nd_DLL_filter::set_pdi (
    float pdi_code)
```

Set Summation interval for code [s].

The documentation for this class was generated from the following file:

- [tracking\\_2nd\\_DLL\\_filter.h](#)

## 10.402 Tracking\_2nd\_PLL\_filter Class Reference

This class implements a 2nd order PLL filter for carrier tracking loop.

```
#include <tracking_2nd_PLL_filter.h>
```

Collaboration diagram for Tracking\_2nd\_PLL\_filter:

Tracking_2nd_PLL_filter
<ul style="list-style-type: none"> <li>+ Tracking_2nd_PLL_filter()</li> <li>+ ~Tracking_2nd_PLL_filter()</li> <li>+ Tracking_2nd_PLL_filter()</li> <li>+ set_PLL_BW()</li> <li>+ set_pdi()</li> <li>+ initialize()</li> <li>+ get_carrier_nco()</li> </ul>

### Public Member Functions

- **Tracking\_2nd\_PLL\_filter** (float pdi\_carr)
- void [set\\_PLL\\_BW](#) (float pll\_bw\_hz)  
Set PLL loop bandwidth [Hz].

- void **set\_pdi** (float pdi\_carr)  
*Set Summation interval for code [s].*
- void **initialize** ()
- float **get\_carrier\_nco** (float PLL\_discriminator)

### 10.402.1 Detailed Description

This class implements a 2nd order PLL filter for carrier tracking loop.

The algorithm is described in: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S. H. Jensen, A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach, Birkhauser, 2007, Applied and Numerical Harmonic Analysis.

Definition at line 39 of file [tracking\\_2nd\\_PLL\\_filter.h](#).

### 10.402.2 Member Function Documentation

#### 10.402.2.1 set\_pdi()

```
void Tracking_2nd_PLL_filter::set_pdi (
    float pdi_carr)
```

Set Summation interval for code [s].

#### 10.402.2.2 set\_PLL\_BW()

```
void Tracking_2nd_PLL_filter::set_PLL_BW (
    float pll_bw_hz)
```

Set PLL loop bandwidth [Hz].

The documentation for this class was generated from the following file:

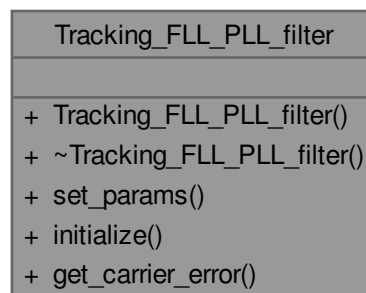
- [tracking\\_2nd\\_PLL\\_filter.h](#)

## 10.403 Tracking\_FLL\_PLL\_filter Class Reference

This class implements a hybrid FLL and PLL filter for tracking carrier loop.

```
#include <tracking_FLL_PLL_filter.h>
```

Collaboration diagram for Tracking\_FLL\_PLL\_filter:



### Public Member Functions

- void **set\_params** (float fll\_bw\_hz, float pll\_bw\_hz, int order)
- void **initialize** (float d\_acq\_carrier\_doppler\_hz)
- float **get\_carrier\_error** (float FLL\_discriminator, float PLL\_discriminator, float correlation\_time\_s)

### 10.403.1 Detailed Description

This class implements a hybrid FLL and PLL filter for tracking carrier loop.

Definition at line 29 of file [tracking\\_FLL\\_PLL\\_filter.h](#).

The documentation for this class was generated from the following file:

- [tracking\\_FLL\\_PLL\\_filter.h](#)

## 10.404 Tracking\_loop\_filter Class Reference

This class implements a generic 1st, 2nd or 3rd order loop filter.

#include <tracking\_loop\_filter.h>

Collaboration diagram for Tracking\_loop\_filter:

Tracking_loop_filter
<ul style="list-style-type: none"> <li>+ Tracking_loop_filter()</li> <li>+ ~Tracking_loop_filter()</li> <li>+ Tracking_loop_filter()</li> <li>+ Tracking_loop_filter()</li> <li>+ operator=()</li> <li>+ get_noise_bandwidth()</li> <li>+ get_update_interval()</li> <li>+ get_include_last_integrator()</li> <li>+ get_order()</li> <li>+ set_noise_bandwidth()</li> <li>+ set_update_interval()</li> <li>+ set_include_last_integrator()</li> <li>+ set_order()</li> <li>+ initialize()</li> <li>+ apply()</li> </ul>

### Public Member Functions

- **Tracking\_loop\_filter** (float update\_interval, float noise\_bandwidth, int loop\_order=2, bool include\_last\_integrator=false)
- **Tracking\_loop\_filter** ([Tracking\\_loop\\_filter](#) &&)=default  
*Move operator.*
- **Tracking\_loop\_filter** & **operator=** ([Tracking\\_loop\\_filter](#) &&)=default  
*Move assignment operator.*
- float **get\_noise\_bandwidth** () const
- float **get\_update\_interval** () const
- bool **get\_include\_last\_integrator** () const
- int **get\_order** () const
- void **set\_noise\_bandwidth** (float noise\_bandwidth)

- void **set\_update\_interval** (float update\_interval)
- void **set\_include\_last\_integrator** (bool include\_last\_integrator)
- void **set\_order** (int loop\_order)
- void **initialize** (float initial\_output=0.0)
- float **apply** (float current\_input)

### 10.404.1 Detailed Description

This class implements a generic 1st, 2nd or 3rd order loop filter.  
Definition at line 35 of file [tracking\\_loop\\_filter.h](#).

### 10.404.2 Constructor & Destructor Documentation

#### 10.404.2.1 Tracking\_loop\_filter()

```
Tracking_loop_filter::Tracking_loop_filter (
    Tracking_loop_filter && ) [default]
```

Move operator.

### 10.404.3 Member Function Documentation

#### 10.404.3.1 operator=()

```
Tracking_loop_filter & Tracking_loop_filter::operator= (
    Tracking_loop_filter && ) [default]
```

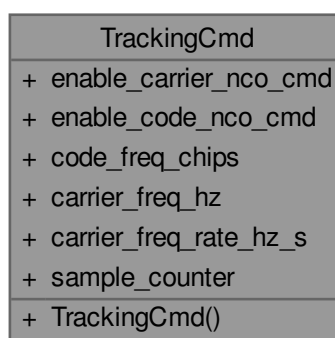
Move assignment operator.

The documentation for this class was generated from the following file:

- [tracking\\_loop\\_filter.h](#)

## 10.405 TrackingCmd Class Reference

Collaboration diagram for TrackingCmd:



### Public Attributes

- bool [enable\\_carrier\\_nco\\_cmd](#) = false
- bool [enable\\_code\\_nco\\_cmd](#) = false
- double [code\\_freq\\_chips](#) = 0.0

- double `carrier_freq_hz` = 0.0
- double `carrier_freq_rate_hz_s` = 0.0
- uint64\_t `sample_counter` = 0UL

### 10.405.1 Detailed Description

Definition at line 28 of file [trackingcmd.h](#).

### 10.405.2 Member Data Documentation

#### 10.405.2.1 `carrier_freq_hz`

double `TrackingCmd::carrier_freq_hz` = 0.0

Definition at line 36 of file [trackingcmd.h](#).

#### 10.405.2.2 `carrier_freq_rate_hz_s`

double `TrackingCmd::carrier_freq_rate_hz_s` = 0.0

Definition at line 37 of file [trackingcmd.h](#).

#### 10.405.2.3 `code_freq_chips`

double `TrackingCmd::code_freq_chips` = 0.0

Definition at line 35 of file [trackingcmd.h](#).

#### 10.405.2.4 `enable_carrier_nco_cmd`

bool `TrackingCmd::enable_carrier_nco_cmd` = false

Definition at line 33 of file [trackingcmd.h](#).

#### 10.405.2.5 `enable_code_nco_cmd`

bool `TrackingCmd::enable_code_nco_cmd` = false

Definition at line 34 of file [trackingcmd.h](#).

#### 10.405.2.6 `sample_counter`

uint64\_t `TrackingCmd::sample_counter` = 0UL

Definition at line 38 of file [trackingcmd.h](#).

The documentation for this class was generated from the following file:

- [trackingcmd.h](#)

## 10.406 TrackingInterface Class Reference

This abstract class represents an interface to a tracking block.

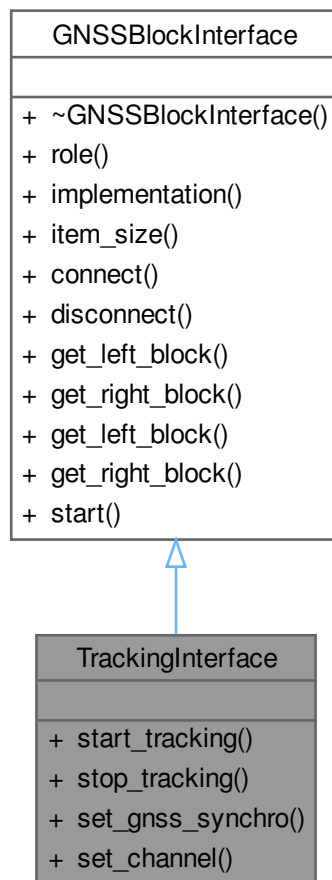
```
#include <tracking_interface.h>
```

Inheritance diagram for `TrackingInterface`:





Collaboration diagram for TrackingInterface:



### Public Member Functions

- virtual void [start\\_tracking](#) ()=0
- virtual void [stop\\_tracking](#) ()=0
- virtual void [set\\_gnss\\_synchro](#) (Gnss\_Synchro \*gnss\_synchro)=0
- virtual void [set\\_channel](#) (unsigned int channel)=0

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual std::string [role](#) ()=0
- virtual std::string [implementation](#) ()=0
- virtual size\_t [item\\_size](#) ()=0
- virtual void [connect](#) (gr::top\_block\_sptr top\_block)=0
- virtual void [disconnect](#) (gr::top\_block\_sptr top\_block)=0
- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) ()=0
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) ()=0
- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### 10.406.1 Detailed Description

This abstract class represents an interface to a tracking block.

Abstract class for tracking interfaces. Since all its methods are virtual, this class cannot be instantiated directly, and a subclass can only be instantiated directly if all inherited pure virtual methods have been implemented by that class or a parent class.

Definition at line 47 of file [tracking\\_interface.h](#).

### 10.406.2 Member Function Documentation

#### 10.406.2.1 set\_channel()

```
virtual void TrackingInterface::set_channel (
    unsigned int channel) [pure virtual]
```

Implemented in [BeidouB1iDIIPIITracking](#), [BeidouB3iDIIPIITracking](#), [GalileoE1DIIPIIVemITracking](#), [GalileoE1DIIPIIVemITrackingFpga](#), [GalileoE1TcpConnectorTracking](#), [GalileoE5aDIIPIITracking](#), [GalileoE5aDIIPIITrackingFpga](#), [GalileoE5bDIIPIITracking](#), [GalileoE6DIIPIITracking](#), [GlonassL1CaDIIPIICAidTracking](#), [GlonassL1CaDIIPIITracking](#), [GlonassL2CaDIIPIICAidTracking](#), [GlonassL2CaDIIPIITracking](#), [GpsL1CaDIIPIITracking](#), [GpsL1CaDIIPIITrackingFpga](#), [GpsL1CaDIIPIITrackingGPU](#), [GpsL1CaGaussianTracking](#), [GpsL1CaKfTracking](#), [GpsL1CaTcpConnectorTracking](#), [GpsL2MDIIPIITracking](#), [GpsL2MDIIPIITrackingFpga](#), [GpsL5DIIPIITracking](#), and [GpsL5DIIPIITrackingFpga](#).

#### 10.406.2.2 set\_gnss\_synchro()

```
virtual void TrackingInterface::set_gnss_synchro (
    Gnss_Synchro * gnss_synchro) [pure virtual]
```

Implemented in [BeidouB1iDIIPIITracking](#), [BeidouB3iDIIPIITracking](#), [GalileoE1DIIPIIVemITracking](#), [GalileoE1DIIPIIVemITrackingFpga](#), [GalileoE1TcpConnectorTracking](#), [GalileoE5aDIIPIITracking](#), [GalileoE5aDIIPIITrackingFpga](#), [GalileoE5bDIIPIITracking](#), [GalileoE6DIIPIITracking](#), [GlonassL1CaDIIPIICAidTracking](#), [GlonassL1CaDIIPIITracking](#), [GlonassL2CaDIIPIICAidTracking](#), [GlonassL2CaDIIPIITracking](#), [GpsL1CaDIIPIITracking](#), [GpsL1CaDIIPIITrackingFpga](#), [GpsL1CaDIIPIITrackingGPU](#), [GpsL1CaGaussianTracking](#), [GpsL1CaKfTracking](#), [GpsL1CaTcpConnectorTracking](#), [GpsL2MDIIPIITracking](#), [GpsL2MDIIPIITrackingFpga](#), [GpsL5DIIPIITracking](#), and [GpsL5DIIPIITrackingFpga](#).

#### 10.406.2.3 start\_tracking()

```
virtual void TrackingInterface::start_tracking () [pure virtual]
```

Implemented in [GalileoE1DIIPIIVemITrackingFpga](#), [GalileoE5aDIIPIITrackingFpga](#), [GpsL1CaDIIPIITrackingFpga](#), and [GpsL5DIIPIITrackingFpga](#).

#### 10.406.2.4 stop\_tracking()

```
virtual void TrackingInterface::stop_tracking () [pure virtual]
```

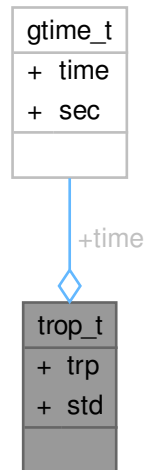
Implemented in [BeidouB1iDIIPIITracking](#), [BeidouB3iDIIPIITracking](#), [GalileoE1DIIPIIVemITracking](#), [GalileoE1DIIPIIVemITrackingFpga](#), [GalileoE1TcpConnectorTracking](#), [GalileoE5aDIIPIITracking](#), [GalileoE5aDIIPIITrackingFpga](#), [GalileoE5bDIIPIITracking](#), [GalileoE6DIIPIITracking](#), [GlonassL1CaDIIPIICAidTracking](#), [GlonassL1CaDIIPIITracking](#), [GlonassL2CaDIIPIICAidTracking](#), [GlonassL2CaDIIPIITracking](#), [GpsL1CaDIIPIITracking](#), [GpsL1CaDIIPIITrackingFpga](#), [GpsL1CaDIIPIITrackingGPU](#), [GpsL1CaGaussianTracking](#), [GpsL1CaKfTracking](#), [GpsL1CaTcpConnectorTracking](#), [GpsL2MDIIPIITracking](#), [GpsL2MDIIPIITrackingFpga](#), [GpsL5DIIPIITracking](#), and [GpsL5DIIPIITrackingFpga](#).

The documentation for this class was generated from the following file:

- [tracking\\_interface.h](#)

## 10.407 trop\_t Struct Reference

Collaboration diagram for trop\_t:



### Public Attributes

- [gtime\\_t time](#)
- double [trp](#) [3]
- float [std](#) [3]

### 10.407.1 Detailed Description

Definition at line [732](#) of file [rtklib.h](#).

### 10.407.2 Member Data Documentation

#### 10.407.2.1 std

```
float trop_t::std[3]
```

Definition at line [736](#) of file [rtklib.h](#).

#### 10.407.2.2 time

```
gtime_t trop_t::time
```

Definition at line [734](#) of file [rtklib.h](#).

#### 10.407.2.3 trp

```
double trop_t::trp[3]
```

Definition at line [735](#) of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

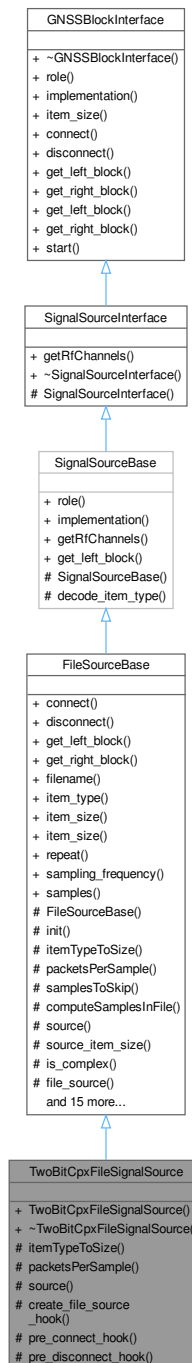
- [rtklib.h](#)

## 10.408 TwoBitCpxFileSignalSource Class Reference

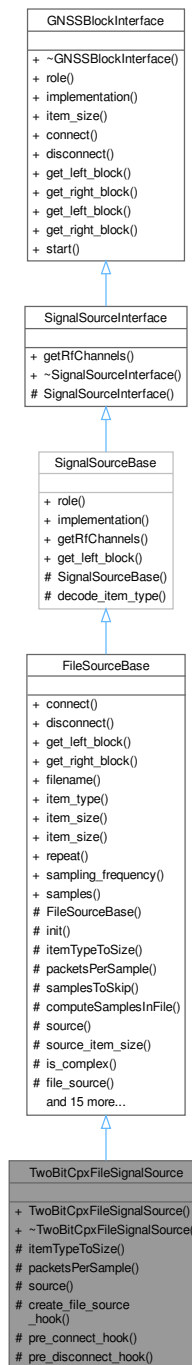
Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).

#include <two\_bit\_cpx\_file\_signal\_source.h>

Inheritance diagram for TwoBitCpxFileSignalSource:



Collaboration diagram for TwoBitCpxFileSignalSource:



## Public Member Functions

- **TwoBitCpxFileSignalSource** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)

## Public Member Functions inherited from [FileSourceBase](#)

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override

- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- std::string [filename](#) () const  
*The file to read.*
- std::string [item\\_type](#) () const  
*The item type.*
- size\_t [item\\_size](#) () override  
*The configured size of each item.*
- virtual size\_t [item\\_size](#) () const
- bool [repeat](#) () const  
*Whether to repeat reading after end-of-file.*
- int64\_t [sampling\\_frequency](#) () const  
*The sampling frequency of the source file.*
- uint64\_t [samples](#) () const  
*The number of samples in the file.*

### Public Member Functions inherited from [SignalSourceBase](#)

- std::string [role](#) () final
- std::string [implementation](#) () final
- size\_t [getRfChannels](#) () const override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()  
*Start the flow of samples if needed.*

### Protected Member Functions

- std::tuple< size\_t, bool > [itemTypeToSize](#) () override  
*Compute the item size, from the [item\\_type](#)(). Subclasses may constrain types that don't make.*
- double [packetsPerSample](#) () const override  
*The number of (possibly unpacked) samples in a (raw) file sample (default=1)*
- gnss\_shared\_ptr< gr::block > [source](#) () const override  
*Abstracted front-end source. Sub-classes may override if they create specialized chains to decode source files into a usable format.*
- void [create\\_file\\_source\\_hook](#) () override
- void [pre\\_connect\\_hook](#) (gr::top\_block\_sptr top\_block) override
- void [pre\\_disconnect\\_hook](#) (gr::top\_block\_sptr top\_block) override

### Protected Member Functions inherited from [FileSourceBase](#)

- [FileSourceBase](#) ([ConfigurationInterface](#) const \*configuration, std::string const &role, std::string impl, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue, std::string default\_item\_type="short")  
*Constructor.*
- void [init](#) ()  
*Perform post-construction initialization.*
- virtual size\_t [samplesToSkip](#) () const  
*Compute the number of samples to skip.*
- size\_t [computeSamplesInFile](#) () const  
*Compute the number of samples in the file.*

- virtual size\_t [source\\_item\\_size](#) () const

*For complex source chains, the size of the file item may not be the same as the size of the.*

- bool **is\_complex** () const
- gnss\_shared\_ptr< gr::block > **file\_source** () const
- gnss\_shared\_ptr< gr::block > **valve** () const
- gnss\_shared\_ptr< gr::block > **throttle** () const
- gnss\_shared\_ptr< gr::block > **sink** () const
- gr::blocks::file\_source::sptr **create\_file\_source** ()
- gr::blocks::throttle::sptr **create\_throttle** ()
- gnss\_shared\_ptr< gr::block > **create\_valve** ()
- gr::blocks::file\_sink::sptr **create\_sink** ()
- virtual void **create\_throttle\_hook** ()
- virtual void **create\_valve\_hook** ()
- virtual void **create\_sink\_hook** ()
- virtual void **post\_connect\_hook** (gr::top\_block\_sptr top\_block)
- virtual void **post\_disconnect\_hook** (gr::top\_block\_sptr top\_block)

## Protected Member Functions inherited from [SignalSourceBase](#)

- [SignalSourceBase](#) ([ConfigurationInterface](#) const \*configuration, std::string role, std::string impl)  
*Constructor.*
- size\_t [decode\\_item\\_type](#) (std::string const &item\_type, bool \*is\_interleaved=nullptr, bool throw\_on\_↵ error=false)  
*utility for decoding passed ".item\_type" values*

### 10.408.1 Detailed Description

Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).  
Definition at line 42 of file [two\\_bit\\_cpx\\_file\\_signal\\_source.h](#).

### 10.408.2 Member Function Documentation

#### 10.408.2.1 [create\\_file\\_source\\_hook\(\)](#)

void TwoBitCpxFileSignalSource::create\_file\_source\_hook () [override], [protected], [virtual]  
Reimplemented from [FileSourceBase](#).

#### 10.408.2.2 [itemTypeToSize\(\)](#)

std::tuple< size\_t, bool > TwoBitCpxFileSignalSource::itemTypeToSize () [override], [protected], [virtual]  
Compute the item size, from the [item\\_type\(\)](#). Subclasses may constrain types that don't make.  
Reimplemented from [FileSourceBase](#).

#### 10.408.2.3 [packetsPerSample\(\)](#)

double TwoBitCpxFileSignalSource::packetsPerSample () const [override], [protected], [virtual]  
The number of (possibly unpacked) samples in a (raw) file sample (default=1)  
Reimplemented from [FileSourceBase](#).

#### 10.408.2.4 [pre\\_connect\\_hook\(\)](#)

void TwoBitCpxFileSignalSource::pre\_connect\_hook (gr::top\_block\_sptr top\_block) [override], [protected], [virtual]  
Reimplemented from [FileSourceBase](#).

#### 10.408.2.5 pre\_disconnect\_hook()

```
void TwoBitCpxFileSignalSource::pre_disconnect_hook (  
    gr::top_block_sptr top_block) [override], [protected], [virtual]
```

Reimplemented from [FileSourceBase](#).

#### 10.408.2.6 source()

```
gnss_shared_ptr< gr::block > TwoBitCpxFileSignalSource::source () const [override], [protected],  
[virtual]
```

Abstracted front-end source. Sub-classes may override if they create specialized chains to decode source files into a usable format.

Reimplemented from [FileSourceBase](#).

The documentation for this class was generated from the following file:

- [two\\_bit\\_cpx\\_file\\_signal\\_source.h](#)

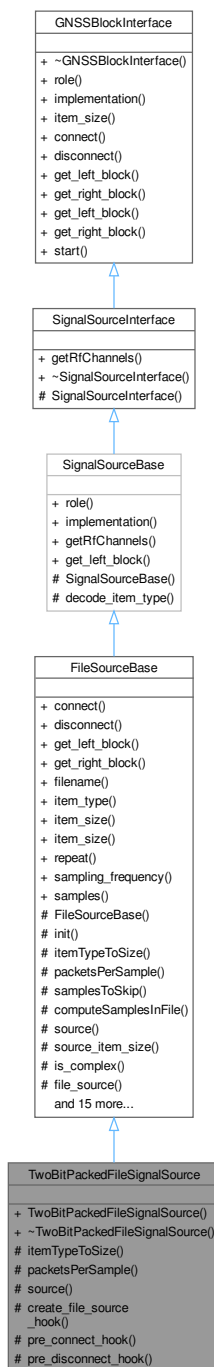
## 10.409 TwoBitPackedFileSignalSource Class Reference

Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).

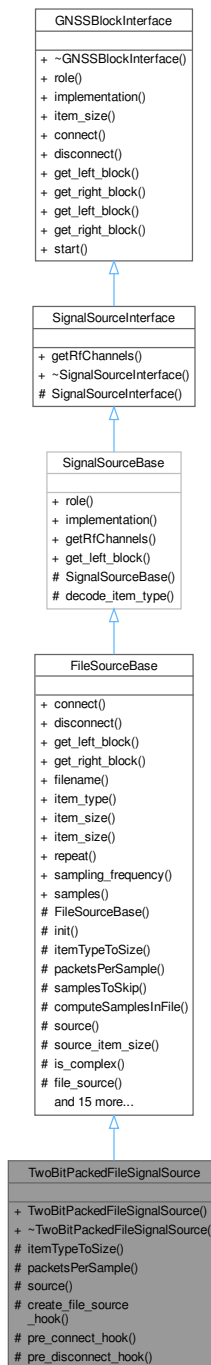
```
#include <two_bit_packed_file_signal_source.h>
```



Inheritance diagram for TwoBitPackedFileSignalSource:



Collaboration diagram for TwoBitPackedFileSignalSource:



## Public Member Functions

- **TwoBitPackedFileSignalSource** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_streams, unsigned int out\_streams, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)

## Public Member Functions inherited from [FileSourceBase](#)

- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override

- gr::basic\_block\_sptr [get\\_left\\_block](#) () override
- gr::basic\_block\_sptr [get\\_right\\_block](#) () override
- std::string [filename](#) () const  
*The file to read.*
- std::string [item\\_type](#) () const  
*The item type.*
- size\_t [item\\_size](#) () override  
*The configured size of each item.*
- virtual size\_t [item\\_size](#) () const
- bool [repeat](#) () const  
*Whether to repeat reading after end-of-file.*
- int64\_t [sampling\\_frequency](#) () const  
*The sampling frequency of the source file.*
- uint64\_t [samples](#) () const  
*The number of samples in the file.*

### Public Member Functions inherited from [SignalSourceBase](#)

- std::string [role](#) () final
- std::string [implementation](#) () final
- size\_t [getRfChannels](#) () const override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual gr::basic\_block\_sptr [get\\_left\\_block](#) (int RF\_channel)
- virtual gr::basic\_block\_sptr [get\\_right\\_block](#) (int RF\_channel)
- virtual void [start](#) ()  
*Start the flow of samples if needed.*

### Protected Member Functions

- std::tuple< size\_t, bool > [itemTypeToSize](#) () override  
*Compute the item size, from the [item\\_type](#)(). Subclasses may constrain types that don't make.*
- double [packetsPerSample](#) () const override  
*The number of (possibly unpacked) samples in a (raw) file sample (default=1)*
- gnss\_shared\_ptr< gr::block > [source](#) () const override  
*Abstracted front-end source. Sub-classes may override if they create specialized chains to decode source files into a usable format.*
- void [create\\_file\\_source\\_hook](#) () override
- void [pre\\_connect\\_hook](#) (gr::top\_block\_sptr top\_block) override
- void [pre\\_disconnect\\_hook](#) (gr::top\_block\_sptr top\_block) override

### Protected Member Functions inherited from [FileSourceBase](#)

- [FileSourceBase](#) ([ConfigurationInterface](#) const \*configuration, std::string const &role, std::string impl, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue, std::string default\_item\_type="short")  
*Constructor.*
- void [init](#) ()  
*Perform post-construction initialization.*
- virtual size\_t [samplesToSkip](#) () const  
*Compute the number of samples to skip.*
- size\_t [computeSamplesInFile](#) () const  
*Compute the number of samples in the file.*

- virtual size\_t [source\\_item\\_size](#) () const

*For complex source chains, the size of the file item may not be the same as the size of the.*

- bool **is\_complex** () const
- gnss\_shared\_ptr< gr::block > **file\_source** () const
- gnss\_shared\_ptr< gr::block > **valve** () const
- gnss\_shared\_ptr< gr::block > **throttle** () const
- gnss\_shared\_ptr< gr::block > **sink** () const
- gr::blocks::file\_source::sptr **create\_file\_source** ()
- gr::blocks::throttle::sptr **create\_throttle** ()
- gnss\_shared\_ptr< gr::block > **create\_valve** ()
- gr::blocks::file\_sink::sptr **create\_sink** ()
- virtual void **create\_throttle\_hook** ()
- virtual void **create\_valve\_hook** ()
- virtual void **create\_sink\_hook** ()
- virtual void **post\_connect\_hook** (gr::top\_block\_sptr top\_block)
- virtual void **post\_disconnect\_hook** (gr::top\_block\_sptr top\_block)

## Protected Member Functions inherited from [SignalSourceBase](#)

- [SignalSourceBase](#) ([ConfigurationInterface](#) const \*configuration, std::string role, std::string impl)  
*Constructor.*
- size\_t [decode\\_item\\_type](#) (std::string const &item\_type, bool \*is\_interleaved=nullptr, bool throw\_on\_↵ error=false)  
*utility for decoding passed ".item\_type" values*

## 10.409.1 Detailed Description

Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).

Definition at line 44 of file [two\\_bit\\_packed\\_file\\_signal\\_source.h](#).

## 10.409.2 Member Function Documentation

### 10.409.2.1 [create\\_file\\_source\\_hook\(\)](#)

void TwoBitPackedFileSignalSource::create\_file\_source\_hook () [override], [protected], [virtual]  
Reimplemented from [FileSourceBase](#).

### 10.409.2.2 [itemTypeToSize\(\)](#)

std::tuple< size\_t, bool > TwoBitPackedFileSignalSource::itemTypeToSize () [override], [protected], [virtual]

Compute the item size, from the [item\\_type\(\)](#). Subclasses may constrain types that don't make.

Reimplemented from [FileSourceBase](#).

### 10.409.2.3 [packetsPerSample\(\)](#)

double TwoBitPackedFileSignalSource::packetsPerSample () const [override], [protected], [virtual]

The number of (possibly unpacked) samples in a (raw) file sample (default=1)

Reimplemented from [FileSourceBase](#).

### 10.409.2.4 [pre\\_connect\\_hook\(\)](#)

void TwoBitPackedFileSignalSource::pre\_connect\_hook (gr::top\_block\_sptr top\_block) [override], [protected], [virtual]

Reimplemented from [FileSourceBase](#).

#### 10.409.2.5 pre\_disconnect\_hook()

```
void TwoBitPackedFileSignalSource::pre_disconnect_hook (  
    gr::top_block_sptr top_block) [override], [protected], [virtual]
```

Reimplemented from [FileSourceBase](#).

#### 10.409.2.6 source()

```
gnss_shared_ptr< gr::block > TwoBitPackedFileSignalSource::source () const [override], [protected],  
[virtual]
```

Abstracted front-end source. Sub-classes may override if they create specialized chains to decode source files into a usable format.

Reimplemented from [FileSourceBase](#).

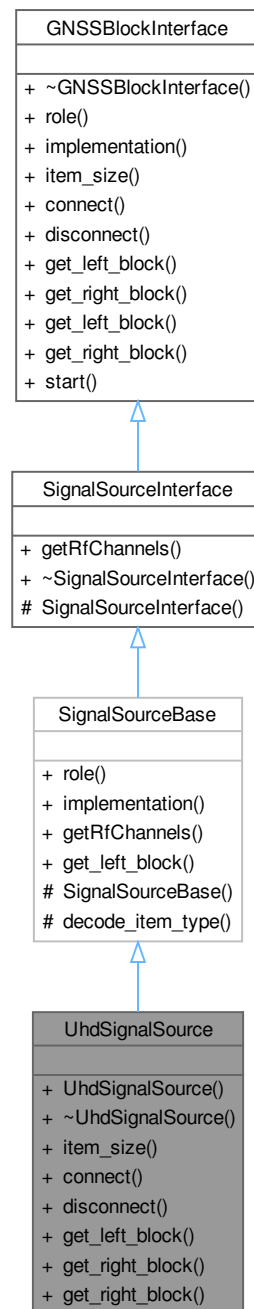
The documentation for this class was generated from the following file:

- [two\\_bit\\_packed\\_file\\_signal\\_source.h](#)

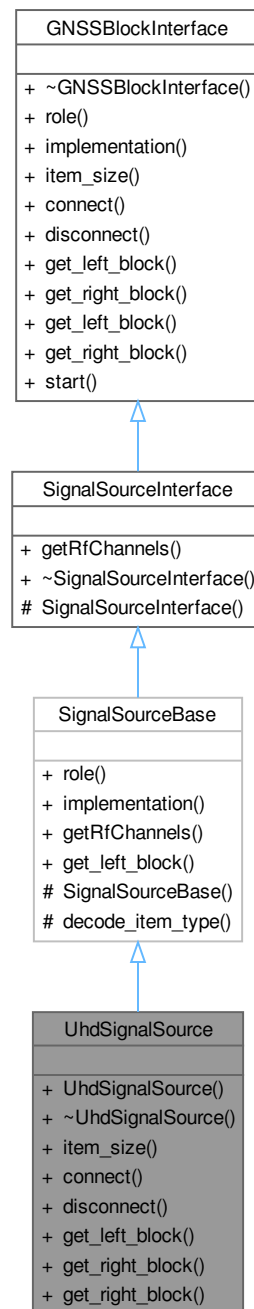
## 10.410 UhdSignalSource Class Reference

This class reads samples from a UHD device (see <http://code.ettus.com/redmine/ettus/projects/uhd/wiki>)  
`#include <uhd_signal_source.h>`

Inheritance diagram for UhdSignalSource:



Collaboration diagram for UhdSignalSource:



## Public Member Functions

- **UhdSignalSource** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_stream, unsigned int out\_stream, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)
- size\_t [item\\_size](#) () override
- void [connect](#) (gr::top\_block\_sptr top\_block) override
- void [disconnect](#) (gr::top\_block\_sptr top\_block) override
- gr::basic\_block\_sptr [get\\_left\\_block](#) () override

- `gr::basic_block_sptr` [get\\_right\\_block](#) () override
- `gr::basic_block_sptr` [get\\_right\\_block](#) (int RF\_channel) override

### Public Member Functions inherited from [SignalSourceBase](#)

- `std::string` [role](#) () final
- `std::string` [implementation](#) () final
- `size_t` [getRfChannels](#) () const override

### Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual `gr::basic_block_sptr` [get\\_left\\_block](#) (int RF\_channel)
- virtual void [start](#) ()

*Start the flow of samples if needed.*

### Additional Inherited Members

### Protected Member Functions inherited from [SignalSourceBase](#)

- [SignalSourceBase](#) ([ConfigurationInterface](#) const \*configuration, std::string role, std::string impl)  
*Constructor.*
- `size_t` [decode\\_item\\_type](#) (std::string const &item\_type, bool \*is\_interleaved=nullptr, bool throw\_on\_↵ error=false)  
*utility for decoding passed ".item\_type" values*

#### 10.410.1 Detailed Description

This class reads samples from a UHD device (see <http://code.ettus.com/redmine/ettus/projects/uhd/wiki>)  
Definition at line 41 of file [uhd\\_signal\\_source.h](#).

#### 10.410.2 Member Function Documentation

##### 10.410.2.1 connect()

```
void UhdSignalSource::connect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

##### 10.410.2.2 disconnect()

```
void UhdSignalSource::disconnect (
    gr::top_block_sptr top_block) [override], [virtual]
```

Implements [GNSSBlockInterface](#).

##### 10.410.2.3 get\_left\_block()

```
gr::basic_block_sptr UhdSignalSource::get_left_block () [override], [virtual]
```

Reimplemented from [SignalSourceBase](#).

##### 10.410.2.4 get\_right\_block() [1/2]

```
gr::basic_block_sptr UhdSignalSource::get_right_block () [override], [virtual]
```

Implements [GNSSBlockInterface](#).

##### 10.410.2.5 get\_right\_block() [2/2]

```
gr::basic_block_sptr UhdSignalSource::get_right_block (
    int RF_channel) [override], [virtual]
```

Reimplemented from [GNSSBlockInterface](#).



### 10.410.2.6 item\_size()

size\_t UhdSignalSource::item\_size () [inline], [override], [virtual]

Implements [GNSSBlockInterface](#).

Definition at line 50 of file [uhd\\_signal\\_source.h](#).

The documentation for this class was generated from the following file:

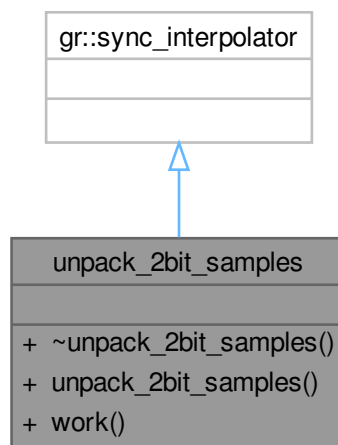
- [uhd\\_signal\\_source.h](#)

## 10.411 unpack\_2bit\_samples Class Reference

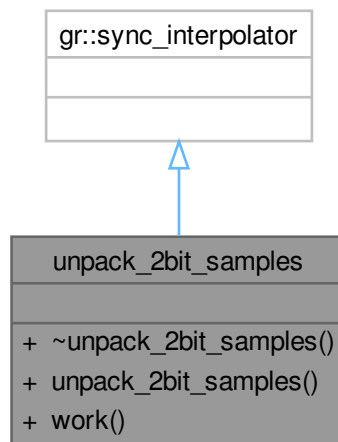
This class takes 2 bit samples that have been packed into bytes or shorts as input and generates a byte for each sample. It generates eight times as much data as is input (every two bits become 16 bits)

#include <unpack\_2bit\_samples.h>

Inheritance diagram for unpack\_2bit\_samples:



Collaboration diagram for `unpack_2bit_samples`:



### Public Member Functions

- **unpack\_2bit\_samples** (bool big\_endian\_bytes, size\_t item\_size, bool big\_endian\_items, bool reverse\_interleaving)
- int **work** (int noutput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)

### Friends

- `unpack_2bit_samples_sptr` **make\_unpack\_2bit\_samples\_sptr** (bool big\_endian\_bytes, size\_t item\_size, bool big\_endian\_items, bool reverse\_interleaving)

### 10.411.1 Detailed Description

This class takes 2 bit samples that have been packed into bytes or shorts as input and generates a byte for each sample. It generates eight times as much data as is input (every two bits become 16 bits)

Definition at line 84 of file [unpack\\_2bit\\_samples.h](#).

The documentation for this class was generated from the following file:

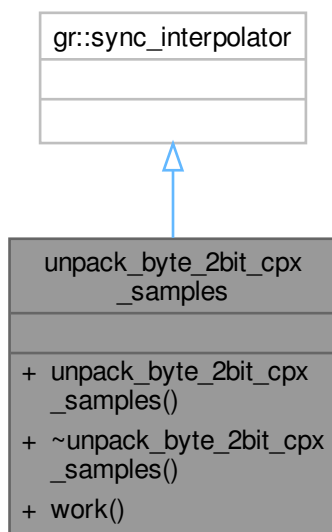
- [unpack\\_2bit\\_samples.h](#)

## 10.412 unpack\_byte\_2bit\_cpx\_samples Class Reference

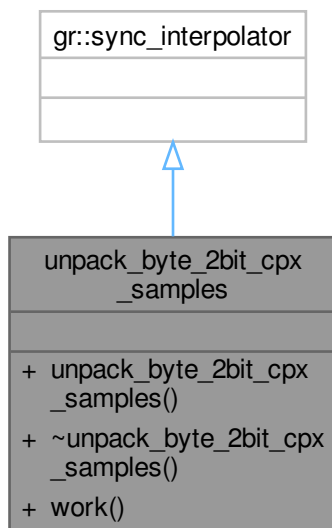
This class implements conversion between byte packet samples to 2bit\_cpx samples 1 byte = 2 x complex 2bit I, + 2bit Q samples.

```
#include <unpack_byte_2bit_cpx_samples.h>
```

Inheritance diagram for unpack\_byte\_2bit\_cpx\_samples:



Collaboration diagram for unpack\_byte\_2bit\_cpx\_samples:



### Public Member Functions

- `int` **work** (`int` noutput\_items, `gr_vector_const_void_star` &input\_items, `gr_vector_void_star` &output\_items)

## Friends

- `unpack_byte_2bit_cpx_samples_sptr` `make_unpack_byte_2bit_cpx_samples_sptr()`

### 10.412.1 Detailed Description

This class implements conversion between byte packet samples to 2bit\_cpx samples 1 byte = 2 x complex 2bit I, + 2bit Q samples.

Definition at line 44 of file [unpack\\_byte\\_2bit\\_cpx\\_samples.h](#).

The documentation for this class was generated from the following file:

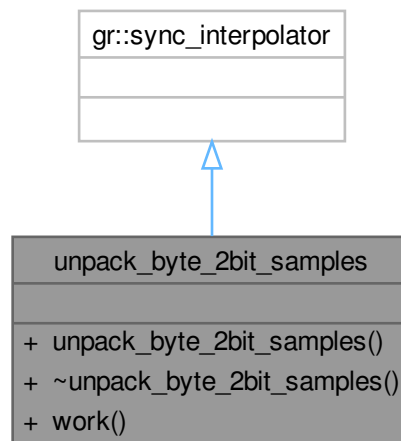
- [unpack\\_byte\\_2bit\\_cpx\\_samples.h](#)

### 10.413 `unpack_byte_2bit_samples` Class Reference

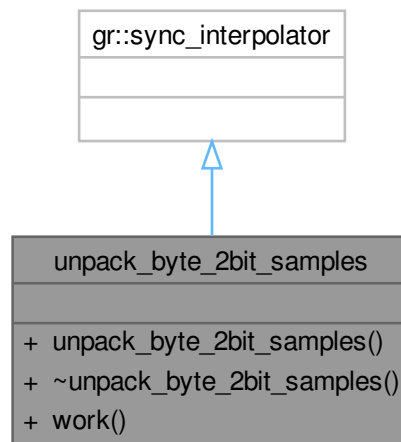
This class implements conversion between byte packet samples to 2bit samples 1 byte = 4 2bit samples.

`#include <unpack_byte_2bit_samples.h>`

Inheritance diagram for `unpack_byte_2bit_samples`:



Collaboration diagram for unpack\_byte\_2bit\_samples:



### Public Member Functions

- `int work (int noutput_items, gr_vector_const_void_star &input_items, gr_vector_void_star &output_items)`

### Friends

- `unpack_byte_2bit_samples_sptr make_unpack_byte_2bit_samples_sptr ()`

### 10.413.1 Detailed Description

This class implements conversion between byte packet samples to 2bit samples 1 byte = 4 2bit samples.

Definition at line 40 of file [unpack\\_byte\\_2bit\\_samples.h](#).

The documentation for this class was generated from the following file:

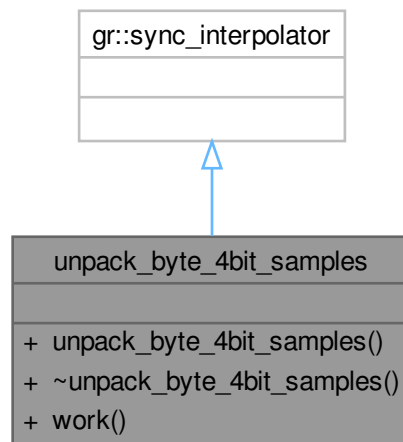
- [unpack\\_byte\\_2bit\\_samples.h](#)

## 10.414 unpack\_byte\_4bit\_samples Class Reference

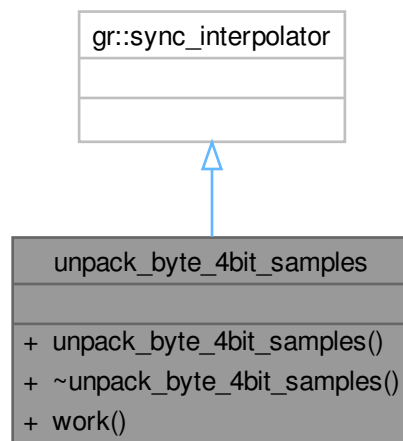
This class implements conversion between byte packet samples to 4bit\_cpx samples 1 byte = 1 x complex 4bit I, + 4bit Q samples.

```
#include <unpack_byte_4bit_samples.h>
```

Inheritance diagram for `unpack_byte_4bit_samples`:



Collaboration diagram for `unpack_byte_4bit_samples`:



### Public Member Functions

- `int` **work** (`int` noutput\_items, `gr_vector_const_void_star` &input\_items, `gr_vector_void_star` &output\_items)

### Friends

- `unpack_byte_4bit_samples_sptr` **make\_unpack\_byte\_4bit\_samples\_sptr** ()

### 10.414.1 Detailed Description

This class implements conversion between byte packet samples to 4bit\_cpx samples 1 byte = 1 x complex 4bit I, + 4bit Q samples.

Definition at line 42 of file [unpack\\_byte\\_4bit\\_samples.h](#).

The documentation for this class was generated from the following file:

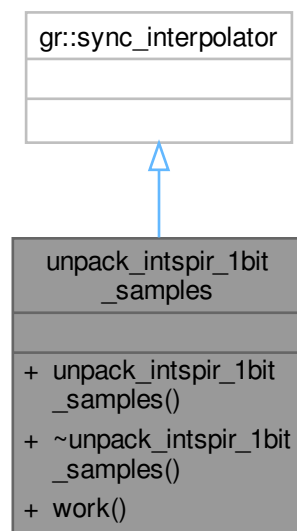
- [unpack\\_byte\\_4bit\\_samples.h](#)

## 10.415 unpack\_intspir\_1bit\_samples Class Reference

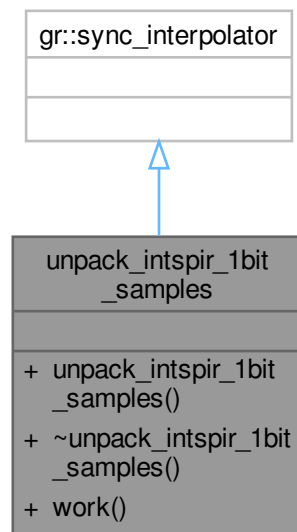
This class implements conversion between byte packet samples to 2bit samples 1 byte = 4 2bit samples.

#include <unpack\_intspir\_1bit\_samples.h>

Inheritance diagram for unpack\_intspir\_1bit\_samples:



Collaboration diagram for `unpack_intspir_1bit_samples`:



### Public Member Functions

- `int` **work** (`int` noutput\_items, `gr_vector_const_void_star` &input\_items, `gr_vector_void_star` &output\_items)

### Friends

- `unpack_intspir_1bit_samples_sptr` **make\_unpack\_intspir\_1bit\_samples\_sptr** ()

### 10.415.1 Detailed Description

This class implements conversion between byte packet samples to 2bit samples 1 byte = 4 2bit samples.

Definition at line 41 of file [unpack\\_intspir\\_1bit\\_samples.h](#).

The documentation for this class was generated from the following file:

- [unpack\\_intspir\\_1bit\\_samples.h](#)

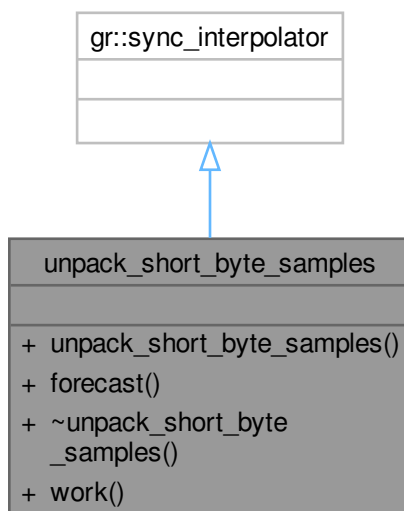
## 10.416 unpack\_short\_byte\_samples Class Reference

This class implements conversion between short packet samples to byte samples 1 short = 2 byte samples.

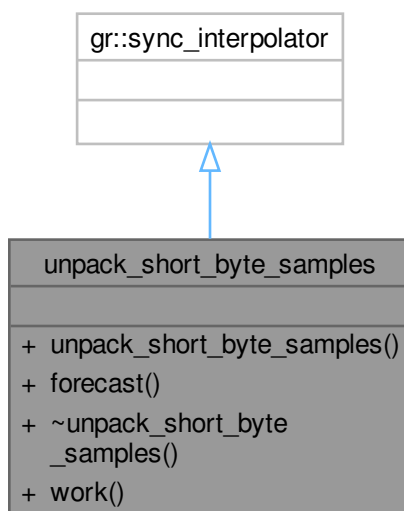
`#include <unpack_short_byte_samples.h>`



Inheritance diagram for unpack\_short\_byte\_samples:



Collaboration diagram for unpack\_short\_byte\_samples:



### Public Member Functions

- void **forecast** (int noutput\_items, gr\_vector\_int &ninput\_items\_required)
- int **work** (int noutput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)

## Friends

- `unpack_short_byte_samples_sptr make_unpack_short_byte_samples_sptr ()`

### 10.416.1 Detailed Description

This class implements conversion between short packet samples to byte samples 1 short = 2 byte samples.

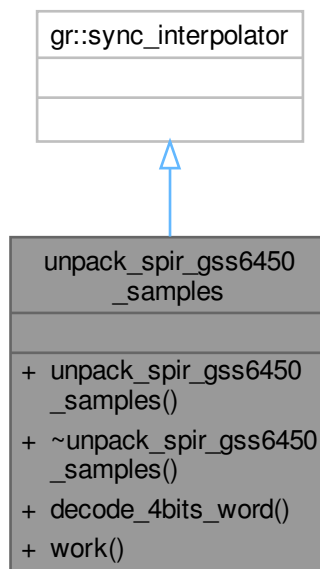
Definition at line 41 of file [unpack\\_short\\_byte\\_samples.h](#).

The documentation for this class was generated from the following file:

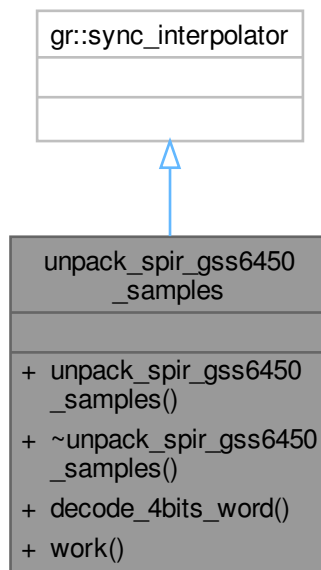
- `unpack_short_byte_samples.h`

### 10.417 `unpack_spir_gss6450_samples` Class Reference

Inheritance diagram for `unpack_spir_gss6450_samples`:



Collaboration diagram for unpack\_spir\_gss6450\_samples:



### Public Member Functions

- **unpack\_spir\_gss6450\_samples** (int adc\_nbit)
- void **decode\_4bits\_word** (uint32\_t input\_uint32, gr\_complex \*out, int adc\_bits\_)
- int **work** (int noutput\_items, gr\_vector\_const\_void\_star &input\_items, gr\_vector\_void\_star &output\_items)

### Friends

- unpack\_spir\_gss6450\_samples\_sptr **make\_unpack\_spir\_gss6450\_samples\_sptr** (int adc\_nbit)

### 10.417.1 Detailed Description

Definition at line 37 of file [unpack\\_spir\\_gss6450\\_samples.h](#).

The documentation for this class was generated from the following file:

- [unpack\\_spir\\_gss6450\\_samples.h](#)

## 10.418 UnscentedFilter Class Reference

Collaboration diagram for UnscentedFilter:

UnscentedFilter
<ul style="list-style-type: none"> <li>+ UnscentedFilter()</li> <li>+ UnscentedFilter()</li> <li>+ UnscentedFilter()</li> <li>+ ~UnscentedFilter()</li> <li>+ initialize()</li> <li>+ predict_sequential()</li> <li>+ update_sequential()</li> <li>+ get_x_pred()</li> <li>+ get_P_x_pred()</li> <li>+ get_x_est()</li> <li>+ get_P_x_est()</li> </ul>

### Public Member Functions

- **UnscentedFilter** (int nx)
- **UnscentedFilter** (const arma::vec &x\_pred\_0, const arma::mat &P\_x\_pred\_0)
- void **initialize** (const arma::mat &x\_pred\_0, const arma::mat &P\_x\_pred\_0)
- void **predict\_sequential** (const arma::vec &x\_post, const arma::mat &P\_x\_post, [ModelFunction](#) \*transition\_fcn, const arma::mat &noise\_covariance)
- void **update\_sequential** (const arma::vec &z\_upd, const arma::vec &x\_pred, const arma::mat &P\_x\_pred, [ModelFunction](#) \*measurement\_fcn, const arma::mat &noise\_covariance)
- arma::mat **get\_x\_pred** () const
- arma::mat **get\_P\_x\_pred** () const
- arma::mat **get\_x\_est** () const
- arma::mat **get\_P\_x\_est** () const

### 10.418.1 Detailed Description

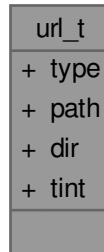
Definition at line 83 of file [nonlinear\\_tracking.h](#).

The documentation for this class was generated from the following file:

- [nonlinear\\_tracking.h](#)

## 10.419 url\_t Struct Reference

Collaboration diagram for url\_t:



### Public Attributes

- char [type](#) [32]
- char [path](#) [1024]
- char [dir](#) [1024]
- double [tint](#)

### 10.419.1 Detailed Description

Definition at line [907](#) of file [rtklib.h](#).

### 10.419.2 Member Data Documentation

#### 10.419.2.1 dir

```
char url_t::dir[1024]
```

Definition at line [911](#) of file [rtklib.h](#).

#### 10.419.2.2 path

```
char url_t::path[1024]
```

Definition at line [910](#) of file [rtklib.h](#).

#### 10.419.2.3 tint

```
double url_t::tint
```

Definition at line [912](#) of file [rtklib.h](#).

#### 10.419.2.4 type

```
char url_t::type[32]
```

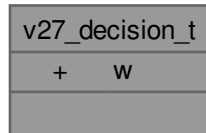
Definition at line [909](#) of file [rtklib.h](#).

The documentation for this struct was generated from the following file:

- [rtklib.h](#)

## 10.420 v27\_decision\_t Struct Reference

Collaboration diagram for v27\_decision\_t:



### Public Attributes

- unsigned int [w](#) [2]

### 10.420.1 Detailed Description

Definition at line 38 of file [fec.h](#).

### 10.420.2 Member Data Documentation

#### 10.420.2.1 w

unsigned int v27\_decision\_t::w[2]

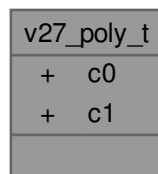
Definition at line 40 of file [fec.h](#).

The documentation for this struct was generated from the following file:

- [fec.h](#)

## 10.421 v27\_poly\_t Struct Reference

Collaboration diagram for v27\_poly\_t:



### Public Attributes

- unsigned char [c0](#) [32]
- unsigned char [c1](#) [32]

### 10.421.1 Detailed Description

Definition at line 32 of file [fec.h](#).

### 10.421.2 Member Data Documentation

#### 10.421.2.1 c0

```
unsigned char v27_poly_t::c0[32]
```

Definition at line 34 of file [fec.h](#).

#### 10.421.2.2 c1

```
unsigned char v27_poly_t::c1[32]
```

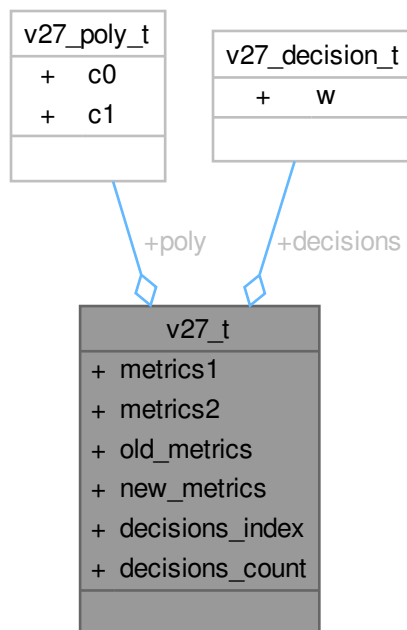
Definition at line 35 of file [fec.h](#).

The documentation for this struct was generated from the following file:

- [fec.h](#)

## 10.422 v27\_t Struct Reference

Collaboration diagram for v27\_t:



### Public Attributes

- unsigned int [metrics1](#) [64]
- unsigned int [metrics2](#) [64]
- unsigned int \* [old\\_metrics](#)
- unsigned int \* [new\\_metrics](#)
- const [v27\\_poly\\_t](#) \* [poly](#)

- [v27\\_decision\\_t](#) \* [decisions](#)
- unsigned int [decisions\\_index](#)
- unsigned int [decisions\\_count](#)

### 10.422.1 Detailed Description

Definition at line 45 of file [fec.h](#).

### 10.422.2 Member Data Documentation

#### 10.422.2.1 decisions

[v27\\_decision\\_t](#)\* [v27\\_t::decisions](#)

Definition at line 52 of file [fec.h](#).

#### 10.422.2.2 decisions\_count

unsigned int [v27\\_t::decisions\\_count](#)

Definition at line 54 of file [fec.h](#).

#### 10.422.2.3 decisions\_index

unsigned int [v27\\_t::decisions\\_index](#)

Definition at line 53 of file [fec.h](#).

#### 10.422.2.4 metrics1

unsigned int [v27\\_t::metrics1](#)[64]

Definition at line 47 of file [fec.h](#).

#### 10.422.2.5 metrics2

unsigned int [v27\\_t::metrics2](#)[64]

Definition at line 48 of file [fec.h](#).

#### 10.422.2.6 new\_metrics

unsigned int \* [v27\\_t::new\\_metrics](#)

Definition at line 50 of file [fec.h](#).

#### 10.422.2.7 old\_metrics

unsigned int\* [v27\\_t::old\\_metrics](#)

Definition at line 50 of file [fec.h](#).

#### 10.422.2.8 poly

const [v27\\_poly\\_t](#)\* [v27\\_t::poly](#)

Definition at line 51 of file [fec.h](#).

The documentation for this struct was generated from the following file:

- [fec.h](#)

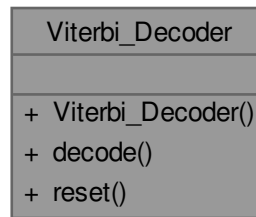
## 10.423 Viterbi\_Decoder Class Reference

Class that implements a Viterbi decoder.

```
#include <viterbi_decoder.h>
```



Collaboration diagram for Viterbi\_Decoder:



### Public Member Functions

- [Viterbi\\_Decoder](#) (int32\_t *KK*, int32\_t *nn*, int32\_t *LL*, const std::array< int32\_t, 2 > &*g*)  
*Constructor of a Viterbi decoder.*
- void [decode](#) (std::vector< int32\_t > &output\_u\_int, const std::vector< float > &input\_c)  
*Uses the Viterbi algorithm to perform hard-decision decoding of a convolutional code.*
- void [reset](#) ()  
*Reset internal status.*

### 10.423.1 Detailed Description

Class that implements a Viterbi decoder.

Definition at line 34 of file [viterbi\\_decoder.h](#).

### 10.423.2 Constructor & Destructor Documentation

#### 10.423.2.1 Viterbi\_Decoder()

```
Viterbi_Decoder::Viterbi_Decoder (
    int32_t KK,
    int32_t nn,
    int32_t LL,
    const std::array< int32_t, 2 > & g)
```

Constructor of a Viterbi decoder.

#### Parameters

in	<i>KK</i>	Constraint length
in	<i>nn</i>	Coding rate 1/n
in	<i>LL</i>	Data length
in	<i>g</i>	Polynomial G1 and G2

### 10.423.3 Member Function Documentation

#### 10.423.3.1 decode()

```
void Viterbi_Decoder::decode (
    std::vector< int32_t > & output_u_int,
    const std::vector< float > & input_c)
```

Uses the Viterbi algorithm to perform hard-decision decoding of a convolutional code.

## Parameters

out	output_u_int	Hard decisions on the data bits
in	input_c	The received signal in LLR-form. For BPSK, must be in form $r = 2*a*y/(\sigma^2)$ .

**10.423.3.2 reset()**

```
void Viterbi_Decoder::reset ()
```

Reset internal status.

The documentation for this class was generated from the following file:

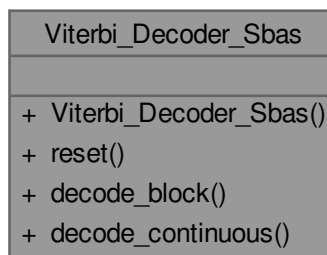
- [viterbi\\_decoder.h](#)

**10.424 Viterbi\_Decoder\_Sbas Class Reference**

Class that implements a Viterbi decoder.

```
#include <viterbi_decoder_sbas.h>
```

Collaboration diagram for Viterbi\_Decoder\_Sbas:

**Public Member Functions**

- **Viterbi\_Decoder\_Sbas** (const int g\_encoder[], int KK, int nn)
- void **reset** ()
- float **decode\_block** (const double input\_c[], int \*output\_u\_int, int LL)  
*Uses the Viterbi algorithm to perform hard-decision decoding of a convolutional code.*
- float **decode\_continuous** (const double sym[], int traceback\_depth, int bits[], int nbits\_requested, int &nbits\_decoded)

**10.424.1 Detailed Description**

Class that implements a Viterbi decoder.

Definition at line 34 of file [viterbi\\_decoder\\_sbas.h](#).

**10.424.2 Member Function Documentation****10.424.2.1 decode\_block()**

```
float Viterbi_Decoder_Sbas::decode_block (
    const double input_c[],
```

```
int * output_u_int,  
int LL)
```

Uses the Viterbi algorithm to perform hard-decision decoding of a convolutional code.

#### Parameters

in	<i>input_c[]</i>	The received signal in LLR-form. For BPSK, must be in form $r = 2*a*y/(sigma^2)$ .
in	<i>LL</i>	The number of data bits to be decoded (does not include the mm zero-tail-bits)

#### Returns

output\_u\_int[] Hard decisions on the data bits (without the mm zero-tail-bits)

The documentation for this class was generated from the following file:

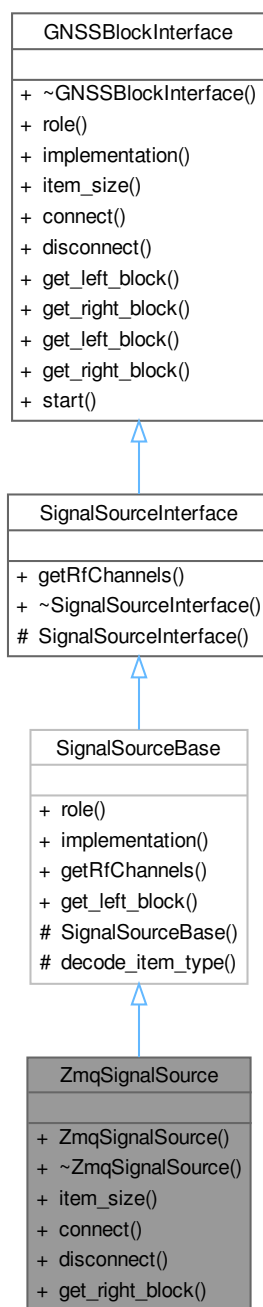
- [viterbi\\_decoder\\_sbass.h](#)

## 10.425 ZmqSignalSource Class Reference

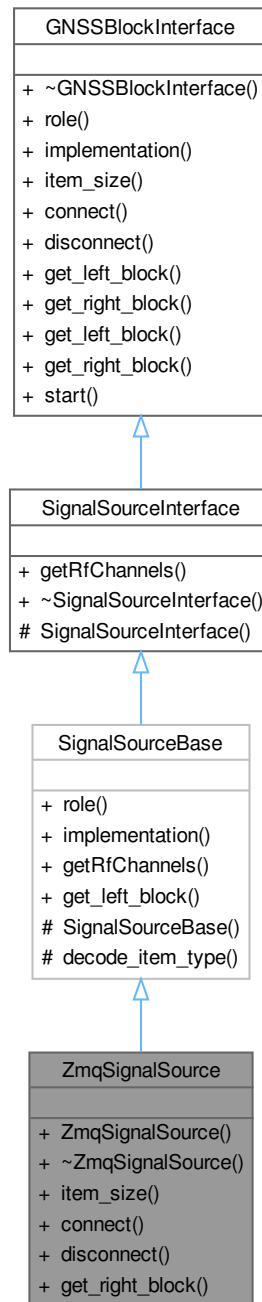
This class supports the following properties:

```
#include <zmq_signal_source.h>
```

Inheritance diagram for ZmqSignalSource:



Collaboration diagram for ZmqSignalSource:



### Public Member Functions

- **ZmqSignalSource** (const [ConfigurationInterface](#) \*configuration, const std::string &role, unsigned int in\_stream, unsigned int out\_stream, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)
- `size_t item_size ()` override
- auto `connect (gr::top_block_sptr top_block) -> void` override
- auto `disconnect (gr::top_block_sptr top_block) -> void` override
- auto `get_right_block () -> gr::basic_block_sptr` override

## Public Member Functions inherited from [SignalSourceBase](#)

- `std::string role ()` final
- `std::string implementation ()` final
- `size_t getRfChannels ()` const override
- `gr::basic_block_sptr get_left_block ()` override

## Public Member Functions inherited from [GNSSBlockInterface](#)

- virtual `gr::basic_block_sptr get_left_block (int RF_channel)`
- virtual `gr::basic_block_sptr get_right_block (int RF_channel)`
- virtual void `start ()`

*Start the flow of samples if needed.*

## Additional Inherited Members

## Protected Member Functions inherited from [SignalSourceBase](#)

- [SignalSourceBase](#) ([ConfigurationInterface](#) const \*configuration, std::string role, std::string impl)  
*Constructor.*
- `size_t decode_item_type (std::string const &item_type, bool *is_interleaved=nullptr, bool throw_on_↔ error=false)`  
*utility for decoding passed ".item\_type" values*

### 10.425.1 Detailed Description

This class supports the following properties:

`.endpoint` - the ZMQ endpoint to be connected to  
`.vlen` - vector length of the input items (default 1, one item) this must match the size of the publisher!  
`.pass_tags` - boolean flag if tags should be propagated (default false)  
`.timeout_ms` - receive timeout, in milliseconds (default 100)  
`.hwm` - ZMQ high water mark (default -1, ZMQ default)  
`.item_type` - data type of the samples (default "gr\_complex")  
 (probably should be abstracted to the base class)  
`.dump` - whether to archive input data  
`.dump_filename` - if dumping, path to file for output  
 Definition at line 51 of file [zmq\\_signal\\_source.h](#).

### 10.425.2 Member Function Documentation

#### 10.425.2.1 connect()

```
auto ZmqSignalSource::connect (
    gr::top_block_sptr top_block) -> void [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.425.2.2 disconnect()

```
auto ZmqSignalSource::disconnect (
    gr::top_block_sptr top_block) -> void [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.425.2.3 get\_right\_block()

```
auto ZmqSignalSource::get_right_block () -> gr::basic_block_sptr [override], [virtual]
```

Implements [GNSSBlockInterface](#).

#### 10.425.2.4 item\_size()

`size_t ZmqSignalSource::item_size () [override], [virtual]`

Implements [GNSSBlockInterface](#).

The documentation for this class was generated from the following file:

- [zmq\\_signal\\_source.h](#)





# Chapter 11

## File Documentation

### 11.1 beidou\_b1i\_pcps\_acquisition.h File Reference

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Beidou B1I signals.

```
#include "channel_fsm.h"
#include "complex_byte_to_float_x2.h"
#include "gnss_synchro.h"
#include "pcps_acquisition.h"
#include <gnuradio/blocks/float_to_complex.h>
#include <gnuradio/blocks/stream_to_vector.h>
#include <volk_gnssssdr/volk_gnssssdr_alloc.h>
#include <stdint>
#include <memory>
#include <string>
#include <utility>
```

Include dependency graph for beidou\_b1i\_pcps\_acquisition.h:



#### Classes

- class [BeidouB1IPcpsAcquisition](#)

*This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.*

#### 11.1.1 Detailed Description

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Beidou B1I signals.

#### Authors

- Sergi Segura, 2018. [sergi.segura.munoz\(at\)gmail.com](mailto:sergi.segura.munoz(at)gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [beidou\\_b1i\\_pcps\\_acquisition.h](#).

---

### 11.2 beidou\_b1i\_pcps\_acquisition.h

[Go to the documentation of this file.](#)

00001 /\*!

```

00002  * \file beidou_bli_pcps_acquisition.h
00003  * \brief Adapts a PCPS acquisition block to an AcquisitionInterface for
00004  * Beidou B1I signals
00005  * \authors <ul>
00006  *     <li> Sergi Segura, 2018. sergi.segura.munoz(at)gmail.com
00007  *     </li>
00008  * </ul>
00009  * -----
00010  *
00011  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00012  * This file is part of GNSS-SDR.
00013  *
00014  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00015  * SPDX-License-Identifier: GPL-3.0-or-later
00016  *
00017  * -----
00018  */
00019
00020 #ifndef GNSS_SDR_BEIDOU_B1I_PCPS_ACQUISITION_H
00021 #define GNSS_SDR_BEIDOU_B1I_PCPS_ACQUISITION_H
00022
00023 #include "channel_fsm.h"
00024 #include "complex_byte_to_float_x2.h"
00025 #include "gnss_synchro.h"
00026 #include "pcps_acquisition.h"
00027 #include <gnuradio/blocks/float_to_complex.h>
00028 #include <gnuradio/blocks/stream_to_vector.h>
00029 #include <volk_gnssdr/volk_gnssdr_alloc.h>
00030 #include <stdint>
00031 #include <memory>
00032 #include <string>
00033 #include <utility>
00034
00035 /** \addtogroup Acquisition
00036  * \{ */
00037 /** \addtogroup Acq_adapters
00038  * \{ */
00039
00040
00041 class ConfigurationInterface;
00042
00043 /*!
00044  * \brief This class adapts a PCPS acquisition block to an AcquisitionInterface
00045  * for GPS L1 C/A signals
00046  */
00047 class BeidouBliPcpsAcquisition : public AcquisitionInterface
00048 {
00049 public:
00050     BeidouBliPcpsAcquisition(const ConfigurationInterface* configuration,
00051                             const std::string& role, unsigned int in_streams,
00052                             unsigned int out_streams);
00053
00054     ~BeidouBliPcpsAcquisition() = default;
00055
00056     inline std::string role() override
00057     {
00058         return role_;
00059     }
00060
00061     /*!
00062     * \brief Returns "BEIDOU_B1I_PCPS_Acquisition"
00063     */
00064     inline std::string implementation() override
00065     {
00066         return "BEIDOU_B1I_PCPS_Acquisition";
00067     }
00068
00069     inline size_t item_size() override
00070     {
00071         return item_size_;
00072     }
00073
00074     void connect(gr::top_block_sptr top_block) override;
00075     void disconnect(gr::top_block_sptr top_block) override;
00076     gr::basic_block_sptr get_left_block() override;
00077     gr::basic_block_sptr get_right_block() override;
00078
00079     /*!
00080     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00081     * to efficiently exchange synchronization data between acquisition and
00082     * tracking blocks
00083     */
00084     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00085
00086     /*!
00087     * \brief Set acquisition channel unique ID
00088     */

```

```

00089     inline void set_channel(unsigned int channel) override
00090     {
00091         channel_ = channel;
00092         acquisition_>set_channel(channel_);
00093     }
00094
00095     /*!
00096     * \brief Set channel fsm associated to this acquisition instance
00097     */
00098     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm) override
00099     {
00100         channel_fsm_ = std::move(channel_fsm);
00101         acquisition_>set_channel_fsm(channel_fsm_);
00102     }
00103
00104     /*!
00105     * \brief Set statistics threshold of PCPS algorithm
00106     */
00107     void set_threshold(float threshold) override;
00108
00109     /*!
00110     * \brief Set maximum Doppler off grid search
00111     */
00112     void set_doppler_max(uint32_t doppler_max) override;
00113
00114     /*!
00115     * \brief Set Doppler steps for the grid search
00116     */
00117     void set_doppler_step(uint32_t doppler_step) override;
00118
00119     /*!
00120     * \brief Initializes acquisition algorithm.
00121     */
00122     void init() override;
00123
00124     /*!
00125     * \brief Sets local code for GPS L1/CA PCPS acquisition algorithm.
00126     */
00127     void set_local_code() override;
00128
00129     /*!
00130     * \brief Returns the maximum peak of grid search
00131     */
00132     signed int mag() override;
00133
00134     /*!
00135     * \brief Restart acquisition algorithm
00136     */
00137     void reset() override;
00138
00139     /*!
00140     * \brief If state = 1, it forces the block to start acquiring from the first sample
00141     */
00142     void set_state(int state) override;
00143
00144     /*!
00145     * \brief Stop running acquisition
00146     */
00147     void stop_acquisition() override;
00148
00149     /*!
00150     * \brief Sets the resampler latency to account it in the acquisition code delay estimation
00151     */
00152     void set_resampler_latency(uint32_t latency_samples) override;
00153
00154 private:
00155     pcps_acquisition_sptr acquisition_;
00156     volk_gnssdr::vector<std::complex<float>> code_;
00157     std::weak_ptr<ChannelFsm> channel_fsm_;
00158     gr::blocks::float_to_complex::sptr float_to_complex_;
00159     complex_byte_to_float_x2_sptr cbyte_to_float_x2_;
00160     Gnss_Synchro* gnss_synchro_;
00161     Acq_Conf acq_parameters_;
00162     std::string item_type_;
00163     std::string role_;
00164     std::string dump_filename_;
00165     size_t item_size_;
00166     int64_t fs_in_;
00167     float threshold_;
00168     unsigned int vector_length_;
00169     unsigned int code_length_;
00170     unsigned int channel_;
00171     unsigned int doppler_max_;
00172     unsigned int doppler_step_;
00173     unsigned int num_codes_;
00174     unsigned int in_streams_;
00175     unsigned int out_streams_;

```

```

00176 };
00177
00178
00179 /** \} */
00180 /** \} */
00181 #endif // GNSS_SDR_BEIDOU_B1I_PCPS_ACQUISITION_H

```

## 11.3 beidou\_b3i\_pcps\_acquisition.h File Reference

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Beidou B3I signals.

```

#include "acq_conf.h"
#include "channel_fsm.h"
#include "complex_byte_to_float_x2.h"
#include "gnss_synchro.h"
#include "pcps_acquisition.h"
#include <gnuradio/blocks/float_to_complex.h>
#include <gnuradio/blocks/stream_to_vector.h>
#include <volk_gnssssdr/volk_gnssssdr_alloc.h>
#include <stdint>
#include <memory>
#include <string>
#include <utility>

```

Include dependency graph for beidou\_b3i\_pcps\_acquisition.h:



### Classes

- class [BeidouB3iPcpsAcquisition](#)

*This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for BeiDou B3I signals.*

### 11.3.1 Detailed Description

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Beidou B3I signals.

#### Author

Damian Miralles, 2019. [dmiralles2009@gmail.com](mailto:dmiralles2009@gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [beidou\\_b3i\\_pcps\\_acquisition.h](#).

---

## 11.4 beidou\_b3i\_pcps\_acquisition.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file beidou_b3i_pcps_acquisition.h
00003  * \brief Adapts a PCPS acquisition block to an AcquisitionInterface for
00004  *       Beidou B3I signals
00005  * \author Damian Miralles, 2019. dmiralles2009@gmail.com
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later

```

```

00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_BEIDOU_B3I_PCPS_ACQUISITION_H
00019 #define GNSS_SDR_BEIDOU_B3I_PCPS_ACQUISITION_H
00020
00021 #include "acq_conf.h"
00022 #include "channel_fsm.h"
00023 #include "complex_byte_to_float_x2.h"
00024 #include "gnss_synchro.h"
00025 #include "pcps_acquisition.h"
00026 #include <gnuradio/blocks/float_to_complex.h>
00027 #include <gnuradio/blocks/stream_to_vector.h>
00028 #include <volk_gnssssdr/volk_gnssssdr_alloc.h>
00029 #include <stdint>
00030 #include <memory>
00031 #include <string>
00032 #include <utility>
00033
00034 /** \addtogroup Acquisition
00035  * \{ */
00036 /** \addtogroup Acq_adapters
00037  * \{ */
00038
00039
00040 class ConfigurationInterface;
00041
00042 /*!
00043  * \brief This class adapts a PCPS acquisition block to an AcquisitionInterface
00044  * for BeiDou B3I signals
00045  */
00046 class BeidouB3iPcpsAcquisition : public AcquisitionInterface
00047 {
00048 public:
00049     BeidouB3iPcpsAcquisition(const ConfigurationInterface* configuration,
00050                             const std::string& role, unsigned int in_streams,
00051                             unsigned int out_streams);
00052
00053     ~BeidouB3iPcpsAcquisition() = default;
00054
00055     inline std::string role() override
00056     {
00057         return role_;
00058     }
00059
00060     /*!
00061     * \brief Returns "BEIDOU_B3I_PCPS_Acquisition"
00062     */
00063     inline std::string implementation() override
00064     {
00065         return "BEIDOU_B3I_PCPS_Acquisition";
00066     }
00067
00068     inline size_t item_size() override
00069     {
00070         return item_size_;
00071     }
00072
00073     void connect(gr::top_block_sptr top_block) override;
00074     void disconnect(gr::top_block_sptr top_block) override;
00075     gr::basic_block_sptr get_left_block() override;
00076     gr::basic_block_sptr get_right_block() override;
00077
00078     /*!
00079     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00080     * to efficiently exchange synchronization data between acquisition and
00081     * tracking blocks
00082     */
00083     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00084
00085     /*!
00086     * \brief Set acquisition channel unique ID
00087     */
00088     inline void set_channel(unsigned int channel) override
00089     {
00090         channel_ = channel;
00091         acquisition_>set_channel(channel_);
00092     }
00093
00094     /*!
00095     * \brief Set channel fsm associated to this acquisition instance
00096     */
00097     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm) override
00098     {
00099         channel_fsm_ = std::move(channel_fsm);
00100         acquisition_>set_channel_fsm(channel_fsm_);

```

```

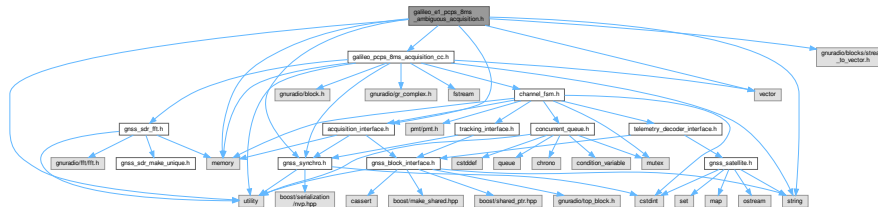
00101     }
00102
00103     /*!
00104     * \brief Set statistics threshold of PCPS algorithm
00105     */
00106     void set_threshold(float threshold) override;
00107
00108     /*!
00109     * \brief Set maximum Doppler off grid search
00110     */
00111     void set_doppler_max(unsigned int doppler_max) override;
00112
00113     /*!
00114     * \brief Set Doppler steps for the grid search
00115     */
00116     void set_doppler_step(unsigned int doppler_step) override;
00117
00118     /*!
00119     * \brief Initializes acquisition algorithm.
00120     */
00121     void init() override;
00122
00123     /*!
00124     * \brief Sets local code for GPS L1/CA PCPS acquisition algorithm.
00125     */
00126     void set_local_code() override;
00127
00128     /*!
00129     * \brief Returns the maximum peak of grid search
00130     */
00131     signed int mag() override;
00132
00133     /*!
00134     * \brief Restart acquisition algorithm
00135     */
00136     void reset() override;
00137
00138     /*!
00139     * \brief If state = 1, it forces the block to start acquiring from the first sample
00140     */
00141     void set_state(int state) override;
00142
00143     /*!
00144     * \brief Stop running acquisition
00145     */
00146     void stop_acquisition() override;
00147
00148     /*!
00149     * \brief Sets the resampler latency to account it in the acquisition code delay estimation
00150     */
00151     void set_resampler_latency(uint32_t latency_samples) override;
00152
00153 private:
00154     pcps_acquisition_sptr acquisition_;
00155     volk_gnssdr::vector<std::complex<float>> code_;
00156     std::weak_ptr<ChannelFsm> channel_fsm_;
00157     gr::blocks::float_to_complex::sptr float_to_complex_;
00158     complex_byte_to_float_x2_sptr cbyte_to_float_x2_;
00159     Gnss_Synchro* gnss_synchro_;
00160     Acq_Conf acq_parameters_;
00161     std::string item_type_;
00162     std::string role_;
00163     std::string dump_filename_;
00164     size_t item_size_;
00165     int64_t fs_in_;
00166     float threshold_;
00167     unsigned int vector_length_;
00168     unsigned int code_length_;
00169     unsigned int channel_;
00170     unsigned int doppler_max_;
00171     unsigned int doppler_step_;
00172     unsigned int num_codes_;
00173     unsigned int in_streams_;
00174     unsigned int out_streams_;
00175 };
00176
00177
00178 /** \} */
00179 /** \} */
00180 #endif // GNSS_SDR_BEIDOU_B3I_PCPS_ACQUISITION_H

```

## 11.5 galileo\_e1\_pcps\_8ms\_ambiguous\_acquisition.h File Reference

Adapts a PCPS 8ms acquisition block to an [AcquisitionInterface](#) for Galileo E1 Signals.

```
#include "acquisition_interface.h"
#include "galileo_pcps_8ms_acquisition_cc.h"
#include "gnss_synchro.h"
#include <gnuradio/blocks/stream_to_vector.h>
#include <memory>
#include <string>
#include <utility>
#include <vector>
Include dependency graph for galileo_e1_pcps_8ms_ambiguous_acquisition.h:
```



## Classes

- class `GalileoE1Pcps8msAmbiguousAcquisition`

Adapts a PCPS 8ms acquisition block to an [AcquisitionInterface](#) for Galileo E1 Signals.

### 11.5.1 Detailed Description

Adapts a PCPS 8ms acquisition block to an [AcquisitionInterface](#) for Galileo E1 Signals.

**Author**

Marc Molina, 2013. marc.molina.pena(at)gmail.com

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [galileo\\_e1\\_pcps\\_8ms\\_ambiguous\\_acquisition.h](#).

### 11.6 galileo\_e1\_pcps\_8ms\_ambiguous\_acquisition.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file galileo_e1_pcps_8ms_ambiguous_acquisition.h
00003  * \brief Adapts a PCPS 8ms acquisition block to an
00004  * AcquisitionInterface for Galileo E1 Signals
00005  * \author Marc Molina, 2013. marc.molina.pena(at)gmail.com
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_GALILEO_E1_PCPS_8MS_AMBIGUOUS_ACQUISITION_H
00019 #define GNSS_SDR_GALILEO_E1_PCPS_8MS_AMBIGUOUS_ACQUISITION_H
00020
00021 #include "acquisition_interface.h"
00022 #include "galileo_pcps_8ms_acquisition_cc.h"
00023 #include "gnss_synchro.h"
00024 #include <gnuradio/blocks/stream_to_vector.h>
00025 #include <memory>
00026 #include <string>
00027 #include <utility>
00028 #include <vector>

```

```

00029
00030 /** \addtogroup Acquisition
00031 * \{ */
00032 /** \addtogroup Acq_adapters
00033 * \{ */
00034
00035
00036 class ConfigurationInterface;
00037
00038 /*!
00039 * \brief Adapts a PCPS 8ms acquisition block to an
00040 * AcquisitionInterface for Galileo E1 Signals
00041 */
00042 class GalileoE1Pcps8msAmbiguousAcquisition : public AcquisitionInterface
00043 {
00044 public:
00045     GalileoE1Pcps8msAmbiguousAcquisition(const ConfigurationInterface* configuration,
00046         const std::string& role,
00047         unsigned int in_streams,
00048         unsigned int out_streams);
00049
00050     ~GalileoE1Pcps8msAmbiguousAcquisition() = default;
00051
00052     inline std::string role() override
00053     {
00054         return role_;
00055     }
00056
00057     /*!
00058     * \brief Returns "Galileo_E1_PCPS_8ms_Ambiguous_Acquisition"
00059     */
00060     inline std::string implementation() override
00061     {
00062         return "Galileo_E1_PCPS_8ms_Ambiguous_Acquisition";
00063     }
00064
00065     inline size_t item_size() override
00066     {
00067         return item_size_;
00068     }
00069
00070     void connect(gr::top_block_sptr top_block) override;
00071     void disconnect(gr::top_block_sptr top_block) override;
00072     gr::basic_block_sptr get_left_block() override;
00073     gr::basic_block_sptr get_right_block() override;
00074
00075     /*!
00076     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00077     * to efficiently exchange synchronization data between acquisition and
00078     * tracking blocks
00079     */
00080     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00081
00082     /*!
00083     * \brief Set acquisition channel unique ID
00084     */
00085     inline void set_channel(unsigned int channel) override
00086     {
00087         channel_ = channel;
00088         acquisition_cc->set_channel(channel_);
00089     }
00090
00091     /*!
00092     * \brief Set channel fsm associated to this acquisition instance
00093     */
00094     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm) override
00095     {
00096         channel_fsm_ = std::move(channel_fsm);
00097         acquisition_cc->set_channel_fsm(channel_fsm_);
00098     }
00099
00100     /*!
00101     * \brief Set statistics threshold of PCPS algorithm
00102     */
00103     void set_threshold(float threshold) override;
00104
00105     /*!
00106     * \brief Set maximum Doppler off grid search
00107     */
00108     void set_doppler_max(unsigned int doppler_max) override;
00109
00110     /*!
00111     * \brief Set Doppler steps for the grid search
00112     */
00113     void set_doppler_step(unsigned int doppler_step) override;
00114
00115     /*!

```



```

00116     * \brief Initializes acquisition algorithm.
00117     */
00118     void init() override;
00119
00120     /*!
00121     * \brief Sets local code for Galileo E1 PCPS acquisition algorithm.
00122     */
00123     void set_local_code() override;
00124
00125     /*!
00126     * \brief Returns the maximum peak of grid search
00127     */
00128     signed int mag() override;
00129
00130     /*!
00131     * \brief Restart acquisition algorithm
00132     */
00133     void reset() override;
00134
00135     /*!
00136     * \brief Stop running acquisition
00137     */
00138     void stop_acquisition() override;
00139
00140     void set_state(int state __attribute__((unused))) override {};
00141
00142     void set_resampler_latency(uint32_t latency_samples __attribute__((unused))) override {};
00143
00144 private:
00145     float calculate_threshold(float pfa) const;
00146
00147     const ConfigurationInterface* configuration_;
00148     galileo_pcps_8ms_acquisition_cc_sptr acquisition_cc_;
00149     gr::blocks::stream_to_vector::sptr stream_to_vector_;
00150     std::weak_ptr<ChannelFsm> channel_fsm_;
00151     std::vector<std::complex<float>> code_;
00152     Gnss_Synchro* gnss_synchro_;
00153     std::string item_type_;
00154     std::string dump_filename_;
00155     std::string role_;
00156     int64_t fs_in_;
00157     size_t item_size_;
00158     float threshold_;
00159     unsigned int vector_length_;
00160     unsigned int code_length_;
00161     unsigned int channel_;
00162     unsigned int doppler_max_;
00163     unsigned int doppler_step_;
00164     unsigned int sampled_ms_;
00165     unsigned int max_dwells_;
00166     unsigned int in_streams_;
00167     unsigned int out_streams_;
00168     bool dump_;
00169 };
00170
00171
00172
00173 /** @} */
00174 /** @} */
00175 #endif // GNSS_SDR_GALILEO_E1_PCPS_8MS_AMBIGUOUS_ACQUISITION_H

```

## 11.7 galileo\_e1\_pcps\_ambiguous\_acquisition.h File Reference

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Galileo E1 Signals.

```

#include "acq_conf.h"
#include "channel_fsm.h"
#include "complex_byte_to_float_x2.h"
#include "gnss_synchro.h"
#include "pcps_acquisition.h"
#include <gnuradio/blocks/float_to_complex.h>
#include <volk_gnssssdr/volk_gnssssdr_alloc.h>
#include <memory>
#include <string>
#include <utility>

```

Include dependency graph for `galileo_e1_pcps_ambiguous_acquisition.h`:



## Classes

- class [GalileoE1PcpsAmbiguousAcquisition](#)

*This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Galileo E1 Signals.*

### 11.7.1 Detailed Description

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Galileo E1 Signals.

#### Author

Luis Esteve, 2012. [luis\(at\)epsilon-formation.com](mailto:luis(at)epsilon-formation.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [galileo\\_e1\\_pcps\\_ambiguous\\_acquisition.h](#).

---

## 11.8 galileo\_e1\_pcps\_ambiguous\_acquisition.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file galileo_e1_pcps_ambiguous_acquisition.h
00003  * \brief Adapts a PCPS acquisition block to an AcquisitionInterface for
00004  * Galileo E1 Signals
00005  * \author Luis Esteve, 2012. luis(at)epsilon-formation.com
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_GALILEO_E1_PCPS_AMBIGUOUS_ACQUISITION_H
00019 #define GNSS_SDR_GALILEO_E1_PCPS_AMBIGUOUS_ACQUISITION_H
00020
00021 #include "acq_conf.h"
00022 #include "channel_fsm.h"
00023 #include "complex_byte_to_float_x2.h"
00024 #include "gnss_synchro.h"
00025 #include "pcps_acquisition.h"
00026 #include <gnuradio/blocks/float_to_complex.h>
00027 #include <volk_gnssdr/volk_gnssdr_alloc.h>
00028 #include <memory>
00029 #include <string>
00030 #include <utility>
00031
00032 /** \addtogroup Acquisition
00033  * \{ */
00034 /** \addtogroup Acq_adapters
00035  * \{ */
00036
00037
00038 class ConfigurationInterface;
00039
00040 /*!
00041  * \brief This class adapts a PCPS acquisition block to an
00042  * AcquisitionInterface for Galileo E1 Signals
00043  */
00044 class GalileoE1PcpsAmbiguousAcquisition : public AcquisitionInterface
00045 {

```

```

00046 public:
00047     GalileoE1PcpsAmbiguousAcquisition(
00048         const ConfigurationInterface* configuration,
00049         const std::string& role,
00050         unsigned int in_streams,
00051         unsigned int out_streams);
00052
00053     ~GalileoE1PcpsAmbiguousAcquisition() = default;
00054
00055     inline std::string role() override
00056     {
00057         return role_;
00058     }
00059
00060     /*!
00061      * \brief Returns "Galileo_E1_PCPS_Ambiguous_Acquisition"
00062      */
00063     inline std::string implementation() override
00064     {
00065         return "Galileo_E1_PCPS_Ambiguous_Acquisition";
00066     }
00067
00068     size_t item_size() override
00069     {
00070         return item_size_;
00071     }
00072
00073     void connect(gr::top_block_sptr top_block) override;
00074     void disconnect(gr::top_block_sptr top_block) override;
00075     gr::basic_block_sptr get_left_block() override;
00076     gr::basic_block_sptr get_right_block() override;
00077
00078     /*!
00079      * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00080      * to efficiently exchange synchronization data between acquisition and
00081      * tracking blocks
00082      */
00083     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00084
00085     /*!
00086      * \brief Set acquisition channel unique ID
00087      */
00088     inline void set_channel(unsigned int channel) override
00089     {
00090         channel_ = channel;
00091         acquisition_->set_channel(channel_);
00092     }
00093
00094     /*!
00095      * \brief Set channel fsm associated to this acquisition instance
00096      */
00097     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm) override
00098     {
00099         channel_fsm_ = std::move(channel_fsm);
00100         acquisition_->set_channel_fsm(channel_fsm_);
00101     }
00102
00103     /*!
00104      * \brief Set statistics threshold of PCPS algorithm
00105      */
00106     void set_threshold(float threshold) override;
00107
00108     /*!
00109      * \brief Set maximum Doppler off grid search
00110      */
00111     void set_doppler_max(unsigned int doppler_max) override;
00112
00113     /*!
00114      * \brief Set Doppler steps for the grid search
00115      */
00116     void set_doppler_step(unsigned int doppler_step) override;
00117
00118     /*!
00119      * \brief Set Doppler center for the grid search
00120      */
00121     void set_doppler_center(int doppler_center) override;
00122
00123     /*!
00124      * \brief Initializes acquisition algorithm.
00125      */
00126     void init() override;
00127
00128     /*!
00129      * \brief Sets local code for Galileo E1 PCPS acquisition algorithm.
00130      */
00131     void set_local_code() override;
00132

```

```

00133     /*!
00134     * \brief Returns the maximum peak of grid search
00135     */
00136     signed int mag() override;
00137
00138     /*!
00139     * \brief Restart acquisition algorithm
00140     */
00141     void reset() override;
00142
00143     /*!
00144     * \brief If state = 1, it forces the block to start acquiring from the first sample
00145     */
00146     void set_state(int state) override;
00147
00148     /*!
00149     * \brief Stop running acquisition
00150     */
00151     void stop_acquisition() override;
00152
00153     /*!
00154     * \brief Sets the resampler latency to account it in the acquisition code delay estimation
00155     */
00156     void set_resampler_latency(uint32_t latency_samples) override;
00157
00158 private:
00159     pcps_acquisition_sptr acquisition_;
00160     volk_gnssssdr::vector<std::complex<float>> code_;
00161     std::weak_ptr<ChannelFsm> channel_fsm_;
00162     gr::blocks::float_to_complex::sptr float_to_complex_;
00163     complex_byte_to_float_x2_sptr cbyte_to_float_x2_;
00164     Gnss_Synchro* gnss_synchro_;
00165     const ConfigurationInterface* configuration_;
00166     Acq_Conf acq_parameters_;
00167     std::string item_type_;
00168     std::string dump_filename_;
00169     std::string role_;
00170     int64_t fs_in_;
00171     size_t item_size_;
00172     float threshold_;
00173     int doppler_center_;
00174     unsigned int vector_length_;
00175     unsigned int code_length_;
00176     unsigned int channel_;
00177     unsigned int doppler_max_;
00178     unsigned int doppler_step_;
00179     unsigned int sampled_ms_;
00180     unsigned int in_streams_;
00181     unsigned int out_streams_;
00182     bool acquire_pilot_;
00183 };
00184
00185
00186 /** @} */
00187 /** @} */
00188 #endif // GNSS_SDR_GALILEO_E1_PCPS_AMBIGUOUS_ACQUISITION_H

```

## 11.9 galileo\_e1\_pcps\_ambiguous\_acquisition\_fpga.h File Reference

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Galileo E1 Signals for the FPGA.

```

#include "acq_conf_fpga.h"
#include "channel_fsm.h"
#include "gnss_synchro.h"
#include "pcps_acquisition_fpga.h"
#include <volk_gnssssdr/volk_gnssssdr_alloc.h>
#include <memory>
#include <string>
#include <utility>
#include <vector>

```



```

00044 {
00045 public:
00046     /*!
00047      * \brief Constructor
00048      */
00049     GalileoElPcpsAmbiguousAcquisitionFpga(
00050         const ConfigurationInterface* configuration,
00051         const std::string& role,
00052         unsigned int in_streams,
00053         unsigned int out_streams);
00054
00055     /*!
00056      * \brief Destructor
00057      */
00058     ~GalileoElPcpsAmbiguousAcquisitionFpga() = default;
00059
00060     /*!
00061      * \brief Role
00062      */
00063     inline std::string role() override
00064     {
00065         return role_;
00066     }
00067
00068     /*!
00069      * \brief Returns "Galileo_El_PCPS_Ambiguous_Acquisition_FPGA"
00070      */
00071     inline std::string implementation() override
00072     {
00073         return "Galileo_El_PCPS_Ambiguous_Acquisition_FPGA";
00074     }
00075
00076     /*!
00077      * \brief Returns size of lv_16sc_t
00078      */
00079     size_t item_size() override
00080     {
00081         return sizeof(int16_t);
00082     }
00083
00084     /*!
00085      * \brief Connect
00086      */
00087     void connect(gr::top_block_sptr top_block) override;
00088
00089     /*!
00090      * \brief Disconnect
00091      */
00092     void disconnect(gr::top_block_sptr top_block) override;
00093
00094     /*!
00095      * \brief Get left block
00096      */
00097     gr::basic_block_sptr get_left_block() override;
00098
00099     /*!
00100      * \brief Get right block
00101      */
00102     gr::basic_block_sptr get_right_block() override;
00103
00104     /*!
00105      * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00106      * to efficiently exchange synchronization data between acquisition and
00107      * tracking blocks
00108      */
00109     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00110
00111     /*!
00112      * \brief Set acquisition channel unique ID
00113      */
00114     inline void set_channel(unsigned int channel) override
00115     {
00116         channel_ = channel;
00117         acquisition_fpga_>set_channel(channel_);
00118     }
00119
00120     /*!
00121      * \brief Set channel fsm associated to this acquisition instance
00122      */
00123     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm) override
00124     {
00125         channel_fsm_ = std::move(channel_fsm);
00126         acquisition_fpga_>set_channel_fsm(channel_fsm_);
00127     }
00128
00129     /*!
00130      * \brief Set statistics threshold of PCPS algorithm

```

```

00131     */
00132 void set_threshold(float threshold) override;
00133
00134     /*!
00135     * \brief Set maximum Doppler off grid search
00136     */
00137 void set_doppler_max(unsigned int doppler_max) override;
00138
00139     /*!
00140     * \brief Set Doppler steps for the grid search
00141     */
00142 void set_doppler_step(unsigned int doppler_step) override;
00143
00144     /*!
00145     * \brief Set Doppler center for the grid search
00146     */
00147 void set_doppler_center(int doppler_center) override;
00148
00149     /*!
00150     * \brief Initializes acquisition algorithm.
00151     */
00152 void init() override;
00153
00154     /*!
00155     * \brief Sets local code for Galileo E1 PCPS acquisition algorithm.
00156     */
00157 void set_local_code() override;
00158
00159     /*!
00160     * \brief Returns the maximum peak of grid search
00161     */
00162 signed int mag() override;
00163
00164     /*!
00165     * \brief Restart acquisition algorithm
00166     */
00167 void reset() override;
00168
00169     /*!
00170     * \brief If state = 1, it forces the block to start acquiring from the first sample
00171     */
00172 void set_state(int state) override;
00173
00174     /*!
00175     * \brief Stop running acquisition
00176     */
00177 void stop_acquisition() override;
00178
00179     /*!
00180     * \brief Set resampler latency
00181     */
00182 void set_resampler_latency(uint32_t latency_samples __attribute__((unused))) override {};
00183
00184 private:
00185     static const uint32_t ACQ_BUFF_0 = 0; // FPGA Acquisition IP buffer containing L1/E1
00186     frequency band samples by default.
00187     static const uint32_t DEFAULT_FPGA_BLK_EXP = 13; // default block exponent
00188     static const uint32_t QUANT_BITS_LOCAL_CODE = 16;
00189     static const uint32_t SELECT_LSBITS = 0x0000FFFF; // Select the 10 LSBits out of a 20-bit
00190     word
00191     static const uint32_t SELECT_MSBITS = 0xFFFF0000; // Select the 10 MSbits out of a 20-bit
00192     word
00193     static const uint32_t SELECT_ALL_CODE_BITS = 0xFFFFFFFF; // Select a 20 bit word
00194     static const uint32_t SHL_CODE_BITS = 65536; // shift left by 10 bits
00195
00196 void generate_galileo_e1_prn_codes();
00197
00198 pcps_acquisition_fpga_sptr acquisition_fpga_;
00199 volk_gnssdr::vector<uint32_t> d_all_fft_codes; // memory that contains all the code ffts
00200 std::weak_ptr<ChannelFsm> channel_fsm_;
00201 Gnss_Synchro* gnss_synchro_;
00202 Acq_Conf_Fpga acq_parameters_;
00203 std::string role_;
00204 int32_t doppler_center_;
00205 uint32_t channel_;
00206 uint32_t doppler_max_;
00207 uint32_t doppler_step_;
00208 unsigned int in_streams_;
00209 unsigned int out_streams_;
00210 bool acquire_pilot_;
00211 };
00212
00213 /** \} */
00214 /** \} */
00215 #endif // GNSS_SDR_GALILEO_E1_PCPS_AMBIGUOUS_ACQUISITION_FPGA_H

```





```

00026 #include <string>
00027 #include <utility>
00028 #include <vector>
00029
00030 /** \addtogroup Acquisition
00031  * \{ */
00032 /** \addtogroup Acq_adapters
00033  * \{ */
00034
00035
00036 class ConfigurationInterface;
00037
00038 /*!
00039  * \brief Adapts a PCPS CCCWSR acquisition block to an AcquisitionInterface
00040  * for Galileo E1 Signals
00041  */
00042 class GalileoE1PcpsCccwsrAmbiguousAcquisition : public AcquisitionInterface
00043 {
00044 public:
00045     GalileoE1PcpsCccwsrAmbiguousAcquisition(
00046         const ConfigurationInterface* configuration,
00047         const std::string& role,
00048         unsigned int in_streams,
00049         unsigned int out_streams);
00050
00051     ~GalileoE1PcpsCccwsrAmbiguousAcquisition() = default;
00052
00053     inline std::string role() override
00054     {
00055         return role_;
00056     }
00057
00058     /*!
00059     * \brief Returns "Galileo_E1_PCPS_CCCWSR_Ambiguous_Acquisition"
00060     */
00061     inline std::string implementation() override
00062     {
00063         return "Galileo_E1_PCPS_CCCWSR_Ambiguous_Acquisition";
00064     }
00065
00066     inline size_t item_size() override
00067     {
00068         return item_size_;
00069     }
00070
00071     void connect(gr::top_block_sptr top_block) override;
00072     void disconnect(gr::top_block_sptr top_block) override;
00073     gr::basic_block_sptr get_left_block() override;
00074     gr::basic_block_sptr get_right_block() override;
00075
00076     /*!
00077     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00078     * to efficiently exchange synchronization data between acquisition and
00079     * tracking blocks
00080     */
00081     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00082
00083     /*!
00084     * \brief Set acquisition channel unique ID
00085     */
00086     inline void set_channel(unsigned int channel) override
00087     {
00088         channel_ = channel;
00089         acquisition_cc->set_channel(channel_);
00090     }
00091
00092     /*!
00093     * \brief Set channel fsm associated to this acquisition instance
00094     */
00095     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm) override
00096     {
00097         channel_fsm_ = std::move(channel_fsm);
00098         acquisition_cc->set_channel_fsm(channel_fsm_);
00099     }
00100
00101     /*!
00102     * \brief Set statistics threshold of CCCWSR algorithm
00103     */
00104     void set_threshold(float threshold) override;
00105
00106     /*!
00107     * \brief Set maximum Doppler off grid search
00108     */
00109     void set_doppler_max(unsigned int doppler_max) override;
00110
00111     /*!
00112     * \brief Set Doppler steps for the grid search

```

```

00113     */
00114 void set_doppler_step(unsigned int doppler_step) override;
00115
00116     /*!
00117     * \brief Initializes acquisition algorithm.
00118     */
00119 void init() override;
00120
00121 void set_local_code() override;
00122
00123     /*!
00124     * \brief Returns the maximum peak of grid search
00125     */
00126 signed int mag() override;
00127
00128     /*!
00129     * \brief Restart acquisition algorithm
00130     */
00131 void reset() override;
00132
00133     /*!
00134     * \brief If state = 1, it forces the block to start acquiring from the first sample
00135     */
00136 void set_state(int state) override;
00137
00138     /*!
00139     * \brief Stop running acquisition
00140     */
00141 void stop_acquisition() override;
00142
00143 void set_resampler_latency(uint32_t latency_samples __attribute__((unused))) override {};
00144
00145 private:
00146 float calculate_threshold(float pfa);
00147
00148 const ConfigurationInterface* configuration_;
00149 pcps_cccwsr_acquisition_cc_sptr acquisition_cc_;
00150 gr::blocks::stream_to_vector::sptr stream_to_vector_;
00151 std::weak_ptr<ChannelFsm> channel_fsm_;
00152 std::vector<std::complex<float>> code_data_;
00153 std::vector<std::complex<float>> code_pilot_;
00154 std::string item_type_;
00155 std::string dump_filename_;
00156 std::string role_;
00157 Gnss_Synchro* gnss_synchro_;
00158 int64_t fs_in_;
00159 size_t item_size_;
00160 float threshold_;
00161 unsigned int vector_length_;
00162 unsigned int code_length_;
00163 unsigned int channel_;
00164 unsigned int doppler_max_;
00165 unsigned int doppler_step_;
00166 unsigned int sampled_ms_;
00167 unsigned int max_dwells_;
00168 unsigned int in_streams_;
00169 unsigned int out_streams_;
00170 bool dump_;
00171 };
00172
00173
00174 /** \} */
00175 /** \} */
00176 #endif // GNSS_SDR_GALILEO_E1_PCPS_CCCWSR_AMBIGUOUS_ACQUISITION_H

```

## 11.13 galileo\_e1\_pcps\_quicksync\_ambiguous\_acquisition.h File Reference

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Galileo E1 Signals.

```

#include "channel_fsm.h"
#include "gnss_synchro.h"
#include "pcps_quicksync_acquisition_cc.h"
#include <gnuradio/blocks/stream_to_vector.h>
#include <memory>
#include <string>
#include <utility>
#include <vector>

```

[illegible]

- class GalileoE1PcpsQuickSyncAmbiguousAcquisition

### 11.13.1 Detailed Description

June, 2014

Damian Miralles Sanchez. [dmiralles2009@gmail.com](mailto:dmiralles2009@gmail.com)

11.14 galileo\_e1\_pcps\_quicksync\_ambiguous\_acquisition.h

Generated by Doxygen

```

00037
00038 /*!
00039  * \brief This class adapts a PCPS acquisition block to an
00040  * AcquisitionInterface for Galileo E1 Signals
00041  */
00042 class GalileoE1PcpsQuickSyncAmbiguousAcquisition : public AcquisitionInterface
00043 {
00044 public:
00045     GalileoE1PcpsQuickSyncAmbiguousAcquisition(
00046         const ConfigurationInterface* configuration,
00047         const std::string& role,
00048         unsigned int in_streams,
00049         unsigned int out_streams);
00050
00051     ~GalileoE1PcpsQuickSyncAmbiguousAcquisition() = default;
00052
00053     inline std::string role() override
00054     {
00055         return role_;
00056     }
00057
00058 /*!
00059  * \brief Returns "Galileo_E1_PCPS_Ambiguous_Acquisition"
00060  */
00061     inline std::string implementation() override
00062     {
00063         return "Galileo_E1_PCPS_QuickSync_Ambiguous_Acquisition";
00064     }
00065
00066     inline size_t item_size() override
00067     {
00068         return item_size_;
00069     }
00070
00071     void connect(gr::top_block_sptr top_block) override;
00072     void disconnect(gr::top_block_sptr top_block) override;
00073     gr::basic_block_sptr get_left_block() override;
00074     gr::basic_block_sptr get_right_block() override;
00075
00076 /*!
00077  * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00078  * to efficiently exchange synchronization data between acquisition and
00079  * tracking blocks
00080  */
00081     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00082
00083 /*!
00084  * \brief Set acquisition channel unique ID
00085  */
00086     inline void set_channel(unsigned int channel) override
00087     {
00088         channel_ = channel;
00089         acquisition_cc->set_channel(channel_);
00090     }
00091
00092 /*!
00093  * \brief Set channel fsm associated to this acquisition instance
00094  */
00095     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm) override
00096     {
00097         channel_fsm_ = std::move(channel_fsm);
00098         acquisition_cc->set_channel_fsm(channel_fsm_);
00099     }
00100
00101 /*!
00102  * \brief Set statistics threshold of PCPS algorithm
00103  */
00104     void set_threshold(float threshold) override;
00105
00106 /*!
00107  * \brief Set maximum Doppler off grid search
00108  */
00109     void set_doppler_max(unsigned int doppler_max) override;
00110
00111 /*!
00112  * \brief Set Doppler steps for the grid search
00113  */
00114     void set_doppler_step(unsigned int doppler_step) override;
00115
00116 /*!
00117  * \brief Initializes acquisition algorithm.
00118  */
00119     void init() override;
00120
00121 /*!
00122  * \brief Sets local code for Galileo E1 PCPS acquisition algorithm.
00123  */

```

```

00124     void set_local_code() override;
00125
00126     /*!
00127      * \brief Returns the maximum peak of grid search
00128      */
00129     signed int mag() override;
00130
00131     /*!
00132      * \brief Restart acquisition algorithm
00133      */
00134     void reset() override;
00135
00136     /*!
00137      * \brief If state = 1, it forces the block to start acquiring from the first sample
00138      */
00139     void set_state(int state) override;
00140
00141     /*!
00142      * \brief Stop running acquisition
00143      */
00144     void stop_acquisition() override;
00145
00146     void set_resampler_latency(uint32_t latency_samples __attribute__((unused))) override {};
00147
00148 private:
00149     float calculate_threshold(float pfa) const;
00150
00151     const ConfigurationInterface* configuration_;
00152     pcps_quicksync_acquisition_cc_sptr acquisition_cc_;
00153     gr::blocks::stream_to_vector::sptr stream_to_vector_;
00154     std::weak_ptr<ChannelFsm> channel_fsm_;
00155     std::vector<std::complex<float>> code_;
00156     std::string item_type_;
00157     std::string role_;
00158     std::string dump_filename_;
00159     Gnss_Synchro* gnss_synchro_;
00160     int64_t fs_in_;
00161     size_t item_size_;
00162     float threshold_;
00163     unsigned int vector_length_;
00164     unsigned int code_length_;
00165     unsigned int channel_;
00166     unsigned int doppler_max_;
00167     unsigned int doppler_step_;
00168     unsigned int sampled_ms_;
00169     unsigned int max_dwells_;
00170     unsigned int folding_factor_;
00171     unsigned int in_streams_;
00172     unsigned int out_streams_;
00173     bool bit_transition_flag_;
00174     bool dump_;
00175 };
00176
00177
00178 /** @} */
00179 /** @} */
00180 #endif // GNSS_SDR_GALILEO_E1_PCPS_QUICKSYNC_AMBIGUOUS_ACQUISITION_H

```

## 11.15 galileo\_e1\_pcps\_tong\_ambiguous\_acquisition.h File Reference

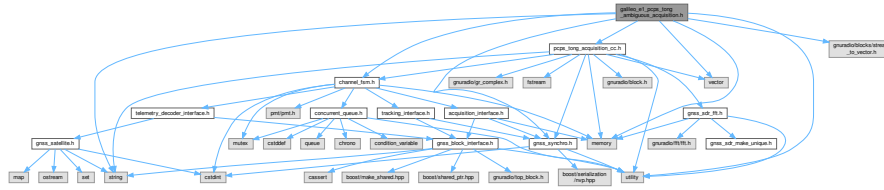
Adapts a PCPS Tong acquisition block to an [AcquisitionInterface](#) for Galileo E1 Signals.

```

#include "channel_fsm.h"
#include "gnss_synchro.h"
#include "pcps_tong_acquisition_cc.h"
#include <gnuradio/blocks/stream_to_vector.h>
#include <memory>
#include <string>
#include <utility>
#include <vector>

```

Include dependency graph for `galileo_e1_pcps_tong_ambiguous_acquisition.h`:



## Classes

- class [GalileoE1PcpsTongAmbiguousAcquisition](#)  
Adapts a PCPS Tong acquisition block to an [AcquisitionInterface](#) for Galileo E1 Signals.

### 11.15.1 Detailed Description

Adapts a PCPS Tong acquisition block to an [AcquisitionInterface](#) for Galileo E1 Signals.

#### Author

Marc Molina, 2013. marc.molina.pena(at)gmail.com

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [galileo\\_e1\\_pcps\\_tong\\_ambiguous\\_acquisition.h](#).

---

## 11.16 galileo\_e1\_pcps\_tong\_ambiguous\_acquisition.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file galileo_e1_pcps_tong_ambiguous_acquisition.h
00003  * \brief Adapts a PCPS Tong acquisition block to an AcquisitionInterface for
00004  * Galileo E1 Signals
00005  * \author Marc Molina, 2013. marc.molina.pena(at)gmail.com
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_GALILEO_E1_PCPS_TONG_AMBIGUOUS_ACQUISITION_H
00019 #define GNSS_SDR_GALILEO_E1_PCPS_TONG_AMBIGUOUS_ACQUISITION_H
00020
00021 #include "channel_fsm.h"
00022 #include "gnss_synchro.h"
00023 #include "pcps_tong_acquisition_cc.h"
00024 #include <gnuradio/blocks/stream_to_vector.h>
00025 #include <memory>
00026 #include <string>
00027 #include <utility>
00028 #include <vector>
00029
00030 /** \addtogroup Acquisition
00031  * \{ */
00032 /** \addtogroup Acq_adapters
00033  * \{ */
00034
00035
00036 class ConfigurationInterface;
00037
00038 /*!
00039  * \brief Adapts a PCPS Tong acquisition block to an AcquisitionInterface
00040  * for Galileo E1 Signals
00041  */
00042 class GalileoE1PcpsTongAmbiguousAcquisition : public AcquisitionInterface
```

```

00043 {
00044 public:
00045     GalileoElPcpsTongAmbiguousAcquisition(
00046         const ConfigurationInterface* configuration,
00047         const std::string& role,
00048         unsigned int in_streams,
00049         unsigned int out_streams);
00050
00051     ~GalileoElPcpsTongAmbiguousAcquisition() = default;
00052
00053     inline std::string role() override
00054     {
00055         return role_;
00056     }
00057
00058     /*!
00059     * \brief Returns "Galileo_El_PCPS_Tong_Ambiguous_Acquisition"
00060     */
00061     inline std::string implementation() override
00062     {
00063         return "Galileo_El_PCPS_Tong_Ambiguous_Acquisition";
00064     }
00065
00066     inline size_t item_size() override
00067     {
00068         return item_size_;
00069     }
00070
00071     void connect(gr::top_block_sptr top_block) override;
00072     void disconnect(gr::top_block_sptr top_block) override;
00073     gr::basic_block_sptr get_left_block() override;
00074     gr::basic_block_sptr get_right_block() override;
00075
00076     /*!
00077     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00078     * to efficiently exchange synchronization data between acquisition and
00079     * tracking blocks
00080     */
00081     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00082
00083     /*!
00084     * \brief Set acquisition channel unique ID
00085     */
00086     inline void set_channel(unsigned int channel) override
00087     {
00088         channel_ = channel;
00089         acquisition_cc->set_channel(channel_);
00090     }
00091
00092     /*!
00093     * \brief Set channel fsm associated to this acquisition instance
00094     */
00095     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm) override
00096     {
00097         channel_fsm_ = std::move(channel_fsm);
00098         acquisition_cc->set_channel_fsm(channel_fsm_);
00099     }
00100
00101     /*!
00102     * \brief Set statistics threshold of TONG algorithm
00103     */
00104     void set_threshold(float threshold) override;
00105
00106     /*!
00107     * \brief Set maximum Doppler off grid search
00108     */
00109     void set_doppler_max(unsigned int doppler_max) override;
00110
00111     /*!
00112     * \brief Set Doppler steps for the grid search
00113     */
00114     void set_doppler_step(unsigned int doppler_step) override;
00115
00116     /*!
00117     * \brief Initializes acquisition algorithm.
00118     */
00119     void init() override;
00120
00121     /*!
00122     * \brief Sets local code for Galileo E1 TONG acquisition algorithm.
00123     */
00124     void set_local_code() override;
00125
00126     /*!
00127     * \brief Returns the maximum peak of grid search
00128     */
00129     signed int mag() override;

```

### 11.17 galileo\_e5a\_noncoherent\_iq\_acquisition\_caf.h File Reference

```
#include "channel_fsm.h"
#include "galileo_e5a_noncoherent_iq_acquisition_caf_cc.h"
#include "gnss_synchro.h"
#include <memory>
#include <string>
#include <utility>
#include <vector>
```

[illegible]



**Classes**

- class [GalileoE5aNoncoherentIQAcquisitionCaf](#)

**11.17.1 Detailed Description**

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Galileo E5a data and pilot Signals.

**Author**

Marc Sales, 2014. marcsales92(at)gmail.com \based on work from:

- Javier Arribas, 2011. jarribas(at)cttc.es
- Luis Esteve, 2012. luis(at)epsilon-formation.com
- Marc Molina, 2013. [marc.molina.pena@gmail.com](mailto:marc.molina.pena@gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [galileo\\_e5a\\_noncoherent\\_iq\\_acquisition\\_caf.h](#).

---

**11.18 galileo\_e5a\_noncoherent\_iq\_acquisition\_caf.h**

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file galileo_e5a_noncoherent_iq_acquisition_caf.h
00003  * \brief Adapts a PCPS acquisition block to an AcquisitionInterface for
00004  * \brief Galileo E5a data and pilot Signals
00005  * \author Marc Sales, 2014. marcsales92(at)gmail.com
00006  * \based on work from:
00007  *      <ul>
00008  *          <li> Javier Arribas, 2011. jarribas(at)cttc.es
00009  *          <li> Luis Esteve, 2012. luis(at)epsilon-formation.com
00010  *          <li> Marc Molina, 2013. marc.molina.pena@gmail.com
00011  *      </ul>
00012  *
00013  * -----
00014  *
00015  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00016  * This file is part of GNSS-SDR.
00017  *
00018  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00019  * SPDX-License-Identifier: GPL-3.0-or-later
00020  *
00021  * -----
00022  */
00023
00024 #ifndef GNSS_SDR_GALILEO_E5A_NONCOHERENT_IQ_ACQUISITION_CAF_H
00025 #define GNSS_SDR_GALILEO_E5A_NONCOHERENT_IQ_ACQUISITION_CAF_H
00026
00027 #include "channel_fsm.h"
00028 #include "galileo_e5a_noncoherent_iq_acquisition_caf_cc.h"
00029 #include "gnss_synchro.h"
00030 #include <memory>
00031 #include <string>
00032 #include <utility>
00033 #include <vector>
00034
00035 /** \addtogroup Acquisition
00036  * \{ */
00037 /** \addtogroup Acq_adapters
00038  * \{ */
00039
00040
00041 class ConfigurationInterface;
00042
00043 class GalileoE5aNoncoherentIQAcquisitionCaf : public AcquisitionInterface
00044 {
00045 public:
00046     GalileoE5aNoncoherentIQAcquisitionCaf(const ConfigurationInterface* configuration,
00047         const std::string& role,
00048         unsigned int in_streams,
00049         unsigned int out_streams);
00050
00051     ~GalileoE5aNoncoherentIQAcquisitionCaf() = default;
00052
00053     inline std::string role() override
00054     {

```

```

00055         return role_;
00056     }
00057
00058     /*!
00059     * \brief Returns "Galileo_E5a_Noncoherent_IQ_Acquisition_CAF"
00060     */
00061     inline std::string implementation() override
00062     {
00063         return "Galileo_E5a_Noncoherent_IQ_Acquisition_CAF";
00064     }
00065
00066     inline size_t item_size() override
00067     {
00068         return item_size_;
00069     }
00070
00071     void connect(gr::top_block_sptr top_block) override;
00072     void disconnect(gr::top_block_sptr top_block) override;
00073     gr::basic_block_sptr get_left_block() override;
00074     gr::basic_block_sptr get_right_block() override;
00075
00076     /*!
00077     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00078     * to efficiently exchange synchronization data between acquisition and
00079     * tracking blocks
00080     */
00081     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00082
00083     /*!
00084     * \brief Set acquisition channel unique ID
00085     */
00086     inline void set_channel(unsigned int channel) override
00087     {
00088         channel_ = channel;
00089         acquisition_cc->set_channel(channel_);
00090     }
00091
00092     /*!
00093     * \brief Set channel fsm associated to this acquisition instance
00094     */
00095     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm) override
00096     {
00097         channel_fsm_ = std::move(channel_fsm);
00098         acquisition_cc->set_channel_fsm(channel_fsm_);
00099     }
00100
00101     /*!
00102     * \brief Set statistics threshold of PCPS algorithm
00103     */
00104     void set_threshold(float threshold) override;
00105
00106     /*!
00107     * \brief Set maximum Doppler off grid search
00108     */
00109     void set_doppler_max(unsigned int doppler_max) override;
00110
00111     /*!
00112     * \brief Set Doppler steps for the grid search
00113     */
00114     void set_doppler_step(unsigned int doppler_step) override;
00115
00116     /*!
00117     * \brief Initializes acquisition algorithm.
00118     */
00119     void init() override;
00120
00121     /*!
00122     * \brief Sets local Galileo E5a code for PCPS acquisition algorithm.
00123     */
00124     void set_local_code() override;
00125
00126     /*!
00127     * \brief Returns the maximum peak of grid search
00128     */
00129     signed int mag() override;
00130
00131     /*!
00132     * \brief Restart acquisition algorithm
00133     */
00134     void reset() override;
00135
00136     /*!
00137     * \brief If set to 1, ensures that acquisition starts at the
00138     * first available sample.
00139     * \param state - int=1 forces start of acquisition
00140     */
00141     void set_state(int state) override;

```

```

00142
00143     /*!
00144     * \brief Stop running acquisition
00145     */
00146     void stop_acquisition() override;
00147
00148     void set_resampler_latency(uint32_t latency_samples __attribute__((unused))) override {};
00149
00150 private:
00151     float calculate_threshold(float pfa) const;
00152
00153     const ConfigurationInterface* configuration_;
00154     galileo_e5a_noncoherentIQ_acquisition_caf_cc_sptr acquisition_cc_;
00155     std::weak_ptr<ChannelFsm> channel_fsm_;
00156     std::vector<std::complex<float>> codeI_;
00157     std::vector<std::complex<float>> codeQ_;
00158     std::string item_type_;
00159     std::string role_;
00160     std::string dump_filename_;
00161     Gnss_Synchro* gnss_synchro_;
00162     int64_t fs_in_;
00163     size_t item_size_;
00164     float threshold_;
00165     int Zero_padding;
00166     int CAF_window_hz_;
00167     int code_length_;
00168     unsigned int vector_length_;
00169     unsigned int channel_;
00170     unsigned int doppler_max_;
00171     unsigned int doppler_step_;
00172     unsigned int sampled_ms_;
00173     unsigned int max_dwells_;
00174     unsigned int in_streams_;
00175     unsigned int out_streams_;
00176     bool bit_transition_flag_;
00177     bool both_signal_components;
00178     bool dump_;
00179 };
00180
00181
00182 /** \} */
00183 /** \} */
00184 #endif // GNSS_SDR_GALILEO_E5A_NONCOHERENT_IQ_ACQUISITION_CAF_H

```

## 11.19 galileo\_e5a\_pcps\_acquisition.h File Reference

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Galileo E5a data and pilot Signals.

```

#include "channel_fsm.h"
#include "gnss_synchro.h"
#include "pcps_acquisition.h"
#include <volk_gnssssdr/volk_gnssssdr_alloc.h>
#include <memory>
#include <string>
#include <utility>

```

Include dependency graph for galileo\_e5a\_pcps\_acquisition.h:



### Classes

- class [GalileoE5aPcpsAcquisition](#)

### 11.19.1 Detailed Description

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Galileo E5a data and pilot Signals.

## Author

Antonio Ramos, 2018. antonio.ramos(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [galileo\\_e5a\\_pcps\\_acquisition.h](#).

---

## 11.20 galileo\_e5a\_pcps\_acquisition.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file galileo_e5a_pcps_acquisition.h
00003  * \brief Adapts a PCPS acquisition block to an AcquisitionInterface for
00004  * Galileo E5a data and pilot Signals
00005  * \author Antonio Ramos, 2018. antonio.ramos(at)cttc.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_GALILEO_E5A_PCPS_ACQUISITION_H
00019 #define GNSS_SDR_GALILEO_E5A_PCPS_ACQUISITION_H
00020
00021
00022 #include "channel_fsm.h"
00023 #include "gnss_synchro.h"
00024 #include "pcps_acquisition.h"
00025 #include <volk_gnssssdr/volk_gnssssdr_alloc.h>
00026 #include <memory>
00027 #include <string>
00028 #include <utility>
00029
00030 /** \addtogroup Acquisition
00031  * \{ */
00032 /** \addtogroup Acq_adapters
00033  * \{ */
00034
00035
00036 class ConfigurationInterface;
00037
00038 class GalileoE5aPcpsAcquisition : public AcquisitionInterface
00039 {
00040 public:
00041     GalileoE5aPcpsAcquisition(
00042         const ConfigurationInterface* configuration,
00043         const std::string& role,
00044         unsigned int in_streams,
00045         unsigned int out_streams);
00046
00047     ~GalileoE5aPcpsAcquisition() = default;
00048
00049     inline std::string role() override
00050     {
00051         return role_;
00052     }
00053
00054     inline std::string implementation() override
00055     {
00056         return "Galileo_E5a_Pcps_Acquisition";
00057     }
00058
00059     inline size_t item_size() override
00060     {
00061         return item_size_;
00062     }
00063
00064     void connect(gr::top_block_sptr top_block) override;
00065     void disconnect(gr::top_block_sptr top_block) override;
00066     gr::basic_block_sptr get_left_block() override;
00067     gr::basic_block_sptr get_right_block() override;
00068
00069     /*!
00070     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00071     * to efficiently exchange synchronization data between acquisition and
00072     * tracking blocks

```

```

00073     */
00074 void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00075
00076     /*!
00077     * \brief Set acquisition channel unique ID
00078     */
00079 inline void set_channel(unsigned int channel) override
00080 {
00081     channel_ = channel;
00082     acquisition_>set_channel(channel_);
00083 }
00084
00085     /*!
00086     * \brief Set channel fsm associated to this acquisition instance
00087     */
00088 inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm) override
00089 {
00090     channel_fsm_ = std::move(channel_fsm);
00091     acquisition_>set_channel_fsm(channel_fsm_);
00092 }
00093
00094     /*!
00095     * \brief Set statistics threshold of PCPS algorithm
00096     */
00097 void set_threshold(float threshold) override;
00098
00099     /*!
00100     * \brief Set maximum Doppler off grid search
00101     */
00102 void set_doppler_max(unsigned int doppler_max) override;
00103
00104     /*!
00105     * \brief Set Doppler steps for the grid search
00106     */
00107 void set_doppler_step(unsigned int doppler_step) override;
00108
00109     /*!
00110     * \brief Set Doppler center for the grid search
00111     */
00112 void set_doppler_center(int doppler_center) override;
00113
00114     /*!
00115     * \brief Initializes acquisition algorithm.
00116     */
00117 void init() override;
00118
00119     /*!
00120     * \brief Sets local Galileo E5a code for PCPS acquisition algorithm.
00121     */
00122 void set_local_code() override;
00123
00124     /*!
00125     * \brief Returns the maximum peak of grid search
00126     */
00127 signed int mag() override;
00128
00129     /*!
00130     * \brief Restart acquisition algorithm
00131     */
00132 void reset() override;
00133
00134     /*!
00135     * \brief If set to 1, ensures that acquisition starts at the
00136     * first available sample.
00137     * \param state - int=1 forces start of acquisition
00138     */
00139 void set_state(int state) override;
00140
00141     /*!
00142     * \brief Stop running acquisition
00143     */
00144 void stop_acquisition() override;
00145
00146     /*!
00147     * \brief Sets the resampler latency to account it in the acquisition code delay estimation
00148     */
00149 void set_resampler_latency(uint32_t latency_samples) override;
00150
00151 private:
00152     pcps_acquisition_sptr acquisition_;
00153     volk_gnss_sdr::vector<std::complex<float>> code_;
00154     std::weak_ptr<ChannelFsm> channel_fsm_;
00155     Gnss_Synchro* gnss_synchro_;
00156     Acq_Conf acq_parameters_;
00157     std::string item_type_;
00158     std::string dump_filename_;
00159     std::string role_;

```

```

00160     int64_t fs_in_;
00161     size_t item_size_;
00162     float threshold_;
00163     int doppler_center_;
00164     unsigned int vector_length_;
00165     unsigned int code_length_;
00166     unsigned int channel_;
00167     unsigned int doppler_max_;
00168     unsigned int doppler_step_;
00169     unsigned int sampled_ms_;
00170     unsigned int in_streams_;
00171     unsigned int out_streams_;
00172     bool acq_pilot_;
00173     bool acq_iq_;
00174 };
00175
00176
00177 /** \} */
00178 /** \} */
00179 #endif // GNSS_SDR_GALILEO_E5A_PCPS_ACQUISITION_H

```

## 11.21 galileo\_e5a\_pcps\_acquisition\_fpga.h File Reference

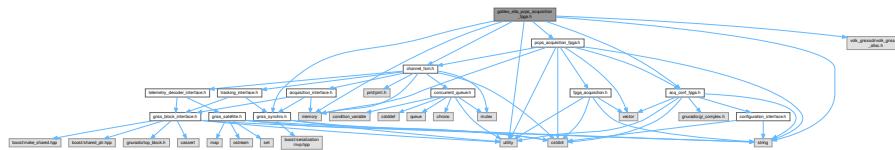
Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Galileo E5a data and pilot Signals for the FPGA.

```

#include "acq_conf_fpga.h"
#include "channel_fsm.h"
#include "gnss_synchro.h"
#include "pcps_acquisition_fpga.h"
#include <volk_gnssdr/volk_gnssdr_alloc.h>
#include <memory>
#include <string>
#include <utility>

```

Include dependency graph for galileo\_e5a\_pcps\_acquisition\_fpga.h:



### Classes

- class [GalileoE5aPcpsAcquisitionFpga](#)

*This class adapts a PCPS acquisition block off-loaded on an FPGA to an [AcquisitionInterface](#) for Galileo E5a signals.*

### 11.21.1 Detailed Description

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Galileo E5a data and pilot Signals for the FPGA.

#### Author

Marc Majoral, 2019. mmajoral(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [galileo\\_e5a\\_pcps\\_acquisition\\_fpga.h](#).

---

## 11.22 galileo\_e5a\_pcps\_acquisition\_fpga.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002 * \file galileo_e5a_pcps_acquisition_fpga.h
00003 * \brief Adapts a PCPS acquisition block to an AcquisitionInterface for
00004 * Galileo E5a data and pilot Signals for the FPGA

```

```

00005  * \author Marc Majoral, 2019. mmajoral(at)cttc.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_GALILEO_E5A_PCPS_ACQUISITION_FPGA_H
00019 #define GNSS_SDR_GALILEO_E5A_PCPS_ACQUISITION_FPGA_H
00020
00021 #include "acq_conf_fpga.h"
00022 #include "channel_fsm.h"
00023 #include "gnss_synchro.h"
00024 #include "pcps_acquisition_fpga.h"
00025 #include <volk_gnssdr/volk_gnssdr_alloc.h>
00026 #include <memory>
00027 #include <string>
00028 #include <utility>
00029
00030 /** \addtogroup Acquisition
00031  * \{ */
00032 /** \addtogroup Acq_adapters
00033  * \{ */
00034
00035
00036 class ConfigurationInterface;
00037
00038
00039 /*!
00040  * \brief This class adapts a PCPS acquisition block off-loaded on an FPGA
00041  * to an AcquisitionInterface for Galileo E5a signals
00042  */
00043 class GalileoE5aPcpsAcquisitionFpga : public AcquisitionInterface
00044 {
00045 public:
00046     /*!
00047      * \brief Constructor
00048      */
00049     GalileoE5aPcpsAcquisitionFpga(
00050         const ConfigurationInterface* configuration,
00051         const std::string& role,
00052         unsigned int in_streams,
00053         unsigned int out_streams);
00054
00055     /*!
00056      * \brief Destructor
00057      */
00058     ~GalileoE5aPcpsAcquisitionFpga() = default;
00059
00060     /*!
00061      * \brief Role
00062      */
00063     inline std::string role() override
00064     {
00065         return role_;
00066     }
00067
00068     /*!
00069      * \brief Returns "Galileo_E5a_Pcps_Acquisition_FPGA"
00070      */
00071     inline std::string implementation() override
00072     {
00073         return "Galileo_E5a_Pcps_Acquisition_FPGA";
00074     }
00075
00076     /*!
00077      * \brief Returns size of lv_16sc_t
00078      */
00079     inline size_t item_size() override
00080     {
00081         return sizeof(int16_t);
00082     }
00083
00084     /*!
00085      * \brief Connect
00086      */
00087     void connect(gr::top_block_sptr top_block) override;
00088
00089     /*!
00090      * \brief Disconnect
00091      */

```

```

00092 void disconnect(gr::top_block_sptr top_block) override;
00093
00094 /*!
00095  * \brief Get left block
00096  */
00097 gr::basic_block_sptr get_left_block() override;
00098
00099 /*!
00100  * \brief Get right block
00101  */
00102 gr::basic_block_sptr get_right_block() override;
00103
00104 /*!
00105  * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00106  * to efficiently exchange synchronization data between acquisition and
00107  * tracking blocks
00108  */
00109 void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00110
00111 /*!
00112  * \brief Set acquisition channel unique ID
00113  */
00114 inline void set_channel(unsigned int channel) override
00115 {
00116     channel_ = channel;
00117     acquisition_fpga_>set_channel(channel_);
00118 }
00119
00120 /*!
00121  * \brief Set channel fsm associated to this acquisition instance
00122  */
00123 inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm) override
00124 {
00125     channel_fsm_ = std::move(channel_fsm);
00126     acquisition_fpga_>set_channel_fsm(channel_fsm_);
00127 }
00128
00129 /*!
00130  * \brief Set statistics threshold of PCPS algorithm
00131  */
00132 void set_threshold(float threshold) override;
00133
00134 /*!
00135  * \brief Set maximum Doppler off grid search
00136  */
00137 void set_doppler_max(unsigned int doppler_max) override;
00138
00139 /*!
00140  * \brief Set Doppler steps for the grid search
00141  */
00142 void set_doppler_step(unsigned int doppler_step) override;
00143
00144 /*!
00145  * \brief Set Doppler center for the grid search
00146  */
00147 void set_doppler_center(int doppler_center) override;
00148
00149 /*!
00150  * \brief Initializes acquisition algorithm.
00151  */
00152 void init() override;
00153
00154 /*!
00155  * \brief Sets local Galileo E5a code for PCPS acquisition algorithm.
00156  */
00157 void set_local_code() override;
00158
00159 /*!
00160  * \brief Returns the maximum peak of grid search
00161  */
00162 signed int mag() override;
00163
00164 /*!
00165  * \brief Restart acquisition algorithm
00166  */
00167 void reset() override;
00168
00169 /*!
00170  * \brief If set to 1, ensures that acquisition starts at the
00171  * first available sample.
00172  * \param state - int=1 forces start of acquisition
00173  */
00174 void set_state(int state) override;
00175
00176 /*!
00177  * \brief This function is only used in the unit tests
00178  */

```



```

00179 void set_single_doppler_flag(unsigned int single_doppler_flag);
00180
00181 /*!
00182  * \brief Stop running acquisition
00183  */
00184 void stop_acquisition() override;
00185
00186 /*!
00187  * \brief Set resampler latency
00188  */
00189 void set_resampler_latency(uint32_t latency_samples __attribute__((unused))) override {};
00190
00191 private:
00192     static const uint32_t ACQ_BUFF_1 = 1; // FPGA Acquisition IP buffer containing L2 or
L5/E5 frequency band samples by default.
00193     static const uint32_t DEFAULT_FPGA_BLK_EXP = 13; // default block exponent
00194     static const uint32_t QUANT_BITS_LOCAL_CODE = 16;
00195     static const uint32_t SELECT_LSBITS = 0x0000FFFF; // Select the 10 Lsbits out of a 20-bit
word
00196     static const uint32_t SELECT_MSBITS = 0xFFFF0000; // Select the 10 MSbits out of a 20-bit
word
00197     static const uint32_t SELECT_ALL_CODE_BITS = 0xFFFFFFFF; // Select a 20 bit word
00198     static const uint32_t SHL_CODE_BITS = 65536; // shift left by 10 bits
00199
00200 void generate_galileo_e5a_prn_codes();
00201
00202 pcps_acquisition_fpga_sptr acquisition_fpga_;
00203 std::weak_ptr<ChannelFsm> channel_fsm_;
00204 volk_gnssdr::vector<uint32_t> d_all_fft_codes_; // memory that contains all the code ffts
00205 Gnss_Synchro* gnss_synchro_;
00206 Acq_Conf_Fpga acq_parameters_;
00207 std::string role_;
00208 int32_t doppler_center_;
00209 uint32_t channel_;
00210 uint32_t doppler_max_;
00211 uint32_t doppler_step_;
00212 unsigned int in_streams_;
00213 unsigned int out_streams_;
00214 bool acq_pilot_;
00215 bool acq_iq_;
00216 };
00217
00218
00219 /** \} */
00220 /** \} */
00221 #endif // GNSS_SDR_GALILEO_E5A_PCPS_ACQUISITION_FPGA_H

```

## 11.23 galileo\_e5b\_pcps\_acquisition.h File Reference

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Galileo E5b data and pilot Signals.

```

#include "channel_fsm.h"
#include "gnss_synchro.h"
#include "pcps_acquisition.h"
#include <volk_gnssdr/volk_gnssdr_alloc.h>
#include <memory>
#include <string>
#include <utility>

```

Include dependency graph for galileo\_e5b\_pcps\_acquisition.h:



### Classes

- class [GalileoE5bPcpsAcquisition](#)

### 11.23.1 Detailed Description

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Galileo E5b data and pilot Signals.

**Author**

Piyush Gupta, 2020. [piyush04111999@gmail.com](mailto:piyush04111999@gmail.com)

**Note**

Code added as part of GSoC 2020 program.

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [galileo\\_e5b\\_pcps\\_acquisition.h](#).

---

## 11.24 galileo\_e5b\_pcps\_acquisition.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file galileo_e5b_pcps_acquisition.h
00003  * \brief Adapts a PCPS acquisition block to an AcquisitionInterface for
00004  * \brief Galileo E5b data and pilot Signals
00005  * \author Piyush Gupta, 2020. piyush04111999@gmail.com
00006  * \note Code added as part of GSoC 2020 program.
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019 #ifndef GNSS_SDR_GALILEO_E5B_PCPS_ACQUISITION_H
00020 #define GNSS_SDR_GALILEO_E5B_PCPS_ACQUISITION_H
00021
00022
00023 #include "channel_fsm.h"
00024 #include "gnss_synchro.h"
00025 #include "pcps_acquisition.h"
00026 #include <volk_gnssdr/volk_gnssdr_alloc.h>
00027 #include <memory>
00028 #include <string>
00029 #include <utility>
00030
00031 /** \addtogroup Acquisition
00032  * \{ */
00033 /** \addtogroup Acq_adapters
00034  * \{ */
00035
00036
00037 class ConfigurationInterface;
00038
00039 class GalileoE5bPcpsAcquisition : public AcquisitionInterface
00040 {
00041 public:
00042     /*!
00043     * \brief Constructor
00044     */
00045     GalileoE5bPcpsAcquisition(const ConfigurationInterface* configuration,
00046                               const std::string& role,
00047                               unsigned int in_streams,
00048                               unsigned int out_streams);
00049
00050     /*!
00051     * \brief Destructor
00052     */
00053     ~GalileoE5bPcpsAcquisition() = default;
00054
00055     /*!
00056     * \brief Role
00057     */
00058     inline std::string role() override
00059     {
00060         return role_;
00061     }
00062
00063     /*!
00064     * \brief Returns "GALILEO_E5b_PCPS_Acquisition"
00065     */
00066

```

```

00067     inline std::string implementation() override
00068     {
00069         return "Galileo_E5b_PCPS_Acquisition";
00070     }
00071
00072     /*!
00073      * \brief Returns size of lv_l6sc_t
00074      */
00075     inline size_t item_size() override
00076     {
00077         return sizeof(int16_t);
00078     }
00079
00080     /*!
00081      * \brief Connect
00082      */
00083     void connect(gr::top_block_sptr top_block) override;
00084
00085     /*!
00086      * \brief Disconnect
00087      */
00088     void disconnect(gr::top_block_sptr top_block) override;
00089
00090     /*!
00091      * \brief Get left block
00092      */
00093     gr::basic_block_sptr get_left_block() override;
00094
00095     /*!
00096      * \brief Get right block
00097      */
00098     gr::basic_block_sptr get_right_block() override;
00099
00100     /*!
00101      * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00102      * to efficiently exchange synchronization data between acquisition and
00103      * tracking blocks
00104      */
00105     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00106
00107     /*!
00108      * \brief Set acquisition channel unique ID
00109      */
00110     inline void set_channel(unsigned int channel) override
00111     {
00112         channel_ = channel;
00113         acquisition_>set_channel(channel_);
00114     }
00115
00116     /*!
00117      * \brief Set channel fsm associated to this acquisition instance
00118      */
00119     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm) override
00120     {
00121         channel_fsm_ = std::move(channel_fsm);
00122         acquisition_>set_channel_fsm(channel_fsm_);
00123     }
00124
00125     /*!
00126      * \brief Set statistics threshold of PCPS algorithm
00127      */
00128     void set_threshold(float threshold) override;
00129
00130     /*!
00131      * \brief Set maximum Doppler off grid search
00132      */
00133     void set_doppler_max(unsigned int doppler_max) override;
00134
00135     /*!
00136      * \brief Set Doppler steps for the grid search
00137      */
00138     void set_doppler_step(unsigned int doppler_step) override;
00139
00140     /*!
00141      * \brief Set Doppler center for the grid search
00142      */
00143     void set_doppler_center(int doppler_center) override;
00144
00145     /*!
00146      * \brief Initializes acquisition algorithm.
00147      */
00148     void init() override;
00149
00150     /*!
00151      * \brief Sets local Galileo E5b code for PCPS acquisition algorithm.
00152      */
00153     void set_local_code() override;

```

```

00154
00155     /*!
00156     * \brief Returns the maximum peak of grid search
00157     */
00158     signed int mag() override;
00159
00160     /*!
00161     * \brief Restart acquisition algorithm
00162     */
00163     void reset() override;
00164
00165     /*!
00166     * \brief If set to 1, ensures that acquisition starts at the
00167     * first available sample.
00168     * \param state - int=1 forces start of acquisition
00169     */
00170     void set_state(int state) override;
00171
00172     /*!
00173     * \brief Stop running acquisition
00174     */
00175     void stop_acquisition() override;
00176
00177     /*!
00178     * \brief Sets the resampler latency to account it in the acquisition code delay estimation
00179     */
00180     void set_resampler_latency(uint32_t latency_samples) override;
00181
00182 private:
00183     pcps_acquisition_sptr acquisition_;
00184
00185     volk_gnssssdr::vector<std::complex<float>> code_;
00186     std::weak_ptr<ChannelFsm> channel_fsm_;
00187
00188     Gnss_Synchro* gnss_synchro_;
00189     Acq_Conf acq_parameters_;
00190
00191     std::string item_type_;
00192     std::string dump_filename_;
00193     std::string role_;
00194
00195     size_t item_size_;
00196     int64_t fs_in_;
00197
00198     float threshold_;
00199     int doppler_center_;
00200     unsigned int vector_length_;
00201     unsigned int code_length_;
00202     unsigned int channel_;
00203     unsigned int doppler_max_;
00204     unsigned int doppler_step_;
00205     unsigned int sampled_ms_;
00206     unsigned int in_streams_;
00207     unsigned int out_streams_;
00208
00209     bool acq_pilot_;
00210     bool acq_iq_;
00211 };
00212
00213
00214 /** \} */
00215 /** \} */
00216 #endif // GNSS_SDR_GALILEO_E5B_PCPS_ACQUISITION_H

```

## 11.25 galileo\_e5b\_pcps\_acquisition\_fpga.h File Reference

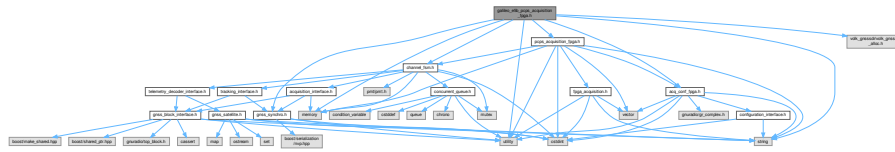
Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Galileo E5b data and pilot Signals for the FPGA.

```

#include "acq_conf_fpga.h"
#include "channel_fsm.h"
#include "gnss_synchro.h"
#include "pcps_acquisition_fpga.h"
#include <volk_gnssssdr/volk_gnssssdr_alloc.h>
#include <memory>
#include <string>
#include <utility>

```

Include dependency graph for galileo\_e5b\_pcps\_acquisition\_fpga.h:



## Classes

- class [GalileoE5bPcpsAcquisitionFpga](#)

*This class adapts a PCPS acquisition block off-loaded on an FPGA to an [AcquisitionInterface](#) for Galileo E5b signals.*

### 11.25.1 Detailed Description

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Galileo E5b data and pilot Signals for the FPGA.

#### Author

Piyush Gupta, 2020. [piyush04111999@gmail.com](mailto:piyush04111999@gmail.com)

#### Note

Code added as part of GSoC 2020 Program.

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [galileo\\_e5b\\_pcps\\_acquisition\\_fpga.h](#).

---

## 11.26 galileo\_e5b\_pcps\_acquisition\_fpga.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file galileo_e5b_pcps_acquisition_fpga.h
00003  * \brief Adapts a PCPS acquisition block to an AcquisitionInterface for
00004  * Galileo E5b data and pilot Signals for the FPGA
00005  * \author Piyush Gupta, 2020. piyush04111999@gmail.com
00006  * \note Code added as part of GSoC 2020 Program.
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019 #ifndef GNSS_SDR_GALILEO_E5B_PCPS_ACQUISITION_FPGA_H
00020 #define GNSS_SDR_GALILEO_E5B_PCPS_ACQUISITION_FPGA_H
00021
00022 #include "acq_conf_fpga.h"
00023 #include "channel_fsm.h"
00024 #include "gnss_synchro.h"
00025 #include "pcps_acquisition_fpga.h"
00026 #include <volk_gnssdr/volk_gnssdr_alloc.h>
00027 #include <memory>
00028 #include <string>
00029 #include <utility>
00030
00031 /** \addtogroup Acquisition
00032  * \{ */
00033 /** \addtogroup Acq_adapters
00034  * \{ */
00035
00036
00037 class ConfigurationInterface;
00038
```

```

00039  /*!
00040  * \brief This class adapts a PCPS acquisition block off-loaded on an FPGA
00041  * to an AcquisitionInterface for Galileo E5b signals
00042  */
00043  class GalileoE5bPcpsAcquisitionFpga : public AcquisitionInterface
00044  {
00045  public:
00046      /*!
00047      * \brief Constructor
00048      */
00049      GalileoE5bPcpsAcquisitionFpga(const ConfigurationInterface* configuration,
00050          const std::string& role,
00051          unsigned int in_streams,
00052          unsigned int out_streams);
00053
00054      /*!
00055      * \brief Destructor
00056      */
00057      ~GalileoE5bPcpsAcquisitionFpga() = default;
00058
00059      /*!
00060      * \brief Role
00061      */
00062      inline std::string role() override
00063      {
00064          return role_;
00065      }
00066
00067      /*!
00068      * \brief Returns "Galileo_E5b_Pcps_Acquisition_FPGA"
00069      */
00070      inline std::string implementation() override
00071      {
00072          return "Galileo_E5b_PCPS_Acquisition_FPGA";
00073      }
00074
00075      /*!
00076      * \brief Returns size of lv_l6sc_t
00077      */
00078      inline size_t item_size() override
00079      {
00080          return sizeof(int16_t);
00081      }
00082
00083      /*!
00084      * \brief Connect
00085      */
00086      void connect(gr::top_block_sptr top_block) override;
00087
00088      /*!
00089      * \brief Disconnect
00090      */
00091      void disconnect(gr::top_block_sptr top_block) override;
00092
00093      /*!
00094      * \brief Get left block
00095      */
00096      gr::basic_block_sptr get_left_block() override;
00097
00098      /*!
00099      * \brief Get right block
00100      */
00101      gr::basic_block_sptr get_right_block() override;
00102
00103      /*!
00104      * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00105      * to efficiently exchange synchronization data between acquisition and
00106      * tracking blocks
00107      */
00108      void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00109
00110      /*!
00111      * \brief Set acquisition channel unique ID
00112      */
00113      inline void set_channel(unsigned int channel) override
00114      {
00115          channel_ = channel;
00116          acquisition_fpga->set_channel(channel_);
00117      }
00118
00119      /*!
00120      * \brief Set channel fsm associated to this acquisition instance
00121      */
00122      inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm) override
00123      {
00124          channel_fsm_ = std::move(channel_fsm);
00125          acquisition_fpga->set_channel_fsm(channel_fsm_);

```

```

00126     }
00127
00128     /*!
00129     * \brief Set statistics threshold of PCPS algorithm
00130     */
00131     void set_threshold(float threshold) override;
00132
00133     /*!
00134     * \brief Set maximum Doppler off grid search
00135     */
00136     void set_doppler_max(unsigned int doppler_max) override;
00137
00138     /*!
00139     * \brief Set Doppler steps for the grid search
00140     */
00141     void set_doppler_step(unsigned int doppler_step) override;
00142
00143     /*!
00144     * \brief Set Doppler center for the grid search
00145     */
00146     void set_doppler_center(int doppler_center) override;
00147
00148     /*!
00149     * \brief Initializes acquisition algorithm.
00150     */
00151     void init() override;
00152
00153     /*!
00154     * \brief Sets local Galileo E5b code for PCPS acquisition algorithm.
00155     */
00156     void set_local_code() override;
00157
00158     /*!
00159     * \brief Returns the maximum peak of grid search
00160     */
00161     signed int mag() override;
00162
00163     /*!
00164     * \brief Restart acquisition algorithm
00165     */
00166     void reset() override;
00167
00168     /*!
00169     * \brief If set to 1, ensures that acquisition starts at the
00170     * first available sample.
00171     * \param state - int=1 forces start of acquisition
00172     */
00173     void set_state(int state) override;
00174
00175     /*!
00176     * \brief This function is only used in the unit tests
00177     */
00178     void set_single_doppler_flag(unsigned int single_doppler_flag);
00179
00180     /*!
00181     * \brief Stop running acquisition
00182     */
00183     void stop_acquisition() override;
00184
00185     /*!
00186     * \brief Set resampler latency
00187     */
00188     void set_resampler_latency(uint32_t latency_samples __attribute__((unused))) override {};
00189
00190 private:
00191     static const uint32_t ACQ_BUFF_1 = 1; // FPGA Acquisition IP buffer containing L2 or
00192     L5/E5 frequency band samples by default.
00193     static const uint32_t DEFAULT_FPGA_BLK_EXP = 13; // default block exponent
00194     static const uint32_t QUANT_BITS_LOCAL_CODE = 16;
00195     static const uint32_t SELECT_LSBITS = 0x0000FFFF; // Select the 10 LSbits out of a 20-bit
00196     word
00197     static const uint32_t SELECT_MSBITS = 0xFFFF0000; // Select the 10 MSbits out of a 20-bit
00198     word
00199     static const uint32_t SELECT_ALL_CODE_BITS = 0xFFFFFFFF; // Select a 20 bit word
00200     static const uint32_t SHL_CODE_BITS = 65536; // shift left by 10 bits
00201
00202     void generate_galileo_e5b_prn_codes();
00203     pcps_acquisition_fpga_sptr acquisition_fpga_;
00204     volk_gnssdr::vector<uint32_t> d_all_fft_codes_; // memory that contains all the code ffts
00205     std::weak_ptr<ChannelFsm> channel_fsm_;
00206
00207     Gnss_Synchro* gnss_synchro_;
00208     Acq_Conf_Fpga acq_parameters_;
00209     std::string role_;
00210     int32_t doppler_center_;
00211     uint32_t channel_;

```

```

00210     uint32_t doppler_max_;
00211     uint32_t doppler_step_;
00212     unsigned int in_streams_;
00213     unsigned int out_streams_;
00214     bool acq_pilot_;
00215     bool acq_iq_;
00216 };
00217
00218
00219 /** \} */
00220 /** \} */
00221 #endif // GNSS_SDR_GALILEO_E5B_PCPS_ACQUISITION_FPGA_H

```

## 11.27 galileo\_e6\_pcps\_acquisition.h File Reference

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Galileo E6 B/C Signals.

```

#include "acq_conf.h"
#include "channel_fsm.h"
#include "complex_byte_to_float_x2.h"
#include "gnss_synchro.h"
#include "pcps_acquisition.h"
#include <gnuradio/blocks/float_to_complex.h>
#include <volk_gnssssdr/volk_gnssssdr_alloc.h>
#include <memory>
#include <string>
#include <utility>

```

Include dependency graph for galileo\_e6\_pcps\_acquisition.h:



### Classes

- class [GalileoE6PcpsAcquisition](#)

*This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Galileo E6 Signals.*

### 11.27.1 Detailed Description

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Galileo E6 B/C Signals.

#### Author

Carles Fernandez-Prades, 2020. cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [galileo\\_e6\\_pcps\\_acquisition.h](#).

---

## 11.28 galileo\_e6\_pcps\_acquisition.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file galileo_e6_pcps_acquisition.h
00003  * \brief Adapts a PCPS acquisition block to an AcquisitionInterface for
00004  * Galileo E6 B/C Signals
00005  * \author Carles Fernandez-Prades, 2020. cfernandez(at)cttc.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.

```



```

00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_GALILEO_E6_PCPS_ACQUISITION_H
00019 #define GNSS_SDR_GALILEO_E6_PCPS_ACQUISITION_H
00020
00021 #include "acq_conf.h"
00022 #include "channel_fsm.h"
00023 #include "complex_byte_to_float_x2.h"
00024 #include "gnss_synchro.h"
00025 #include "pcps_acquisition.h"
00026 #include <gnuradio/blocks/float_to_complex.h>
00027 #include <volk_gnssssdr/volk_gnssssdr_alloc.h>
00028 #include <memory>
00029 #include <string>
00030 #include <utility>
00031
00032 /** \addtogroup Acquisition
00033  * \{ */
00034 /** \addtogroup Acq_adapters
00035  * \{ */
00036
00037
00038 class ConfigurationInterface;
00039
00040 /*!
00041  * \brief This class adapts a PCPS acquisition block to an
00042  * AcquisitionInterface for Galileo E6 Signals
00043  */
00044 class GalileoE6PcpsAcquisition : public AcquisitionInterface
00045 {
00046 public:
00047     GalileoE6PcpsAcquisition(
00048         const ConfigurationInterface* configuration,
00049         const std::string& role,
00050         unsigned int in_streams,
00051         unsigned int out_streams);
00052
00053     ~GalileoE6PcpsAcquisition() = default;
00054
00055     inline std::string role() override
00056     {
00057         return role_;
00058     }
00059
00060     /*!
00061     * \brief Returns "Galileo_E6_PCPS_Acquisition"
00062     */
00063     inline std::string implementation() override
00064     {
00065         return "Galileo_E6_PCPS_Acquisition";
00066     }
00067
00068     size_t item_size() override
00069     {
00070         return item_size_;
00071     }
00072
00073     void connect(gr::top_block_sptr top_block) override;
00074     void disconnect(gr::top_block_sptr top_block) override;
00075     gr::basic_block_sptr get_left_block() override;
00076     gr::basic_block_sptr get_right_block() override;
00077
00078     /*!
00079     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00080     * to efficiently exchange synchronization data between acquisition and
00081     * tracking blocks
00082     */
00083     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00084
00085     /*!
00086     * \brief Set acquisition channel unique ID
00087     */
00088     inline void set_channel(unsigned int channel) override
00089     {
00090         channel_ = channel;
00091         acquisition_>set_channel(channel_);
00092     }
00093
00094     /*!
00095     * \brief Set channel fsm associated to this acquisition instance
00096     */
00097     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm) override

```

```

00098     {
00099         channel_fsm_ = std::move(channel_fsm);
00100         acquisition_>set_channel_fsm(channel_fsm_);
00101     }
00102
00103     /*!
00104      * \brief Set statistics threshold of PCPS algorithm
00105      */
00106     void set_threshold(float threshold) override;
00107
00108     /*!
00109      * \brief Set maximum Doppler off grid search
00110      */
00111     void set_doppler_max(unsigned int doppler_max) override;
00112
00113     /*!
00114      * \brief Set Doppler steps for the grid search
00115      */
00116     void set_doppler_step(unsigned int doppler_step) override;
00117
00118     /*!
00119      * \brief Set Doppler center for the grid search
00120      */
00121     void set_doppler_center(int doppler_center) override;
00122
00123     /*!
00124      * \brief Initializes acquisition algorithm.
00125      */
00126     void init() override;
00127
00128     /*!
00129      * \brief Sets local code for Galileo E1 PCPS acquisition algorithm.
00130      */
00131     void set_local_code() override;
00132
00133     /*!
00134      * \brief Returns the maximum peak of grid search
00135      */
00136     signed int mag() override;
00137
00138     /*!
00139      * \brief Restart acquisition algorithm
00140      */
00141     void reset() override;
00142
00143     /*!
00144      * \brief If state = 1, it forces the block to start acquiring from the first sample
00145      */
00146     void set_state(int state) override;
00147
00148     /*!
00149      * \brief Stop running acquisition
00150      */
00151     void stop_acquisition() override;
00152
00153     /*!
00154      * \brief Sets the resampler latency to account it in the acquisition code delay estimation
00155      */
00156     void set_resampler_latency(uint32_t latency_samples) override;
00157
00158 private:
00159     pcps_acquisition_sptr acquisition_;
00160     volk_gnssdr::vector<std::complex<float>> code_;
00161     std::weak_ptr<ChannelFsm> channel_fsm_;
00162     gr::blocks::float_to_complex::sptr float_to_complex_;
00163     complex_byte_to_float_x2_sptr cbyte_to_float_x2_;
00164     Gnss_Synchro* gnss_synchro_;
00165     const ConfigurationInterface* configuration_;
00166     Acq_Conf acq_parameters_;
00167     std::string item_type_;
00168     std::string dump_filename_;
00169     std::string role_;
00170     int64_t fs_in_;
00171     size_t item_size_;
00172     float threshold_;
00173     int doppler_center_;
00174     unsigned int vector_length_;
00175     unsigned int code_length_;
00176     unsigned int channel_;
00177     unsigned int doppler_max_;
00178     unsigned int doppler_step_;
00179     unsigned int sampled_ms_;
00180     unsigned int in_streams_;
00181     unsigned int out_streams_;
00182 };
00183
00184

```

```

00185 /** \} */
00186 /** \} */
00187 #endif // GNSS_SDR_GALILEO_E6_PCPS_ACQUISITION_H

```

## 11.29 glonass\_l1\_ca\_pcps\_acquisition.h File Reference

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Glonass L1 C/A signals.

```

#include "acq_conf.h"
#include "channel_fsm.h"
#include "complex_byte_to_float_x2.h"
#include "gnss_synchro.h"
#include "pcps_acquisition.h"
#include <gnuradio/blocks/float_to_complex.h>
#include <volk_gnssssdr/volk_gnssssdr_alloc.h>
#include <memory>
#include <string>
#include <utility>

```

Include dependency graph for glonass\_l1\_ca\_pcps\_acquisition.h:



### Classes

- class [GlonassL1CaPcpsAcquisition](#)

*This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.*

### 11.29.1 Detailed Description

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Glonass L1 C/A signals.

#### Author

Gabriel Araujo, 2017. gabriel.araujo.5000(at)gmail.com

Luis Esteve, 2017. luis(at)epsilon-formacion.com

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [glonass\\_l1\\_ca\\_pcps\\_acquisition.h](#).

---

## 11.30 glonass\_l1\_ca\_pcps\_acquisition.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002 * \file glonass_l1_ca_pcps_acquisition.h
00003 * \brief Adapts a PCPS acquisition block to an AcquisitionInterface for
00004 * Glonass L1 C/A signals
00005 * \author Gabriel Araujo, 2017. gabriel.araujo.5000(at)gmail.com
00006 * \author Luis Esteve, 2017. luis(at)epsilon-formacion.com
00007 *
00008 *
00009 * -----
00010 *
00011 * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00012 * This file is part of GNSS-SDR.
00013 *
00014 * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00015 * SPDX-License-Identifier: GPL-3.0-or-later
00016 *
00017 * -----

```

```

00018  */
00019
00020 #ifndef GNSS_SDR_GLONASS_L1_CA_PCPS_ACQUISITION_H
00021 #define GNSS_SDR_GLONASS_L1_CA_PCPS_ACQUISITION_H
00022
00023 #include "acq_conf.h"
00024 #include "channel_fsm.h"
00025 #include "complex_byte_to_float_x2.h"
00026 #include "gnss_synchro.h"
00027 #include "pcps_acquisition.h"
00028 #include <gnuradio/blocks/float_to_complex.h>
00029 #include <volk_gnssssdr/volk_gnssssdr_alloc.h>
00030 #include <memory>
00031 #include <string>
00032 #include <utility>
00033
00034 /** \addtogroup Acquisition
00035  * \{ */
00036 /** \addtogroup Acq_adapters
00037  * \{ */
00038
00039
00040 class ConfigurationInterface;
00041
00042 /*!
00043  * \brief This class adapts a PCPS acquisition block to an AcquisitionInterface
00044  * for GPS L1 C/A signals
00045  */
00046 class GlonassL1CaPcpsAcquisition : public AcquisitionInterface
00047 {
00048 public:
00049     GlonassL1CaPcpsAcquisition(
00050         const ConfigurationInterface* configuration,
00051         const std::string& role,
00052         unsigned int in_streams,
00053         unsigned int out_streams);
00054
00055     ~GlonassL1CaPcpsAcquisition() = default;
00056
00057     inline std::string role() override
00058     {
00059         return role_;
00060     }
00061
00062     /*!
00063     * \brief Returns "GLONASS_L1_CA_PCPS_Acquisition"
00064     */
00065     inline std::string implementation() override
00066     {
00067         return "GLONASS_L1_CA_PCPS_Acquisition";
00068     }
00069
00070     inline size_t item_size() override
00071     {
00072         return item_size_;
00073     }
00074
00075     void connect(gr::top_block_sptr top_block) override;
00076     void disconnect(gr::top_block_sptr top_block) override;
00077     gr::basic_block_sptr get_left_block() override;
00078     gr::basic_block_sptr get_right_block() override;
00079
00080     /*!
00081     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00082     * to efficiently exchange synchronization data between acquisition and
00083     * tracking blocks
00084     */
00085     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00086
00087     /*!
00088     * \brief Set acquisition channel unique ID
00089     */
00090     inline void set_channel(unsigned int channel) override
00091     {
00092         channel_ = channel;
00093         acquisition_->set_channel(channel_);
00094     }
00095
00096     /*!
00097     * \brief Set channel fsm associated to this acquisition instance
00098     */
00099     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm) override
00100     {
00101         channel_fsm_ = std::move(channel_fsm);
00102         acquisition_->set_channel_fsm(channel_fsm_);
00103     }
00104

```

```

00105     /*!
00106      * \brief Set statistics threshold of PCPS algorithm
00107      */
00108     void set_threshold(float threshold) override;
00109
00110     /*!
00111      * \brief Set maximum Doppler off grid search
00112      */
00113     void set_doppler_max(unsigned int doppler_max) override;
00114
00115     /*!
00116      * \brief Set Doppler steps for the grid search
00117      */
00118     void set_doppler_step(unsigned int doppler_step) override;
00119
00120     /*!
00121      * \brief Initializes acquisition algorithm.
00122      */
00123     void init() override;
00124
00125     /*!
00126      * \brief Sets local code for GPS L1/CA PCPS acquisition algorithm.
00127      */
00128     void set_local_code() override;
00129
00130     /*!
00131      * \brief Returns the maximum peak of grid search
00132      */
00133     signed int mag() override;
00134
00135     /*!
00136      * \brief Restart acquisition algorithm
00137      */
00138     void reset() override;
00139
00140     /*!
00141      * \brief If state = 1, it forces the block to start acquiring from the first sample
00142      */
00143     void set_state(int state) override;
00144
00145     /*!
00146      * \brief Stop running acquisition
00147      */
00148     void stop_acquisition() override;
00149
00150     void set_resampler_latency(uint32_t latency_samples __attribute__((unused))) override {};
00151
00152 private:
00153     pcps_acquisition_sptr acquisition_;
00154     volk_gnss_sdr::vector<std::complex<float>> code_;
00155     std::weak_ptr<ChannelFsm> channel_fsm_;
00156     gr::blocks::float_to_complex::sptr float_to_complex_;
00157     complex_byte_to_float_x2_sptr cbyte_to_float_x2_;
00158     Gnss_Synchro* gnss_synchro_;
00159     Acq_Conf acq_parameters_;
00160     std::string item_type_;
00161     std::string dump_filename_;
00162     std::string role_;
00163     int64_t fs_in_;
00164     size_t item_size_;
00165     float threshold_;
00166     unsigned int vector_length_;
00167     unsigned int code_length_;
00168     unsigned int channel_;
00169     unsigned int doppler_max_;
00170     unsigned int doppler_step_;
00171     unsigned int sampled_ms_;
00172     unsigned int in_streams_;
00173     unsigned int out_streams_;
00174 };
00175
00176
00177 /** @} */
00178 /** @} */
00179 #endif // GNSS_SDR_GLOPASS_L1_CA_PCPS_ACQUISITION_H

```

## 11.31 glonass\_l2\_ca\_pcps\_acquisition.h File Reference

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Glonass L2 C/A signals.

```

#include "acq_conf.h"
#include "channel_fsm.h"
#include "complex_byte_to_float_x2.h"

```

```
#include "gnss_synchro.h"
#include "pcps_acquisition.h"
#include <gnuradio/blocks/float_to_complex.h>
#include <volk_gnssdr/volk_gnssdr_alloc.h>
#include <memory>
#include <string>
#include <utility>
```

Include dependency graph for glonass\_l2\_ca\_pcps\_acquisition.h:



## Classes

- class [GlonassL2CaPcpsAcquisition](#)

*This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GLONASS L2 C/A signals.*

### 11.31.1 Detailed Description

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Glonass L2 C/A signals.

#### Author

Damian Miralles, 2018, [dmiralles2009@gmail.com](mailto:dmiralles2009@gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [glonass\\_l2\\_ca\\_pcps\\_acquisition.h](#).

---

## 11.32 glonass\_l2\_ca\_pcps\_acquisition.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file glonass_l2_ca_pcps_acquisition.h
00003  * \brief Adapts a PCPS acquisition block to an AcquisitionInterface for
00004  * Glonass L2 C/A signals
00005  * \author Damian Miralles, 2018, dmiralles2009@gmail.com
00006  *
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019 #ifndef GNSS_SDR_GLOPASS_L2_CA_PCPS_ACQUISITION_H
00020 #define GNSS_SDR_GLOPASS_L2_CA_PCPS_ACQUISITION_H
00021
00022 #include "acq_conf.h"
00023 #include "channel_fsm.h"
00024 #include "complex_byte_to_float_x2.h"
00025 #include "gnss_synchro.h"
00026 #include "pcps_acquisition.h"
00027 #include <gnuradio/blocks/float_to_complex.h>
00028 #include <volk_gnssdr/volk_gnssdr_alloc.h>
00029 #include <memory>
00030 #include <string>
00031 #include <utility>
00032
00033 /** \addtogroup Acquisition
00034  * \{ */
```

```

00035 /** \addtogroup Acq_adapters
00036 * \{ */
00037
00038
00039 class ConfigurationInterface;
00040
00041 /*!
00042 * \brief This class adapts a PCPS acquisition block to an AcquisitionInterface
00043 * for GLONASS L2 C/A signals
00044 */
00045 class GlonassL2CaPcpsAcquisition : public AcquisitionInterface
00046 {
00047 public:
00048     GlonassL2CaPcpsAcquisition(
00049         const ConfigurationInterface* configuration,
00050         const std::string& role,
00051         unsigned int in_streams,
00052         unsigned int out_streams);
00053
00054     ~GlonassL2CaPcpsAcquisition() = default;
00055
00056     inline std::string role() override
00057     {
00058         return role_;
00059     }
00060
00061     /*!
00062     * \brief Returns "GLONASS_L2_CA_PCPS_Acquisition"
00063     */
00064     inline std::string implementation() override
00065     {
00066         return "GLONASS_L2_CA_PCPS_Acquisition";
00067     }
00068
00069     inline size_t item_size() override
00070     {
00071         return item_size_;
00072     }
00073
00074     void connect(gr::top_block_sptr top_block) override;
00075     void disconnect(gr::top_block_sptr top_block) override;
00076     gr::basic_block_sptr get_left_block() override;
00077     gr::basic_block_sptr get_right_block() override;
00078
00079     /*!
00080     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00081     * to efficiently exchange synchronization data between acquisition and
00082     * tracking blocks
00083     */
00084     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00085
00086     /*!
00087     * \brief Set acquisition channel unique ID
00088     */
00089     inline void set_channel(unsigned int channel) override
00090     {
00091         channel_ = channel;
00092         acquisition_->set_channel(channel_);
00093     }
00094
00095     /*!
00096     * \brief Set channel fsm associated to this acquisition instance
00097     */
00098     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm) override
00099     {
00100         channel_fsm_ = std::move(channel_fsm);
00101         acquisition_->set_channel_fsm(channel_fsm_);
00102     }
00103
00104     /*!
00105     * \brief Set statistics threshold of PCPS algorithm
00106     */
00107     void set_threshold(float threshold) override;
00108
00109     /*!
00110     * \brief Set maximum Doppler off grid search
00111     */
00112     void set_doppler_max(unsigned int doppler_max) override;
00113
00114     /*!
00115     * \brief Set Doppler steps for the grid search
00116     */
00117     void set_doppler_step(unsigned int doppler_step) override;
00118
00119     /*!
00120     * \brief Initializes acquisition algorithm.
00121     */

```

```

00122     void init() override;
00123
00124     /*!
00125      * \brief Sets local code for GLONASS L2/CA PCPS acquisition algorithm.
00126      */
00127     void set_local_code() override;
00128
00129     /*!
00130      * \brief Returns the maximum peak of grid search
00131      */
00132     signed int mag() override;
00133
00134     /*!
00135      * \brief Restart acquisition algorithm
00136      */
00137     void reset() override;
00138
00139     /*!
00140      * \brief If state = 1, it forces the block to start acquiring from the first sample
00141      */
00142     void set_state(int state) override;
00143
00144     /*!
00145      * \brief Stop running acquisition
00146      */
00147     void stop_acquisition() override;
00148
00149     void set_resampler_latency(uint32_t latency_samples __attribute__((unused))) override {};
00150
00151 private:
00152     pcps_acquisition_sptr acquisition_;
00153     volk_gnssssdr::vector<std::complex<float>> code_;
00154     std::weak_ptr<ChannelFsm> channel_fsm_;
00155     gr::blocks::float_to_complex::sptr float_to_complex_;
00156     complex_byte_to_float_x2_sptr cbyte_to_float_x2_;
00157     Gnss_Synchro* gnss_synchro_;
00158     Acq_Conf acq_parameters_;
00159     std::string item_type_;
00160     std::string dump_filename_;
00161     std::string role_;
00162     int64_t fs_in_;
00163     size_t item_size_;
00164     float threshold_;
00165     unsigned int vector_length_;
00166     unsigned int code_length_;
00167     unsigned int channel_;
00168     unsigned int doppler_max_;
00169     unsigned int doppler_step_;
00170     unsigned int sampled_ms_;
00171     unsigned int in_streams_;
00172     unsigned int out_streams_;
00173 };
00174
00175
00176 /** \} */
00177 /** \} */
00178 #endif // GNSS_SDR_GLO_NASS_L2_CA_PCPS_ACQUISITION_H

```

## 11.33 gps\_l1\_ca\_pcps\_acquisition.h File Reference

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.

```

#include "acq_conf.h"
#include "channel_fsm.h"
#include "complex_byte_to_float_x2.h"
#include "gnss_synchro.h"
#include "pcps_acquisition.h"
#include <gnuradio/blocks/float_to_complex.h>
#include <volk_gnssssdr/volk_gnssssdr_alloc.h>
#include <memory>
#include <string>
#include <utility>

```





```

00042
00043
00044 class ConfigurationInterface;
00045
00046 /*!
00047  * \brief This class adapts a PCPS acquisition block to an AcquisitionInterface
00048  * for GPS L1 C/A signals
00049  */
00050 class GpsL1CaPcpsAcquisition : public AcquisitionInterface
00051 {
00052 public:
00053     GpsL1CaPcpsAcquisition(
00054         const ConfigurationInterface* configuration,
00055         const std::string& role,
00056         unsigned int in_streams,
00057         unsigned int out_streams);
00058
00059     ~GpsL1CaPcpsAcquisition() = default;
00060
00061     inline std::string role() override
00062     {
00063         return role_;
00064     }
00065
00066     /*!
00067     * \brief Returns "GPS_L1_CA_PCPS_Acquisition"
00068     */
00069     inline std::string implementation() override
00070     {
00071         return "GPS_L1_CA_PCPS_Acquisition";
00072     }
00073
00074     inline size_t item_size() override
00075     {
00076         return item_size_;
00077     }
00078
00079     void connect(gr::top_block_sptr top_block) override;
00080     void disconnect(gr::top_block_sptr top_block) override;
00081     gr::basic_block_sptr get_left_block() override;
00082     gr::basic_block_sptr get_right_block() override;
00083
00084     /*!
00085     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00086     * to efficiently exchange synchronization data between acquisition and
00087     * tracking blocks
00088     */
00089     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00090
00091     /*!
00092     * \brief Set acquisition channel unique ID
00093     */
00094     inline void set_channel(unsigned int channel) override
00095     {
00096         channel_ = channel;
00097         acquisition_>set_channel(channel_);
00098     }
00099
00100     /*!
00101     * \brief Set channel fsm associated to this acquisition instance
00102     */
00103     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm) override
00104     {
00105         channel_fsm_ = std::move(channel_fsm);
00106         acquisition_>set_channel_fsm(channel_fsm_);
00107     }
00108
00109     /*!
00110     * \brief Set statistics threshold of PCPS algorithm
00111     */
00112     void set_threshold(float threshold) override;
00113
00114     /*!
00115     * \brief Set maximum Doppler off grid search
00116     */
00117     void set_doppler_max(unsigned int doppler_max) override;
00118
00119     /*!
00120     * \brief Set Doppler steps for the grid search
00121     */
00122     void set_doppler_step(unsigned int doppler_step) override;
00123
00124     /*!
00125     * \brief Set Doppler center for the grid search
00126     */
00127     void set_doppler_center(int doppler_center) override;
00128

```

```

00129     /*!
00130     * \brief Initializes acquisition algorithm.
00131     */
00132     void init() override;
00133
00134     /*!
00135     * \brief Sets local code for GPS L1/CA PCPS acquisition algorithm.
00136     */
00137     void set_local_code() override;
00138
00139     /*!
00140     * \brief Returns the maximum peak of grid search
00141     */
00142     signed int mag() override;
00143
00144     /*!
00145     * \brief Restart acquisition algorithm
00146     */
00147     void reset() override;
00148
00149     /*!
00150     * \brief If state = 1, it forces the block to start acquiring from the first sample
00151     */
00152     void set_state(int state) override;
00153
00154     /*!
00155     * \brief Stop running acquisition
00156     */
00157     void stop_acquisition() override;
00158
00159     /*!
00160     * \brief Sets the resampler latency to account it in the acquisition code delay estimation
00161     */
00162     void set_resampler_latency(uint32_t latency_samples) override;
00163
00164 private:
00165     pcps_acquisition_sptr acquisition_;
00166     volk_gnssdr::vector<std::complex<float>> code_;
00167     std::weak_ptr<ChannelFsm> channel_fsm_;
00168     gr::blocks::float_to_complex::sptr float_to_complex_;
00169     complex_byte_to_float_x2_sptr cbyte_to_float_x2_;
00170     Gnss_Synchro* gnss_synchro_;
00171     Acq_Conf acq_parameters_;
00172     std::string item_type_;
00173     std::string dump_filename_;
00174     std::string role_;
00175     size_t item_size_;
00176     float threshold_;
00177     int doppler_center_;
00178     unsigned int vector_length_;
00179     unsigned int code_length_;
00180     unsigned int channel_;
00181     unsigned int doppler_max_;
00182     unsigned int doppler_step_;
00183     unsigned int sampled_ms_;
00184     unsigned int in_streams_;
00185     unsigned int out_streams_;
00186 };
00187
00188
00189 /** @} */
00190 /** @} */
00191 #endif // GNSS_SDR_GPS_L1_CA_PCPS_ACQUISITION_H

```

## 11.35 gps\_l1\_ca\_pcps\_acquisition\_fine\_doppler.h File Reference

Adapts a PCPS acquisition block with fine Doppler estimation to an [AcquisitionInterface](#) for GPS L1 C/A signals.

```

#include "channel_fsm.h"
#include "gnss_synchro.h"
#include "pcps_acquisition_fine_doppler_cc.h"
#include <memory>
#include <string>
#include <utility>
#include <vector>

```



```

00035
00036
00037 using pcps_acquisition_fine_doppler_cc_sptr = gnss_shared_ptr<pcps_acquisition_fine_doppler_cc>;
00038
00039 class ConfigurationInterface;
00040
00041 /*!
00042  * \brief This class Adapts a PCPS acquisition block with fine Doppler estimation to an
00043  * AcquisitionInterface for
00044  * GPS L1 C/A signals
00045  */
00045 class GpsL1CaPcpsAcquisitionFineDoppler : public AcquisitionInterface
00046 {
00047 public:
00048     GpsL1CaPcpsAcquisitionFineDoppler(const ConfigurationInterface* configuration,
00049         const std::string& role,
00050         unsigned int in_streams,
00051         unsigned int out_streams);
00052
00053     ~GpsL1CaPcpsAcquisitionFineDoppler() = default;
00054
00055     inline std::string role() override
00056     {
00057         return role_;
00058     }
00059
00060     /*!
00061     * \brief Returns "GPS_L1_CA_PCPS_Acquisition_Fine_Doppler"
00062     */
00063     inline std::string implementation() override
00064     {
00065         return "GPS_L1_CA_PCPS_Acquisition_Fine_Doppler";
00066     }
00067
00068     inline size_t item_size() override
00069     {
00070         return item_size_;
00071     }
00072
00073     void connect(gnss_shared_ptr<gr::top_block> top_block) override;
00074     void disconnect(gnss_shared_ptr<gr::top_block> top_block) override;
00075     gnss_shared_ptr<gr::basic_block> get_left_block() override;
00076     gnss_shared_ptr<gr::basic_block> get_right_block() override;
00077
00078     /*!
00079     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00080     * to efficiently exchange synchronization data between acquisition and
00081     * tracking blocks
00082     */
00083     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00084
00085     /*!
00086     * \brief Set acquisition channel unique ID
00087     */
00088     inline void set_channel(unsigned int channel) override
00089     {
00090         channel_ = channel;
00091         acquisition_cc_>set_channel(channel_);
00092     }
00093
00094     /*!
00095     * \brief Set channel fsm associated to this acquisition instance
00096     */
00097     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm) override
00098     {
00099         channel_fsm_ = std::move(channel_fsm);
00100         acquisition_cc_>set_channel_fsm(channel_fsm_);
00101     }
00102
00103     /*!
00104     * \brief Set statistics threshold of PCPS algorithm
00105     */
00106     void set_threshold(float threshold) override;
00107
00108     /*!
00109     * \brief Set maximum Doppler off grid search
00110     */
00111     void set_doppler_max(unsigned int doppler_max) override;
00112
00113     /*!
00114     * \brief Set Doppler steps for the grid search
00115     */
00116     void set_doppler_step(unsigned int doppler_step) override;
00117
00118     /*!
00119     * \brief Initializes acquisition algorithm.
00120     */

```

```

00121     void init() override;
00122
00123     void set_local_code() override;
00124
00125     /*!
00126      * \brief Returns the maximum peak of grid search
00127      */
00128     signed int mag() override;
00129
00130     /*!
00131      * \brief Restart acquisition algorithm
00132      */
00133     void reset() override;
00134
00135     /*!
00136      * \brief If state = 1, it forces the block to start acquiring from the first sample
00137      */
00138     void set_state(int state) override;
00139
00140     /*!
00141      * \brief Stop running acquisition
00142      */
00143     void stop_acquisition() override;
00144
00145     void set_resampler_latency(uint32_t latency_samples __attribute__((unused))) override {};
00146
00147 private:
00148     pcps_acquisition_fine_doppler_cc_sptr acquisition_cc_;
00149     std::weak_ptr<ChannelFsm> channel_fsm_;
00150     std::vector<std::complex<float>> code_;
00151     std::string item_type_;
00152     std::string dump_filename_;
00153     std::string role_;
00154     Gnss_Synchro* gnss_synchro_;
00155     int64_t fs_in_;
00156     size_t item_size_;
00157     float threshold_;
00158     int doppler_max_;
00159     int max_dwells_;
00160     unsigned int vector_length_;
00161     unsigned int channel_;
00162     unsigned int doppler_step_;
00163     unsigned int sampled_ms_;
00164     unsigned int in_streams_;
00165     unsigned int out_streams_;
00166     bool dump_;
00167 };
00168
00169
00170 /** \} */
00171 /** \} */
00172 #endif // GNSS_SDR_GPS_L1_CA_PCPS_ACQUISITION_FINE_DOPPLER_H

```

## 11.37 gps\_l1\_ca\_pcps\_acquisition\_fpga.h File Reference

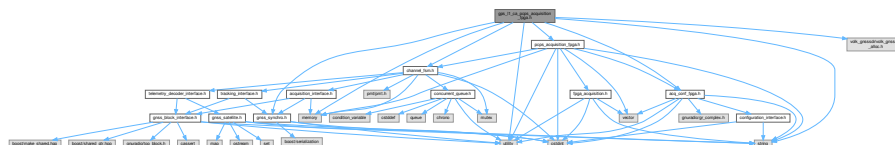
Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals for the FPGA.

```

#include "acq_conf_fpga.h"
#include "channel_fsm.h"
#include "gnss_synchro.h"
#include "pcps_acquisition_fpga.h"
#include <volk_gnssssdr/volk_gnssssdr_alloc.h>
#include <memory>
#include <string>
#include <utility>

```

Include dependency graph for `gps_l1_ca_pcps_acquisition_fpga.h`:



**Classes**

- class [GpsL1CaPcpsAcquisitionFpga](#)

*This class adapts a PCPS acquisition block off-loaded on an FPGA to an [AcquisitionInterface](#) for GPS L1 C/A signals.*

**11.37.1 Detailed Description**

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals for the FPGA.

**Authors**

- Marc Majoral, 2019. mmajoral(at)cttc.es
- Javier Arribas, 2019. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gps\\_l1\\_ca\\_pcps\\_acquisition\\_fpga.h](#).

---

**11.38 gps\_l1\_ca\_pcps\_acquisition\_fpga.h**

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file gps_l1_ca_pcps_acquisition_fpga.h
00003  * \brief Adapts a PCPS acquisition block to an AcquisitionInterface
00004  * for GPS L1 C/A signals for the FPGA
00005  * \authors <ul>
00006  * <li> Marc Majoral, 2019. mmajoral(at)cttc.es
00007  * <li> Javier Arribas, 2019. jarribas(at)cttc.es
00008  * </ul>
00009  *
00010  * -----
00011  *
00012  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00013  * This file is part of GNSS-SDR.
00014  *
00015  * Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors)
00016  * SPDX-License-Identifier: GPL-3.0-or-later
00017  *
00018  * -----
00019  */
00020
00021 #ifndef GNSS_SDR_GPS_L1_CA_PCPS_ACQUISITION_FPGA_H
00022 #define GNSS_SDR_GPS_L1_CA_PCPS_ACQUISITION_FPGA_H
00023
00024 #include "acq_conf_fpga.h"
00025 #include "channel_fsm.h"
00026 #include "gnss_synchro.h"
00027 #include "pcps_acquisition_fpga.h"
00028 #include <volk_gnssdr/volk_gnssdr_alloc.h>
00029 #include <memory>
00030 #include <string>
00031 #include <utility>
00032
00033 /** \addtogroup Acquisition
00034  * \{ */
00035 /** \addtogroup Acq_adapters
00036  * \{ */
00037
00038
00039 class ConfigurationInterface;
00040
00041 /*!
00042 * \brief This class adapts a PCPS acquisition block off-loaded on an FPGA
00043 * to an AcquisitionInterface for GPS L1 C/A signals
00044 */
00045 class GpsL1CaPcpsAcquisitionFpga : public AcquisitionInterface
00046 {
00047 public:
00048     /*!
00049     * \brief Constructor
00050     */
00051     GpsL1CaPcpsAcquisitionFpga(const ConfigurationInterface* configuration,
00052                                const std::string& role,
00053                                unsigned int in_streams,
00054                                unsigned int out_streams);
00055
00056     /*!
00057     * \brief Destructor

```

```

00058     */
00059 ~GpsLlCaPcpsAcquisitionFpga() = default;
00060
00061     /*!
00062     * \brief Role
00063     */
00064     inline std::string role() override
00065     {
00066         return role_;
00067     }
00068
00069     /*!
00070     * \brief Returns "GPS_L1_CA_PCPS_Acquisition_FPGA"
00071     */
00072     inline std::string implementation() override
00073     {
00074         return "GPS_L1_CA_PCPS_Acquisition_FPGA";
00075     }
00076
00077     /*!
00078     * \brief Returns size of lv_l6sc_t
00079     */
00080     inline size_t item_size() override
00081     {
00082         return sizeof(int16_t);
00083     }
00084
00085     /*!
00086     * \brief Connect
00087     */
00088     void connect(gr::top_block_sptr top_block) override;
00089
00090     /*!
00091     * \brief Disconnect
00092     */
00093     void disconnect(gr::top_block_sptr top_block) override;
00094
00095     /*!
00096     * \brief Get left block
00097     */
00098     gr::basic_block_sptr get_left_block() override;
00099
00100     /*!
00101     * \brief Get right block
00102     */
00103     gr::basic_block_sptr get_right_block() override;
00104
00105     /*!
00106     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00107     * to efficiently exchange synchronization data between acquisition and
00108     * tracking blocks
00109     */
00110     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00111
00112     /*!
00113     * \brief Set acquisition channel unique ID
00114     */
00115     inline void set_channel(unsigned int channel) override
00116     {
00117         channel_ = channel;
00118         acquisition_fpga->set_channel(channel_);
00119     }
00120
00121     /*!
00122     * \brief Set channel fsm associated to this acquisition instance
00123     */
00124     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm) override
00125     {
00126         channel_fsm_ = std::move(channel_fsm);
00127         acquisition_fpga->set_channel_fsm(channel_fsm_);
00128     }
00129
00130     /*!
00131     * \brief Set statistics threshold of PCPS algorithm
00132     */
00133     std::string item_type_;
00134
00135     void set_threshold(float threshold) override;
00136
00137     /*!
00138     * \brief Set maximum Doppler off grid search
00139     */
00140     void set_doppler_max(unsigned int doppler_max) override;
00141
00142     /*!
00143     * \brief Set Doppler steps for the grid search
00144     */

```



```

00145     void set_doppler_step(unsigned int doppler_step) override;
00146
00147     /*!
00148      * \brief Set Doppler center for the grid search
00149      */
00150     void set_doppler_center(int doppler_center) override;
00151
00152     /*!
00153      * \brief Initializes acquisition algorithm.
00154      */
00155     void init() override;
00156
00157     /*!
00158      * \brief Sets local code for GPS L1/CA PCPS acquisition algorithm.
00159      */
00160     void set_local_code() override;
00161
00162     /*!
00163      * \brief Returns the maximum peak of grid search
00164      */
00165     signed int mag() override;
00166
00167     /*!
00168      * \brief Restart acquisition algorithm
00169      */
00170     void reset() override;
00171
00172     /*!
00173      * \brief If state = 1, it forces the block to start acquiring from the first sample
00174      */
00175     void set_state(int state) override;
00176
00177     /*!
00178      * \brief Stop running acquisition
00179      */
00180     void stop_acquisition() override;
00181
00182     /*!
00183      * \brief Set Resampler Latency
00184      */
00185     void set_resampler_latency(uint32_t latency_samples __attribute__((unused))) override {};
00186
00187 private:
00188     static const uint32_t ACQ_BUFF_0 = 0; // FPGA Acquisition IP buffer containing L1/E1
00189     frequency band samples by default.
00190     static const uint32_t DEFAULT_FPGA_BLK_EXP = 10; // default block exponent
00191     static const uint32_t NUM_PRNs = 32;
00192     static const uint32_t QUANT_BITS_LOCAL_CODE = 16;
00193     static const uint32_t SELECT_LSBITS = 0x0000FFFF; // Select the 10 LSBits out of a 20-bit
00194     word
00195     static const uint32_t SELECT_MSBITS = 0xFFFF0000; // Select the 10 MSbits out of a 20-bit
00196     word
00197     static const uint32_t SELECT_ALL_CODE_BITS = 0xFFFFFFFF; // Select a 20 bit word
00198     static const uint32_t SHL_CODE_BITS = 65536; // shift left by 10 bits
00199
00200     void generate_gps_l1_ca_prn_codes();
00201
00202     pcps_acquisition_fpga_sptr acquisition_fpga_;
00203     std::weak_ptr<ChannelFsm> channel_fsm_;
00204     volk_gnssdr::vector<uint32_t> d_all_fft_codes; // memory that contains all the code ffts
00205     Gnss_Synchro* gnss_synchro_;
00206     Acq_Conf_Fpga acq_parameters_;
00207     std::string role_;
00208     int32_t doppler_center_;
00209     uint32_t channel_;
00210     uint32_t doppler_max_;
00211     uint32_t doppler_step_;
00212     unsigned int in_streams_;
00213     unsigned int out_streams_;
00214 };
00215
00216 /** \} */
00217 /** \} */
00218 #endif // GNSS_SDR_GPS_L1_CA_PCPS_ACQUISITION_FPGA_H

```

## 11.39 gps\_l1\_ca\_pcps\_assisted\_acquisition.h File Reference

Adapts a PCPS Assisted acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.

```

#include "channel_fsm.h"
#include "gnss_synchro.h"
#include "pcps_assisted_acquisition_cc.h"

```



```

00036
00037 class ConfigurationInterface;
00038
00039 /*!
00040 * \brief This class adapts a PCPS acquisition block to an AcquisitionInterface
00041 * for GPS L1 C/A signals
00042 */
00043 class GpsL1CaPcpsAssistedAcquisition : public AcquisitionInterface
00044 {
00045 public:
00046     GpsL1CaPcpsAssistedAcquisition(
00047         const ConfigurationInterface* configuration,
00048         const std::string& role,
00049         unsigned int in_streams,
00050         unsigned int out_streams);
00051
00052     ~GpsL1CaPcpsAssistedAcquisition() = default;
00053
00054     inline std::string role() override
00055     {
00056         return role_;
00057     }
00058
00059     /*!
00060     * \brief Returns "GPS_L1_CA_PCPS_Assisted_Acquisition"
00061     */
00062     inline std::string implementation() override
00063     {
00064         return "GPS_L1_CA_PCPS_Assisted_Acquisition";
00065     }
00066
00067     inline size_t item_size() override
00068     {
00069         return item_size_;
00070     }
00071
00072     void connect(gr::top_block_sptr top_block) override;
00073     void disconnect(gr::top_block_sptr top_block) override;
00074     gr::basic_block_sptr get_left_block() override;
00075     gr::basic_block_sptr get_right_block() override;
00076
00077     /*!
00078     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00079     * to efficiently exchange synchronization data between acquisition and
00080     * tracking blocks
00081     */
00082     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00083
00084     /*!
00085     * \brief Set acquisition channel unique ID
00086     */
00087     inline void set_channel(unsigned int channel) override
00088     {
00089         channel_ = channel;
00090         acquisition_cc->set_channel(channel_);
00091     }
00092
00093     /*!
00094     * \brief Set channel fsm associated to this acquisition instance
00095     */
00096     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm) override
00097     {
00098         channel_fsm_ = std::move(channel_fsm);
00099         acquisition_cc->set_channel_fsm(channel_fsm_);
00100     }
00101
00102     /*!
00103     * \brief Set statistics threshold of PCPS algorithm
00104     */
00105     void set_threshold(float threshold) override;
00106
00107     /*!
00108     * \brief Set maximum Doppler off grid search
00109     */
00110     void set_doppler_max(unsigned int doppler_max) override;
00111
00112     /*!
00113     * \brief Set Doppler steps for the grid search
00114     */
00115     void set_doppler_step(unsigned int doppler_step) override;
00116
00117     /*!
00118     * \brief Initializes acquisition algorithm.
00119     */
00120     void init() override;
00121
00122     void set_local_code() override;

```



**Classes**

- class [GpsL1CaPcpsOpenClAcquisition](#)

*This class adapts an OpenCL PCPS acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.*

**11.41.1 Detailed Description**

Adapts an OpenCL PCPS acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.

**Author**

Marc Molina, 2013. marc.molina.pena(at)gmail.com

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [gps\\_l1\\_ca\\_pcps\\_openc1\\_acquisition.h](#).

---

**11.42 gps\_l1\_ca\_pcps\_openc1\_acquisition.h**

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file gps_l1_ca_pcps_openc1_acquisition.h
00003  * \brief Adapts an OpenCL PCPS acquisition block to an
00004  * \brief AcquisitionInterface for GPS L1 C/A signals
00005  * \author Marc Molina, 2013. marc.molina.pena(at)gmail.com
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_GPS_L1_CA_PCPS_OPENC1_ACQUISITION_H
00019 #define GNSS_SDR_GPS_L1_CA_PCPS_OPENC1_ACQUISITION_H
00020
00021 #include "channel_fsm.h"
00022 #include "gnss_synchro.h"
00023 #include "pcps_openc1_acquisition_cc.h"
00024 #include <gnuradio/blocks/stream_to_vector.h>
00025 #include <memory>
00026 #include <string>
00027 #include <utility>
00028 #include <vector>
00029
00030 /** \addtogroup Acquisition
00031  * \{ */
00032 /** \addtogroup Acq_adapters
00033  * \{ */
00034
00035
00036 class ConfigurationInterface;
00037
00038 /*!
00039 * \brief This class adapts an OpenCL PCPS acquisition block to an
00040 * \brief AcquisitionInterface for GPS L1 C/A signals
00041 */
00042 class GpsL1CaPcpsOpenClAcquisition : public AcquisitionInterface
00043 {
00044 public:
00045     GpsL1CaPcpsOpenClAcquisition(const ConfigurationInterface* configuration,
00046         const std::string& role,
00047         unsigned int in_streams,
00048         unsigned int out_streams);
00049
00050     ~GpsL1CaPcpsOpenClAcquisition() = default;
00051
00052     inline std::string role() override
00053     {
00054         return role_;
00055     }
00056
00057     /*!
00058     * \brief Returns "GPS_L1_CA_PCPS_OpenCl_Acquisition"
00059     */

```

```

00060     inline std::string implementation() override
00061     {
00062         return "GPS_L1_CA_PCPS_OpenCl_Acquisition";
00063     }
00064
00065     inline size_t item_size() override
00066     {
00067         return item_size_;
00068     }
00069
00070     void connect(gr::top_block_sptr top_block) override;
00071     void disconnect(gr::top_block_sptr top_block) override;
00072     gr::basic_block_sptr get_left_block() override;
00073     gr::basic_block_sptr get_right_block() override;
00074
00075     /*!
00076      * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00077      * to efficiently exchange synchronization data between acquisition and
00078      * tracking blocks
00079      */
00080     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00081
00082     /*!
00083      * \brief Set acquisition channel unique ID
00084      */
00085     inline void set_channel(unsigned int channel) override
00086     {
00087         channel_ = channel;
00088         acquisition_cc->set_channel(channel_);
00089     }
00090
00091     /*!
00092      * \brief Set channel fsm associated to this acquisition instance
00093      */
00094     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm) override
00095     {
00096         channel_fsm_ = std::move(channel_fsm);
00097         acquisition_cc->set_channel_fsm(channel_fsm_);
00098     }
00099
00100     /*!
00101      * \brief Set statistics threshold of PCPS algorithm
00102      */
00103     void set_threshold(float threshold) override;
00104
00105     /*!
00106      * \brief Set maximum Doppler off grid search
00107      */
00108     void set_doppler_max(unsigned int doppler_max) override;
00109
00110     /*!
00111      * \brief Set Doppler steps for the grid search
00112      */
00113     void set_doppler_step(unsigned int doppler_step) override;
00114
00115     /*!
00116      * \brief Initializes acquisition algorithm.
00117      */
00118     void init() override;
00119
00120     /*!
00121      * \brief Sets local code for GPS L1/CA PCPS acquisition algorithm.
00122      */
00123     void set_local_code() override;
00124
00125     /*!
00126      * \brief Returns the maximum peak of grid search
00127      */
00128     signed int mag() override;
00129
00130     /*!
00131      * \brief Restart acquisition algorithm
00132      */
00133     void reset() override;
00134     void set_state(int state __attribute__((unused))) override {};
00135
00136     /*!
00137      * \brief Stop running acquisition
00138      */
00139     void stop_acquisition() override;
00140
00141     void set_resampler_latency(uint32_t latency_samples __attribute__((unused))) override {};
00142
00143     inline bool openc1_ready() const
00144     {
00145         bool ready = this->acquisition_cc->openc1_ready();
00146         return ready;

```

```

00147     }
00148
00149 private:
00150     float calculate_threshold(float pfa) const;
00151     const ConfigurationInterface* configuration_;
00152     pcps_openc1_acquisition_cc_sptr acquisition_cc_;
00153     gr::blocks::stream_to_vector::sptr stream_to_vector_;
00154     std::weak_ptr<ChannelFsm> channel_fsm_;
00155     std::vector<std::complex<float>> code_;
00156     Gnss_Synchro* gnss_synchro_;
00157
00158     std::string item_type_;
00159     std::string dump_filename_;
00160     std::string role_;
00161
00162     int64_t fs_in_;
00163     size_t item_size_;
00164
00165     float threshold_;
00166
00167     unsigned int vector_length_;
00168     unsigned int code_length_;
00169     unsigned int channel_;
00170     unsigned int doppler_max_;
00171     unsigned int doppler_step_;
00172     unsigned int sampled_ms_;
00173     unsigned int max_dwells_;
00174     unsigned int in_streams_;
00175     unsigned int out_streams_;
00176     bool bit_transition_flag_;
00177     bool dump_;
00178 };
00179
00180
00181 /** \} */
00182 /** \} */
00183 #endif // GNSS_SDR_GPS_L1_CA_PCPS_OPENC1_ACQUISITION_H

```

## 11.43 gps\_l1\_ca\_pcps\_quicksync\_acquisition.h File Reference

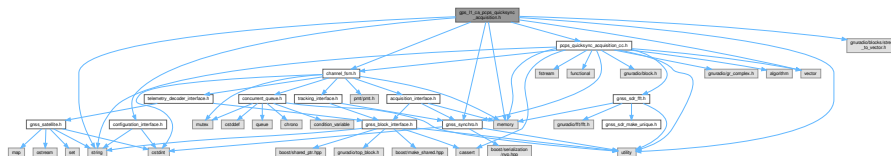
Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals implementing the QuickSync Algorithm.

```

#include "channel_fsm.h"
#include "configuration_interface.h"
#include "gnss_synchro.h"
#include "pcps_quicksync_acquisition_cc.h"
#include <gnuradio/blocks/stream_to_vector.h>
#include <memory>
#include <string>
#include <utility>
#include <vector>

```

Include dependency graph for `gps_l1_ca_pcps_quicksync_acquisition.h`:



### Classes

- class [GpsL1CaPcpsQuickSyncAcquisition](#)

*This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.*

### 11.43.1 Detailed Description

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals implementing the QuickSync Algorithm.

**Date**

June, 2014

**Author**Damian Miralles Sanchez. [dmiralles2009@gmail.com](mailto:dmiralles2009@gmail.com)


---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gps\\_l1\\_ca\\_pcps\\_quicksync\\_acquisition.h](#).

---

**11.44 gps\_l1\_ca\_pcps\_quicksync\_acquisition.h**

[Go to the documentation of this file.](#)

```

00001  /*!
00002   * \file gps_l1_ca_pcps_quicksync_acquisition.h
00003   * \brief Adapts a PCPS acquisition block to an AcquisitionInterface for GPS L1 C/A signals
00004   *         implementing the QuickSync Algorithm.
00005   * \date June, 2014
00006   * \author Damian Miralles Sanchez. dmiralles2009@gmail.com
00007   *
00008   * -----
00009   *
00010   * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011   * This file is part of GNSS-SDR.
00012   *
00013   * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014   * SPDX-License-Identifier: GPL-3.0-or-later
00015   *
00016   * -----
00017   */
00018
00019 #ifndef GNSS_SDR_GPS_L1_CA_PCPS_QUICKSYNC_ACQUISITION_H
00020 #define GNSS_SDR_GPS_L1_CA_PCPS_QUICKSYNC_ACQUISITION_H
00021
00022 #include "channel_fsm.h"
00023 #include "configuration_interface.h"
00024 #include "gnss_synchro.h"
00025 #include "pcps_quicksync_acquisition_cc.h"
00026 #include <gnuradio/blocks/stream_to_vector.h>
00027 #include <memory>
00028 #include <string>
00029 #include <utility>
00030 #include <vector>
00031
00032 /** \addtogroup Acquisition
00033   * \{ */
00034 /** \addtogroup Acq_adapters
00035   * \{ */
00036
00037 class ConfigurationInterface;
00038
00039 /*!
00040   * \brief This class adapts a PCPS acquisition block to an AcquisitionInterface
00041   *        for GPS L1 C/A signals
00042   */
00043 class GpsL1CaPcpsQuickSyncAcquisition : public AcquisitionInterface
00044 {
00045 public:
00046     GpsL1CaPcpsQuickSyncAcquisition(
00047         const ConfigurationInterface* configuration,
00048         const std::string& role,
00049         unsigned int in_streams,
00050         unsigned int out_streams);
00051
00052     ~GpsL1CaPcpsQuickSyncAcquisition() = default;
00053
00054     inline std::string role() override
00055     {
00056         return role_;
00057     }
00058
00059 /*!
00060   * \brief Returns "GPS_L1_CA_PCPS_QuickSync_Acquisition"
00061   */
00062     inline std::string implementation() override
00063     {
00064         return "GPS_L1_CA_PCPS_QuickSync_Acquisition";
00065     }

```



```

00066     }
00067
00068     inline size_t item_size() override
00069     {
00070         return item_size_;
00071     }
00072
00073     void connect(gr::top_block_sptr top_block) override;
00074     void disconnect(gr::top_block_sptr top_block) override;
00075     gr::basic_block_sptr get_left_block() override;
00076     gr::basic_block_sptr get_right_block() override;
00077
00078     /*!
00079     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00080     * to efficiently exchange synchronization data between acquisition and
00081     * tracking blocks
00082     */
00083     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00084
00085     /*!
00086     * \brief Set acquisition channel unique ID
00087     */
00088     inline void set_channel(unsigned int channel) override
00089     {
00090         channel_ = channel;
00091         acquisition_cc->set_channel(channel_);
00092     }
00093
00094     /*!
00095     * \brief Set channel fsm associated to this acquisition instance
00096     */
00097     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm) override
00098     {
00099         channel_fsm_ = std::move(channel_fsm);
00100         acquisition_cc->set_channel_fsm(channel_fsm_);
00101     }
00102
00103     /*!
00104     * \brief Set statistics threshold of PCPS algorithm
00105     */
00106     void set_threshold(float threshold) override;
00107
00108     /*!
00109     * \brief Set maximum Doppler off grid search
00110     */
00111     void set_doppler_max(unsigned int doppler_max) override;
00112
00113     /*!
00114     * \brief Set Doppler steps for the grid search
00115     */
00116     void set_doppler_step(unsigned int doppler_step) override;
00117
00118     /*!
00119     * \brief Initializes acquisition algorithm.
00120     */
00121     void init() override;
00122
00123     /*!
00124     * \brief Sets local code for GPS L1/CA PCPS acquisition algorithm.
00125     */
00126     void set_local_code() override;
00127
00128     /*!
00129     * \brief Returns the maximum peak of grid search
00130     */
00131     signed int mag() override;
00132
00133     /*!
00134     * \brief Restart acquisition algorithm
00135     */
00136     void reset() override;
00137
00138     /*!
00139     * \brief If state = 1, it forces the block to start acquiring from the first sample
00140     */
00141     void set_state(int state) override;
00142
00143     /*!
00144     * \brief Stop running acquisition
00145     */
00146     void stop_acquisition() override;
00147
00148     void set_resampler_latency(uint32_t latency_samples __attribute__((unused))) override {};
00149
00150 private:
00151     float calculate_threshold(float pfa) const;
00152

```

```

00153     const ConfigurationInterface* configuration_;
00154     pcps_quicksync_acquisition_cc_sptr acquisition_cc_;
00155     std::weak_ptr<ChannelFsm> channel_fsm_;
00156
00157     gr::blocks::stream_to_vector::sptr stream_to_vector_;
00158     std::vector<std::complex<float>> code_;
00159     std::string item_type_;
00160     std::string dump_filename_;
00161     std::string role_;
00162
00163     Gnss_Synchro* gnss_synchro_;
00164
00165     int64_t fs_in_;
00166     size_t item_size_;
00167     float threshold_;
00168     unsigned int vector_length_;
00169     unsigned int code_length_;
00170     unsigned int channel_;
00171     unsigned int doppler_max_;
00172     unsigned int doppler_step_;
00173     unsigned int sampled_ms_;
00174     unsigned int max_dwells_;
00175     unsigned int folding_factor_;
00176     unsigned int in_streams_;
00177     unsigned int out_streams_;
00178     bool bit_transition_flag_;
00179     bool dump_;
00180 };
00181
00182
00183 /** \} */
00184 /** \} */
00185 #endif // GNSS_SDR_GPS_L1_CA_PCPS_QUICKSYNC_ACQUISITION_H

```

## 11.45 gps\_l1\_ca\_pcps\_tong\_acquisition.h File Reference

Adapts a PCPS Tong acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.

```

#include "channel_fsm.h"
#include "configuration_interface.h"
#include "gnss_synchro.h"
#include "pcps_tong_acquisition_cc.h"
#include <gnuradio/blocks/stream_to_vector.h>
#include <memory>
#include <string>
#include <utility>
#include <vector>

```

Include dependency graph for `gps_l1_ca_pcps_tong_acquisition.h`:



### Classes

- class [GpsL1CaPcpsTongAcquisition](#)

*This class adapts a PCPS Tong acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.*

### 11.45.1 Detailed Description

Adapts a PCPS Tong acquisition block to an [AcquisitionInterface](#) for GPS L1 C/A signals.

## Author

Marc Molina, 2013. marc.molina.pena(at)gmail.com

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [gps\\_l1\\_ca\\_pcps\\_tong\\_acquisition.h](#).

## 11.46 gps\_l1\_ca\_pcps\_tong\_acquisition.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002   * \file gps_l1_ca_pcps_tong_acquisition.h
00003   * \brief Adapts a PCPS Tong acquisition block to an AcquisitionInterface for
00004   *        GPS L1 C/A signals
00005   * \author Marc Molina, 2013. marc.molina.pena(at)gmail.com
00006   *
00007   * -----
00008   *
00009   * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010   * This file is part of GNSS-SDR.
00011   *
00012   * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013   * SPDX-License-Identifier: GPL-3.0-or-later
00014   *
00015   * -----
00016   */
00017
00018 #ifndef GNSS_SDR_GPS_L1_CA_TONG_ACQUISITION_H
00019 #define GNSS_SDR_GPS_L1_CA_TONG_ACQUISITION_H
00020
00021 #include "channel_fsm.h"
00022 #include "configuration_interface.h"
00023 #include "gnss_synchro.h"
00024 #include "pcps_tong_acquisition_cc.h"
00025 #include <gnuradio/blocks/stream_to_vector.h>
00026 #include <memory>
00027 #include <string>
00028 #include <utility>
00029 #include <vector>
00030
00031 /** \addtogroup Acquisition
00032  * \{ */
00033 /** \addtogroup Acq_adapters
00034  * \{ */
00035
00036
00037 class ConfigurationInterface;
00038
00039 /*!
00040  * \brief This class adapts a PCPS Tong acquisition block to an
00041  *        AcquisitionInterface for GPS L1 C/A signals
00042  */
00043 class GpsL1CaPcpsTongAcquisition : public AcquisitionInterface
00044 {
00045 public:
00046     GpsL1CaPcpsTongAcquisition(const ConfigurationInterface* configuration,
00047                                const std::string& role,
00048                                unsigned int in_streams,
00049                                unsigned int out_streams);
00050
00051     ~GpsL1CaPcpsTongAcquisition() = default;
00052
00053     inline std::string role() override
00054     {
00055         return role_;
00056     }
00057
00058     /*!
00059     * \brief Returns "GPS_L1_CA_PCPS_Tong_Acquisition"
00060     */
00061     inline std::string implementation() override
00062     {
00063         return "GPS_L1_CA_PCPS_Tong_Acquisition";
00064     }
00065
00066     inline size_t item_size() override
00067     {
00068         return item_size_;
00069     }
00070
00071     void connect(gr::top_block_sptr top_block) override;
00072     void disconnect(gr::top_block_sptr top_block) override;

```

```

00073     gr::basic_block_sptr get_left_block() override;
00074     gr::basic_block_sptr get_right_block() override;
00075
00076     /*!
00077      * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00078      * to efficiently exchange synchronization data between acquisition and
00079      * tracking blocks
00080      */
00081     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00082
00083     /*!
00084      * \brief Set acquisition channel unique ID
00085      */
00086     inline void set_channel(unsigned int channel) override
00087     {
00088         channel_ = channel;
00089         acquisition_cc->set_channel(channel_);
00090     }
00091
00092     /*!
00093      * \brief Set channel fsm associated to this acquisition instance
00094      */
00095     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm) override
00096     {
00097         channel_fsm_ = std::move(channel_fsm);
00098         acquisition_cc->set_channel_fsm(channel_fsm_);
00099     }
00100
00101     /*!
00102      * \brief Set statistics threshold of TONG algorithm
00103      */
00104     void set_threshold(float threshold) override;
00105
00106     /*!
00107      * \brief Set maximum Doppler off grid search
00108      */
00109     void set_doppler_max(unsigned int doppler_max) override;
00110
00111     /*!
00112      * \brief Set Doppler steps for the grid search
00113      */
00114     void set_doppler_step(unsigned int doppler_step) override;
00115
00116     /*!
00117      * \brief Initializes acquisition algorithm.
00118      */
00119     void init() override;
00120
00121     /*!
00122      * \brief Sets local code for GPS L1/CA TONG acquisition algorithm.
00123      */
00124     void set_local_code() override;
00125
00126     /*!
00127      * \brief Returns the maximum peak of grid search
00128      */
00129     signed int mag() override;
00130
00131     /*!
00132      * \brief Restart acquisition algorithm
00133      */
00134     void reset() override;
00135
00136     /*!
00137      * \brief If state = 1, it forces the block to start acquiring from the first sample
00138      */
00139     void set_state(int state) override;
00140
00141     /*!
00142      * \brief Stop running acquisition
00143      */
00144     void stop_acquisition() override;
00145
00146     void set_resampler_latency(uint32_t latency_samples __attribute__((unused))) override {};
00147
00148 private:
00149     float calculate_threshold(float pfa) const;
00150
00151     const ConfigurationInterface* configuration_;
00152     pcps_tong_acquisition_cc_sptr acquisition_cc_;
00153     gr::blocks::stream_to_vector::sptr stream_to_vector_;
00154     std::weak_ptr<ChannelFsm> channel_fsm_;
00155     std::vector<std::complex<float>> code_;
00156     Gnss_Synchro* gnss_synchro_;
00157     std::string item_type_;
00158     std::string dump_filename_;
00159     std::string role_;

```

```

00160     int64_t fs_in_;
00161     size_t item_size_;
00162     float threshold_;
00163     unsigned int vector_length_;
00164     unsigned int code_length_;
00165     unsigned int channel_;
00166     unsigned int doppler_max_;
00167     unsigned int doppler_step_;
00168     unsigned int sampled_ms_;
00169     unsigned int tong_init_val_;
00170     unsigned int tong_max_val_;
00171     unsigned int tong_max_dwells_;
00172     unsigned int in_streams_;
00173     unsigned int out_streams_;
00174     bool dump_;
00175 };
00176
00177
00178 /** \} */
00179 /** \} */
00180 #endif // GNSS_SDR_GPS_L1_CA_TONG_ACQUISITION_H

```

## 11.47 gps\_l2\_m\_pcps\_acquisition.h File Reference

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L2 M signals.

```

#include "channel_fsm.h"
#include "complex_byte_to_float_x2.h"
#include "gnss_synchro.h"
#include "pcps_acquisition.h"
#include <gnuradio/blocks/float_to_complex.h>
#include <volk_gnssssdr/volk_gnssssdr_alloc.h>
#include <memory>
#include <string>
#include <utility>
#include <vector>

```

Include dependency graph for `gps_l2_m_pcps_acquisition.h`:



### Classes

- class [GpsL2MPcpsAcquisition](#)

*This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L2 M signals.*

### 11.47.1 Detailed Description

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L2 M signals.

#### Authors

- Javier Arribas, 2015. jarribas(at)cttc.es

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [gps\\_l2\\_m\\_pcps\\_acquisition.h](#).

## 11.48 gps\_l2\_m\_pcps\_acquisition.h

[Go to the documentation of this file.](#)

```
00001 /*!
```

```

00002 * \file gps_l2_m_pcps_acquisition.h
00003 * \brief Adapts a PCPS acquisition block to an AcquisitionInterface for
00004 * GPS L2 M signals
00005 * \authors <ul>
00006 *     <li> Javier Arribas, 2015. jarribas(at)cttc.es
00007 *     </li>
00008 *
00009 * -----
00010 *
00011 * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00012 * This file is part of GNSS-SDR.
00013 *
00014 * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00015 * SPDX-License-Identifier: GPL-3.0-or-later
00016 *
00017 * -----
00018 */
00019
00020 #ifndef GNSS_SDR_GPS_L2_M_PCPS_ACQUISITION_H
00021 #define GNSS_SDR_GPS_L2_M_PCPS_ACQUISITION_H
00022
00023 #include "channel_fsm.h"
00024 #include "complex_byte_to_float_x2.h"
00025 #include "gnss_synchro.h"
00026 #include "pcps_acquisition.h"
00027 #include <gnuradio/blocks/float_to_complex.h>
00028 #include <volk_gnssssdr/volk_gnssssdr_alloc.h>
00029 #include <memory>
00030 #include <string>
00031 #include <utility>
00032 #include <vector>
00033
00034 /** \addtogroup Acquisition
00035 *  \{ */
00036 /** \addtogroup Acq_adapters
00037 *  \{ */
00038
00039
00040 class ConfigurationInterface;
00041
00042 /*!
00043 * \brief This class adapts a PCPS acquisition block to an AcquisitionInterface
00044 * for GPS L2 M signals
00045 */
00046 class GpsL2MPcpsAcquisition : public AcquisitionInterface
00047 {
00048 public:
00049     GpsL2MPcpsAcquisition(
00050         const ConfigurationInterface* configuration,
00051         const std::string& role,
00052         unsigned int in_streams,
00053         unsigned int out_streams);
00054
00055     ~GpsL2MPcpsAcquisition() = default;
00056
00057     inline std::string role() override
00058     {
00059         return role_;
00060     }
00061
00062     /*!
00063      * \brief Returns "GPS_L2_M_PCPS_Acquisition"
00064      */
00065     inline std::string implementation() override
00066     {
00067         return "GPS_L2_M_PCPS_Acquisition";
00068     }
00069
00070     inline size_t item_size() override
00071     {
00072         return item_size_;
00073     }
00074
00075     void connect(gr::top_block_sptr top_block) override;
00076     void disconnect(gr::top_block_sptr top_block) override;
00077     gr::basic_block_sptr get_left_block() override;
00078     gr::basic_block_sptr get_right_block() override;
00079
00080     /*!
00081      * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00082      * to efficiently exchange synchronization data between acquisition and
00083      * tracking blocks
00084      */
00085     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00086
00087     /*!
00088      * \brief Set acquisition channel unique ID

```

```

00089     */
00090     inline void set_channel(unsigned int channel) override
00091     {
00092         channel_ = channel;
00093         acquisition_>set_channel(channel_);
00094     }
00095
00096     /*!
00097     * \brief Set channel fsm associated to this acquisition instance
00098     */
00099     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm) override
00100     {
00101         channel_fsm_ = std::move(channel_fsm);
00102         acquisition_>set_channel_fsm(channel_fsm_);
00103     }
00104
00105     /*!
00106     * \brief Set statistics threshold of PCPS algorithm
00107     */
00108     void set_threshold(float threshold) override;
00109
00110     /*!
00111     * \brief Set maximum Doppler off grid search
00112     */
00113     void set_doppler_max(unsigned int doppler_max) override;
00114
00115     /*!
00116     * \brief Set Doppler steps for the grid search
00117     */
00118     void set_doppler_step(unsigned int doppler_step) override;
00119
00120     /*!
00121     * \brief Set Doppler center for the grid search
00122     */
00123     void set_doppler_center(int doppler_center) override;
00124
00125     /*!
00126     * \brief Initializes acquisition algorithm.
00127     */
00128     void init() override;
00129
00130     /*!
00131     * \brief Sets local code for GPS L2/M PCPS acquisition algorithm.
00132     */
00133     void set_local_code() override;
00134
00135     /*!
00136     * \brief Returns the maximum peak of grid search
00137     */
00138     signed int mag() override;
00139
00140     /*!
00141     * \brief Restart acquisition algorithm
00142     */
00143     void reset() override;
00144
00145     /*!
00146     * \brief If state = 1, it forces the block to start acquiring from the first sample
00147     */
00148     void set_state(int state) override;
00149
00150     /*!
00151     * \brief Stop running acquisition
00152     */
00153     void stop_acquisition() override;
00154
00155     /*!
00156     * \brief Sets the resampler latency to account it in the acquisition code delay estimation
00157     */
00158     void set_resampler_latency(uint32_t latency_samples) override;
00159
00160 private:
00161     pcps_acquisition_sptr acquisition_;
00162     volk_gnss_sdr::vector<std::complex<float>> code_;
00163     gr::blocks::float_to_complex::sptr float_to_complex_;
00164     complex_byte_to_float_x2_sptr cbyte_to_float_x2_;
00165     std::weak_ptr<ChannelFsm> channel_fsm_;
00166     Gnss_Synchro* gnss_synchro_;
00167     Acq_Conf acq_parameters_;
00168     std::string item_type_;
00169     std::string dump_filename_;
00170     std::string role_;
00171     size_t item_size_;
00172     int64_t fs_in_;
00173     float threshold_;
00174     int doppler_center_;
00175     unsigned int vector_length_;

```

```

00176     unsigned int code_length_;
00177     unsigned int channel_;
00178     unsigned int doppler_max_;
00179     unsigned int doppler_step_;
00180     unsigned int in_streams_;
00181     unsigned int out_streams_;
00182     unsigned int num_codes_;
00183 };
00184
00185
00186 /** \} */
00187 /** \} */
00188 #endif // GNSS_SDR_GPS_L2_M_PCPS_ACQUISITION_H

```

## 11.49 gps\_l2\_m\_pcps\_acquisition\_fpga.h File Reference

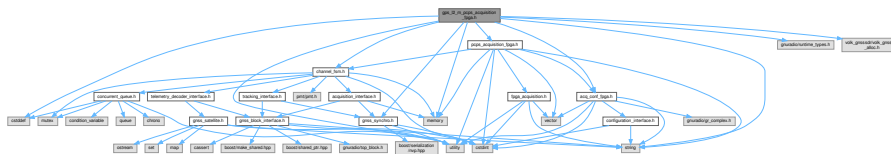
Adapts an FPGA-offloaded PCPS acquisition block to an [AcquisitionInterface](#) for GPS L2 M signals.

```

#include "acq_conf_fpga.h"
#include "channel_fsm.h"
#include "gnss_synchro.h"
#include "pcps_acquisition_fpga.h"
#include <gnuradio/runtime_types.h>
#include <volk_gnssssdr/volk_gnssssdr_alloc.h>
#include <cstdint>
#include <memory>
#include <string>
#include <utility>

```

Include dependency graph for `gps_l2_m_pcps_acquisition_fpga.h`:



### Classes

- class [GpsL2MPcpsAcquisitionFpga](#)

*This class adapts a PCPS acquisition block off-loaded on an FPGA to an [AcquisitionInterface](#) for GPS L2 M signals.*

### 11.49.1 Detailed Description

Adapts an FPGA-offloaded PCPS acquisition block to an [AcquisitionInterface](#) for GPS L2 M signals.

#### Authors

- Javier Arribas, 2019. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gps\\_l2\\_m\\_pcps\\_acquisition\\_fpga.h](#).

---

## 11.50 gps\_l2\_m\_pcps\_acquisition\_fpga.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file gps_l2_m_pcps_acquisition_fpga.h
00003  * \brief Adapts an FPGA-offloaded PCPS acquisition block
00004  * to an AcquisitionInterface for GPS L2 M signals
00005  * \authors <ul>
00006  * <li> Javier Arribas, 2019. jarribas(at)cttc.es
00007  * </li>
00008  *

```



```

00009  * -----
00010  *
00011  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00012  * This file is part of GNSS-SDR.
00013  *
00014  * Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors)
00015  * SPDX-License-Identifier: GPL-3.0-or-later
00016  *
00017  * -----
00018  */
00019
00020 #ifndef GNSS_SDR_GPS_L2_M_PCPS_ACQUISITION_FPGA_H
00021 #define GNSS_SDR_GPS_L2_M_PCPS_ACQUISITION_FPGA_H
00022
00023 #include "acq_conf_fpga.h"
00024 #include "channel_fsm.h"
00025 #include "gnss_synchro.h"
00026 #include "pcps_acquisition_fpga.h"
00027 #include <gnuradio/runtime_types.h> // for basic_block_sptr, top_block_sptr
00028 #include <volk_gnssssdr/volk_gnssssdr_alloc.h>
00029 #include <cstdint> // for size_t
00030 #include <memory> // for weak_ptr
00031 #include <string> // for string
00032 #include <utility>
00033
00034 /** \addtogroup Acquisition
00035  * \{ */
00036 /** \addtogroup Acq_adapters
00037  * \{ */
00038
00039
00040 class ConfigurationInterface;
00041
00042 /*!
00043  * \brief This class adapts a PCPS acquisition block off-loaded on an FPGA
00044  * to an AcquisitionInterface for GPS L2 M signals
00045  */
00046 class GpsL2MPcpsAcquisitionFpga : public AcquisitionInterface
00047 {
00048 public:
00049     GpsL2MPcpsAcquisitionFpga(
00050         const ConfigurationInterface* configuration,
00051         const std::string& role,
00052         unsigned int in_streams,
00053         unsigned int out_streams);
00054
00055     ~GpsL2MPcpsAcquisitionFpga() = default;
00056
00057     inline std::string role() override
00058     {
00059         return role_;
00060     }
00061
00062     /*!
00063     * \brief Returns "GPS_L2_M_PCPS_Acquisition_FPGA"
00064     */
00065     inline std::string implementation() override
00066     {
00067         return "GPS_L2_M_PCPS_Acquisition_FPGA";
00068     }
00069
00070     inline size_t item_size() override
00071     {
00072         return sizeof(float);
00073     }
00074
00075     void connect(gr::top_block_sptr top_block) override;
00076     void disconnect(gr::top_block_sptr top_block) override;
00077     gr::basic_block_sptr get_left_block() override;
00078     gr::basic_block_sptr get_right_block() override;
00079
00080     /*!
00081     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00082     * to efficiently exchange synchronization data between acquisition and
00083     * tracking blocks
00084     */
00085     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00086
00087     /*!
00088     * \brief Set acquisition channel unique ID
00089     */
00090     inline void set_channel(unsigned int channel) override
00091     {
00092         channel_ = channel;
00093         acquisition_fpga->set_channel(channel_);
00094     }
00095

```

```

00096     /*!
00097     * \brief Set channel fsm associated to this acquisition instance
00098     */
00099     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm) override
00100     {
00101         channel_fsm_ = std::move(channel_fsm);
00102         acquisition_fpga->set_channel_fsm(channel_fsm_);
00103     }
00104
00105     /*!
00106     * \brief Set statistics threshold of PCPS algorithm
00107     */
00108     void set_threshold(float threshold) override;
00109
00110     /*!
00111     * \brief Set maximum Doppler off grid search
00112     */
00113     void set_doppler_max(unsigned int doppler_max) override;
00114
00115     /*!
00116     * \brief Set Doppler steps for the grid search
00117     */
00118     void set_doppler_step(unsigned int doppler_step) override;
00119
00120     /*!
00121     * \brief Initializes acquisition algorithm.
00122     */
00123     void init() override;
00124
00125     /*!
00126     * \brief Sets local code for GPS L2/M PCPS acquisition algorithm.
00127     */
00128     void set_local_code() override;
00129
00130     /*!
00131     * \brief Returns the maximum peak of grid search
00132     */
00133     signed int mag() override;
00134
00135     /*!
00136     * \brief Restart acquisition algorithm
00137     */
00138     void reset() override;
00139
00140     /*!
00141     * \brief If state = 1, it forces the block to start acquiring from the first sample
00142     */
00143     void set_state(int state) override;
00144
00145     /*!
00146     * \brief Stop running acquisition
00147     */
00148     void stop_acquisition() override;
00149
00150     void set_resampler_latency(uint32_t latency_samples __attribute__((unused))) override {};
00151
00152 private:
00153     static const uint32_t ACQ_BUFF_1 = 1; // FPGA Acquisition IP buffer containing L2 or
00154     L5/E5 frequency band samples by default.
00155     static const uint32_t DEFAULT_FPGA_BLK_EXP = 13; // default block exponent
00156     static const uint32_t NUM_PRNs = 32;
00157     static const uint32_t QUANT_BITS_LOCAL_CODE = 16;
00158     static const uint32_t SELECT_LSBITS = 0x0000FFFF; // Select the 10 LSbits out of a 20-bit
00159     word
00160     static const uint32_t SELECT_MSBITS = 0xFFFF0000; // Select the 10 MSbits out of a 20-bit
00161     word
00162     static const uint32_t SELECT_ALL_CODE_BITS = 0xFFFFFFFF; // Select a 20 bit word
00163     static const uint32_t SHL_CODE_BITS = 65536; // shift left by 10 bits
00164
00165     void generate_gps_l2c_m_prn_codes();
00166
00167     pcps_acquisition_fpga_sptr acquisition_fpga_;
00168     volk_gnssdr::vector<uint32_t> d_all_fft_codes_; // memory that contains all the code ffts
00169     std::weak_ptr<ChannelFsm> channel_fsm_;
00170     Gnss_Synchro* gnss_synchro_;
00171     Acq_Conf_Fpga acq_parameters_;
00172     std::string role_;
00173     float threshold_;
00174     unsigned int channel_;
00175     unsigned int doppler_max_;
00176     unsigned int doppler_step_;
00177     unsigned int in_streams_;
00178     unsigned int out_streams_;
00179 };
00180
00181 /** \} */

```

```
00180 /** \} */
00181 #endif // GNSS_SDR_GPS_L2_M_PCPS_ACQUISITION_FPGA_H
```

## 11.51 gps\_l5i\_pcps\_acquisition.h File Reference

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L5i signals.

```
#include "channel_fsm.h"
#include "complex_byte_to_float_x2.h"
#include "gnss_synchro.h"
#include "pcps_acquisition.h"
#include <gnuradio/blocks/float_to_complex.h>
#include <volk_gnssssdr/volk_gnssssdr_alloc.h>
#include <memory>
#include <string>
#include <utility>
```

Include dependency graph for gps\_l5i\_pcps\_acquisition.h:



### Classes

- class [GpsL5iPcpsAcquisition](#)

*This class adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L5i signals.*

### 11.51.1 Detailed Description

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L5i signals.

#### Authors

- Javier Arribas, 2017. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gps\\_l5i\\_pcps\\_acquisition.h](#).

---

## 11.52 gps\_l5i\_pcps\_acquisition.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002 * \file gps_l5i_pcps_acquisition.h
00003 * \brief Adapts a PCPS acquisition block to an AcquisitionInterface for
00004 * GPS L5i signals
00005 * \authors <ul>
00006 * <li> Javier Arribas, 2017. jarribas(at)cttc.es
00007 * </ul>
00008 *
00009 * -----
00010 *
00011 * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00012 * This file is part of GNSS-SDR.
00013 *
00014 * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00015 * SPDX-License-Identifier: GPL-3.0-or-later
00016 *
00017 * -----
00018 */
00019
00020 #ifndef GNSS_SDR_GPS_L5I_PCPS_ACQUISITION_H
00021 #define GNSS_SDR_GPS_L5I_PCPS_ACQUISITION_H
```

```

00022
00023 #include "channel_fsm.h"
00024 #include "complex_byte_to_float_x2.h"
00025 #include "gnss_synchro.h"
00026 #include "pcps_acquisition.h"
00027 #include <gnuradio/blocks/float_to_complex.h>
00028 #include <volk_gnssssdr/volk_gnssssdr_alloc.h>
00029 #include <memory>
00030 #include <string>
00031 #include <utility>
00032
00033 /** \addtogroup Acquisition
00034 * \{ */
00035 /** \addtogroup Acq_adapters
00036 * \{ */
00037
00038
00039 class ConfigurationInterface;
00040
00041 /*!
00042 * \brief This class adapts a PCPS acquisition block to an AcquisitionInterface
00043 * for GPS L5i signals
00044 */
00045 class GpsL5iPcpsAcquisition : public AcquisitionInterface
00046 {
00047 public:
00048     GpsL5iPcpsAcquisition(
00049         const ConfigurationInterface* configuration,
00050         const std::string& role,
00051         unsigned int in_streams,
00052         unsigned int out_streams);
00053
00054     ~GpsL5iPcpsAcquisition() = default;
00055
00056     inline std::string role() override
00057     {
00058         return role_;
00059     }
00060
00061     /*!
00062     * \brief Returns "GPS_L5i_PCPS_Acquisition"
00063     */
00064     inline std::string implementation() override
00065     {
00066         return "GPS_L5i_PCPS_Acquisition";
00067     }
00068
00069     inline size_t item_size() override
00070     {
00071         return item_size_;
00072     }
00073
00074     void connect(gr::top_block_sptr top_block) override;
00075     void disconnect(gr::top_block_sptr top_block) override;
00076     gr::basic_block_sptr get_left_block() override;
00077     gr::basic_block_sptr get_right_block() override;
00078
00079     /*!
00080     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00081     * to efficiently exchange synchronization data between acquisition and
00082     * tracking blocks
00083     */
00084     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00085
00086     /*!
00087     * \brief Set acquisition channel unique ID
00088     */
00089     inline void set_channel(unsigned int channel) override
00090     {
00091         channel_ = channel;
00092         acquisition_->set_channel(channel_);
00093     }
00094
00095     /*!
00096     * \brief Set channel fsm associated to this acquisition instance
00097     */
00098     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm) override
00099     {
00100         channel_fsm_ = std::move(channel_fsm);
00101         acquisition_->set_channel_fsm(channel_fsm_);
00102     }
00103
00104     /*!
00105     * \brief Set statistics threshold of PCPS algorithm
00106     */
00107     void set_threshold(float threshold) override;
00108

```

```

00109     /*!
00110      * \brief Set maximum Doppler off grid search
00111      */
00112     void set_doppler_max(unsigned int doppler_max) override;
00113
00114     /*!
00115      * \brief Set Doppler steps for the grid search
00116      */
00117     void set_doppler_step(unsigned int doppler_step) override;
00118
00119     /*!
00120      * \brief Set Doppler center for the grid search
00121      */
00122     void set_doppler_center(int doppler_center) override;
00123
00124     /*!
00125      * \brief Initializes acquisition algorithm.
00126      */
00127     void init() override;
00128
00129     /*!
00130      * \brief Sets local code for GPS L2/M PCPS acquisition algorithm.
00131      */
00132     void set_local_code() override;
00133
00134     /*!
00135      * \brief Returns the maximum peak of grid search
00136      */
00137     signed int mag() override;
00138
00139     /*!
00140      * \brief Restart acquisition algorithm
00141      */
00142     void reset() override;
00143
00144     /*!
00145      * \brief If state = 1, it forces the block to start acquiring from the first sample
00146      */
00147     void set_state(int state) override;
00148
00149     /*!
00150      * \brief Stop running acquisition
00151      */
00152     void stop_acquisition() override;
00153
00154     /*!
00155      * \brief Sets the resampler latency to account it in the acquisition code delay estimation
00156      */
00157     void set_resampler_latency(uint32_t latency_samples) override;
00158
00159 private:
00160     pcps_acquisition_sptr acquisition_;
00161     volk_gnss_sdr::vector<std::complex<float>> code_;
00162     gr::blocks::float_to_complex::sptr float_to_complex_;
00163     complex_byte_to_float_x2_sptr cbyte_to_float_x2_;
00164     std::weak_ptr<ChannelFsm> channel_fsm_;
00165     Gnss_Synchro* gnss_synchro_;
00166     Acq_Conf acq_parameters_;
00167     std::string item_type_;
00168     std::string dump_filename_;
00169     std::string role_;
00170     size_t item_size_;
00171     int64_t fs_in_;
00172     float threshold_;
00173     int doppler_center_;
00174     unsigned int vector_length_;
00175     unsigned int code_length_;
00176     unsigned int channel_;
00177     unsigned int doppler_max_;
00178     unsigned int doppler_step_;
00179     unsigned int in_streams_;
00180     unsigned int out_streams_;
00181     unsigned int num_codes_;
00182 };
00183
00184
00185 /** @} */
00186 /** @} */
00187 #endif // GNSS_SDR_GPS_L5I_PCPS_ACQUISITION_H

```

## 11.53 gps\_l5i\_pcps\_acquisition\_fpga.h File Reference

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for GPS L5i signals for the FPGA.



```

00029 #include <memory>
00030 #include <string>
00031 #include <utility>
00032 #include <vector>
00033
00034 /** \addtogroup Acquisition
00035  * \{ */
00036 /** \addtogroup Acq_adapters
00037  * \{ */
00038
00039
00040 class ConfigurationInterface;
00041
00042 /*!
00043  * \brief This class adapts a PCPS acquisition block off-loaded on an FPGA
00044  * to an AcquisitionInterface for GPS L5i signals
00045  */
00046 class GpsL5iPcpsAcquisitionFpga : public AcquisitionInterface
00047 {
00048 public:
00049     /*!
00050      * \brief Constructor
00051      */
00052     GpsL5iPcpsAcquisitionFpga(
00053         const ConfigurationInterface* configuration,
00054         const std::string& role,
00055         unsigned int in_streams,
00056         unsigned int out_streams);
00057
00058     /*!
00059      * \brief Destructor
00060      */
00061     ~GpsL5iPcpsAcquisitionFpga() = default;
00062
00063     /*!
00064      * \brief Role
00065      */
00066     inline std::string role() override
00067     {
00068         return role_;
00069     }
00070
00071     /*!
00072      * \brief Returns "GPS_L5i_PCPS_Acquisition_FPGA"
00073      */
00074     inline std::string implementation() override
00075     {
00076         return "GPS_L5i_PCPS_Acquisition_FPGA";
00077     }
00078
00079     /*!
00080      * \brief Returns size of lv_l6sc_t
00081      */
00082     inline size_t item_size() override
00083     {
00084         return sizeof(int16_t);
00085     }
00086
00087     /*!
00088      * \brief Connect
00089      */
00090     void connect(gr::top_block_sptr top_block) override;
00091
00092     /*!
00093      * \brief Disconnect
00094      */
00095     void disconnect(gr::top_block_sptr top_block) override;
00096
00097     /*!
00098      * \brief Get left block
00099      */
00100     gr::basic_block_sptr get_left_block() override;
00101
00102     /*!
00103      * \brief Get right block
00104      */
00105     gr::basic_block_sptr get_right_block() override;
00106
00107     /*!
00108      * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00109      * to efficiently exchange synchronization data between acquisition and
00110      * tracking blocks
00111      */
00112     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00113
00114     /*!
00115      * \brief Set acquisition channel unique ID

```

```


00116     */
00117     inline void set_channel(unsigned int channel) override
00118     {
00119         channel_ = channel;
00120         acquisition_fpga->set_channel(channel_);
00121     }
00122
00123     /*!
00124     * \brief Set channel fsm associated to this acquisition instance
00125     */
00126     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm) override
00127     {
00128         channel_fsm_ = std::move(channel_fsm);
00129         acquisition_fpga->set_channel_fsm(channel_fsm_);
00130     }
00131
00132     /*!
00133     * \brief Set statistics threshold of PCPS algorithm
00134     */
00135     void set_threshold(float threshold) override;
00136
00137     /*!
00138     * \brief Set maximum Doppler off grid search
00139     */
00140     void set_doppler_max(unsigned int doppler_max) override;
00141
00142     /*!
00143     * \brief Set Doppler steps for the grid search
00144     */
00145     void set_doppler_step(unsigned int doppler_step) override;
00146
00147     /*!
00148     * \brief Set Doppler center for the grid search
00149     */
00150     void set_doppler_center(int doppler_center) override;
00151
00152     /*!
00153     * \brief Initializes acquisition algorithm.
00154     */
00155     void init() override;
00156
00157     /*!
00158     * \brief Sets local code for GPS L5 PCPS acquisition algorithm.
00159     */
00160     void set_local_code() override;
00161
00162     /*!
00163     * \brief Returns the maximum peak of grid search
00164     */
00165     signed int mag() override;
00166
00167     /*!
00168     * \brief Restart acquisition algorithm
00169     */
00170     void reset() override;
00171
00172     /*!
00173     * \brief If state = 1, it forces the block to start acquiring from the first sample
00174     */
00175     void set_state(int state) override;
00176
00177     /*!
00178     * \brief Stop running acquisition
00179     */
00180     void stop_acquisition() override;
00181
00182     /*!
00183     * \brief Set resampler latency
00184     */
00185     void set_resampler_latency(uint32_t latency_samples __attribute__((unused))) override {};
00186
00187 private:
00188     static const uint32_t ACQ_BUFF_1 = 1; // FPGA Acquisition IP buffer containing L2 or
00189     L5/E5 frequency band samples by default.
00189     static const uint32_t DEFAULT_FPGA_BLK_EXP = 13; // default block exponent
00190     static const uint32_t NUM_PRNs = 32;
00191     static const uint32_t QUANT_BITS_LOCAL_CODE = 16;
00192     static const uint32_t SELECT_LSBITS = 0x0000FFFF; // Select the 10 Lsbits out of a 20-bit
00193     word
00193     static const uint32_t SELECT_MSBITS = 0xFFFF0000; // Select the 10 MSbits out of a 20-bit
00194     word
00194     static const uint32_t SELECT_ALL_CODE_BITS = 0xFFFFFFFF; // Select a 20 bit word
00195     static const uint32_t SHL_CODE_BITS = 65536; // shift left by 10 bits
00196
00197     void generate_gps_l5i_prn_codes();
00198     float calculate_threshold(float pfa);
00199

```



### 11.55 galileo\_e5a\_noncoherent\_iq\_acquisition\_caf\_cc.h File Reference

```
#include "channel_fsm.h"
#include "gnss_sdr_fft.h"
#include "gnss_synchro.h"
#include <gnuradio/block.h>
#include <gnuradio/gr_complex.h>
#include <fstream>
#include <memory>
#include <string>
#include <utility>
#include <vector>
```

[illegible]

```
graph TD; A[galileo_e5a_noncoherent_iq_acquisition_caf_cc.h] --> B[galileo_e5a_noncoherent_iq_acquisition_caf.h]
```

- class galileo\_e5a\_noncoherentIQ\_acquisition\_caf\_cc

*This class implements a Parallel Code Phase Search Acquisition.*

## Typedefs

- using [galileo\\_e5a\\_noncoherentIQ\\_acquisition\\_caf\\_cc\\_sptr](#) = gnss\_shared\_ptr<[galileo\\_e5a\\_noncoherentIQ\\_acquisition\\_caf\\_cc](#)

## Functions

- [galileo\\_e5a\\_noncoherentIQ\\_acquisition\\_caf\\_cc\\_sptr](#) [galileo\\_e5a\\_noncoherentIQ\\_make\\_acquisition\\_](#)  
[caf\\_cc](#) (unsigned int sampled\_ms, unsigned int max\_dwells, unsigned int doppler\_max, int64\_t fs\_in, int samples\_per\_ms, int samples\_per\_code, bool bit\_transition\_flag, bool dump, const std::string &dump\_  
filename, bool both\_signal\_components\_, int CAF\_window\_hz\_, int Zero\_padding\_, bool enable\_monitor\_  
output)

### 11.55.1 Detailed Description

Adapts a PCPS acquisition block to an [AcquisitionInterface](#) for Galileo E5a data and pilot Signals.

#### Author

Marc Sales, 2014. [marcsales92\(at\)gmail.com](mailto:marcsales92(at)gmail.com) \based on work from:

- Javier Arribas, 2011. [jarribas\(at\)cttc.es](mailto:jarribas(at)cttc.es)
- Luis Esteve, 2012. [luis\(at\)epsilon-formation.com](mailto:luis(at)epsilon-formation.com)
- Marc Molina, 2013. [marc.molina.pena@gmail.com](mailto:marc.molina.pena@gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [galileo\\_e5a\\_noncoherent\\_iq\\_acquisition\\_caf\\_cc.h](#).

---

## 11.56 galileo\_e5a\_noncoherent\_iq\_acquisition\_caf\_cc.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file galileo_e5a_noncoherent_iq_acquisition_caf_cc.h
00003  * \brief Adapts a PCPS acquisition block to an AcquisitionInterface for
00004  * Galileo E5a data and pilot Signals
00005  * \author Marc Sales, 2014. marcsales92\(at\)gmail.com
00006  * \based on work from:
00007  *      <ul>
00008  *          <li> Javier Arribas, 2011. jarribas\(at\)cttc.es
00009  *          <li> Luis Esteve, 2012. luis\(at\)epsilon-formation.com
00010  *          <li> Marc Molina, 2013. marc.molina.pena@gmail.com
00011  *      </ul>
00012  *
00013  * -----
00014  *
00015  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00016  * This file is part of GNSS-SDR.
00017  *
00018  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00019  * SPDX-License-Identifier: GPL-3.0-or-later
00020  *
00021  * -----
00022  */
00023
00024 #ifndef GNSS_SDR_GALILEO_E5A_NONCOHERENT_IQ_ACQUISITION_CAF_CC_H
00025 #define GNSS_SDR_GALILEO_E5A_NONCOHERENT_IQ_ACQUISITION_CAF_CC_H
00026
00027 #include "channel_fsm.h"
00028 #include "gnss_sdr_fft.h"
00029 #include "gnss_synchro.h"
00030 #include <gnuradio/block.h>
00031 #include <gnuradio/gr_complex.h>
00032 #include <fstream>
00033 #include <memory>
00034 #include <string>
00035 #include <utility>
00036 #include <vector>
00037
00038 /** \addtogroup Acquisition
00039  * \{ */
```

```

00040 /** \addtogroup Acq_gnuradio_blocks
00041 * \{ */
00042
00043
00044 class galileo_e5a_noncoherentIQ_acquisition_caf_cc;
00045
00046 using galileo_e5a_noncoherentIQ_acquisition_caf_cc_sptr =
    gnss_shared_ptr<galileo_e5a_noncoherentIQ_acquisition_caf_cc>;
00047
00048 galileo_e5a_noncoherentIQ_acquisition_caf_cc_sptr galileo_e5a_noncoherentIQ_make_acquisition_caf_cc(
00049     unsigned int sampled_ms,
00050     unsigned int max_dwells,
00051     unsigned int doppler_max, int64_t fs_in,
00052     int samples_per_ms, int samples_per_code,
00053     bool bit_transition_flag,
00054     bool dump,
00055     const std::string& dump_filename,
00056     bool both_signal_components_,
00057     int CAF_window_hz_,
00058     int Zero_padding_,
00059     bool enable_monitor_output);
00060
00061 /*!
00062 * \brief This class implements a Parallel Code Phase Search Acquisition.
00063 *
00064 * Check \ref Navitec2012 "An Open Source Galileo E1 Software Receiver",
00065 * Algorithm 1, for a pseudocode description of this implementation.
00066 */
00067 class galileo_e5a_noncoherentIQ_acquisition_caf_cc : public gr::block
00068 {
00069 public:
00070     /*!
00071     * \brief Default destructor.
00072     */
00073     ~galileo_e5a_noncoherentIQ_acquisition_caf_cc();
00074
00075     /*!
00076     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00077     * to exchange synchronization data between acquisition and tracking blocks.
00078     * \param p_gnss_synchro Satellite information shared by the processing blocks.
00079     */
00080     inline void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro)
00081     {
00082         d_gnss_synchro = p_gnss_synchro;
00083     }
00084
00085     /*!
00086     * \brief Returns the maximum peak of grid search.
00087     */
00088     inline unsigned int mag() const
00089     {
00090         return d_mag;
00091     }
00092
00093     /*!
00094     * \brief Initializes acquisition algorithm.
00095     */
00096     void init();
00097
00098     /*!
00099     * \brief Sets local code for PCPS acquisition algorithm.
00100     * \param code - Pointer to the PRN code.
00101     */
00102     void set_local_code(std::complex<float>* code, std::complex<float>* codeQ);
00103
00104     /*!
00105     * \brief Starts acquisition algorithm, turning from standby mode to
00106     * active mode
00107     * \param active - bool that activates/deactivates the block.
00108     */
00109     inline void set_active(bool active)
00110     {
00111         d_active = active;
00112     }
00113
00114     /*!
00115     * \brief If set to 1, ensures that acquisition starts at the
00116     * first available sample.
00117     * \param state - int=1 forces start of acquisition
00118     */
00119     void set_state(int state);
00120
00121     /*!
00122     * \brief Set acquisition channel unique ID
00123     * \param channel - receiver channel.
00124     */
00125     inline void set_channel(unsigned int channel)

```

```

00126     {
00127         d_channel = channel;
00128     }
00129
00130     /*!
00131     * \brief Set channel fsm associated to this acquisition instance
00132     */
00133     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm)
00134     {
00135         d_channel_fsm = std::move(channel_fsm);
00136     }
00137
00138     /*!
00139     * \brief Set statistics threshold of PCPS algorithm.
00140     * \param threshold - Threshold for signal detection (check \ref Navitec2012,
00141     * Algorithm 1, for a definition of this threshold).
00142     */
00143     inline void set_threshold(float threshold)
00144     {
00145         d_threshold = threshold;
00146     }
00147
00148     /*!
00149     * \brief Set maximum Doppler grid search
00150     * \param doppler_max - Maximum Doppler shift considered in the grid search [Hz].
00151     */
00152     inline void set_doppler_max(unsigned int doppler_max)
00153     {
00154         d_doppler_max = doppler_max;
00155     }
00156
00157     /*!
00158     * \brief Set Doppler steps for the grid search
00159     * \param doppler_step - Frequency bin of the search grid [Hz].
00160     */
00161     inline void set_doppler_step(unsigned int doppler_step)
00162     {
00163         d_doppler_step = doppler_step;
00164     }
00165
00166     /*!
00167     * \brief Parallel Code Phase Search Acquisition signal processing.
00168     */
00169     int general_work(int noutput_items, gr_vector_int& ninput_items,
00170                     gr_vector_const_void_star& input_items,
00171                     gr_vector_void_star& output_items);
00172
00173 private:
00174     friend galileo_e5a_noncoherentIQ_acquisition_caf_cc_sptr
00175     galileo_e5a_noncoherentIQ_make_acquisition_caf_cc(
00176         unsigned int sampled_ms,
00177         unsigned int max_dwells,
00178         unsigned int doppler_max, int64_t fs_in,
00179         int samples_per_ms, int samples_per_code,
00180         bool bit_transition_flag,
00181         bool dump,
00182         const std::string& dump_filename,
00183         bool both_signal_components_,
00184         int CAF_window_hz_,
00185         int Zero_padding_,
00186         bool enable_monitor_output);
00187
00188     galileo_e5a_noncoherentIQ_acquisition_caf_cc(
00189         unsigned int sampled_ms,
00190         unsigned int max_dwells,
00191         unsigned int doppler_max, int64_t fs_in,
00192         int samples_per_ms, int samples_per_code,
00193         bool bit_transition_flag,
00194         bool dump,
00195         const std::string& dump_filename,
00196         bool both_signal_components_,
00197         int CAF_window_hz_,
00198         int Zero_padding_,
00199         bool enable_monitor_output);
00200
00201     void calculate_magnitudes(gr_complex* fft_begin, int doppler_shift,
00202                             int doppler_offset);
00203
00204     float estimate_input_power(gr_complex* in);
00205
00206     std::weak_ptr<ChannelFsm> d_channel_fsm;
00207     std::unique_ptr<gnss_fft_complex_fwd> d_fft_if;
00208     std::unique_ptr<gnss_fft_complex_rev> d_ifft;
00209
00210     std::vector<std::vector<gr_complex>> d_grid_doppler_wipeoffs;
00211     std::vector<gr_complex> d_fft_code_I_A;
00212     std::vector<gr_complex> d_fft_code_I_B;

```

```

00213     std::vector<gr_complex> d_fft_code_Q_A;
00214     std::vector<gr_complex> d_fft_code_Q_B;
00215     std::vector<gr_complex> d_inbuffer;
00216     std::vector<float> d_magnitudeIA;
00217     std::vector<float> d_magnitudeIB;
00218     std::vector<float> d_magnitudeQA;
00219     std::vector<float> d_magnitudeQB;
00220     std::vector<float> d_CAF_vector;
00221     std::vector<float> d_CAF_vector_I;
00222     std::vector<float> d_CAF_vector_Q;
00223
00224     std::string d_satellite_str;
00225     std::string d_dump_filename;
00226
00227     std::ofstream d_dump_file;
00228
00229     Gnss_Synchro* d_gnss_synchro;
00230
00231     int64_t d_fs_in;
00232     uint64_t d_sample_counter;
00233
00234     float d_threshold;
00235     float d_doppler_freq;
00236     float d_mag;
00237     float d_input_power;
00238     float d_test_statistics;
00239
00240     int d_state;
00241     int d_samples_per_ms;
00242     int d_samples_per_code;
00243     int d_CAF_window_hz;
00244     int d_buffer_count;
00245     int d_doppler_resolution;
00246     int d_doppler_max;
00247     int d_doppler_step;
00248     int d_fft_size;
00249     int d_num_doppler_bins;
00250     unsigned int d_gr_stream_buffer;
00251     unsigned int d_channel;
00252     unsigned int d_max_dwells;
00253     unsigned int d_well_count;
00254     unsigned int d_sampled_ms;
00255     unsigned int d_code_phase;
00256
00257     bool d_bit_transition_flag;
00258     bool d_active;
00259     bool d_dump;
00260     bool d_both_signal_components;
00261     bool d_enable_monitor_output;
00262 };
00263
00264
00265 /** \} */
00266 /** \} */
00267 #endif // GNSS_SDR_GALILEO_E5A_NONCOHERENT_IQ_ACQUISITION_CAF_CC_H

```

## 11.57 galileo\_pcps\_8ms\_acquisition\_cc.h File Reference

This class implements a Parallel Code Phase Search Acquisition for Galileo E1 signals with coherent integration time = 8 ms (two codes)

```

#include "channel_fsm.h"
#include "gnss_sdr_fft.h"
#include "gnss_synchro.h"
#include <gnuradio/block.h>
#include <gnuradio/gr_complex.h>
#include <fstream>
#include <memory>
#include <string>
#include <utility>
#include <vector>

```



## 11.58 galileo\_pcps\_8ms\_acquisition\_cc.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file galileo_pcps_8ms_acquisition_cc.h
00003  * \brief This class implements a Parallel Code Phase Search Acquisition for
00004  * Galileo E1 signals with coherent integration time = 8 ms (two codes)
00005  * \author Marc Molina, 2013. marc.molina.pena(at)gmail.com
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_PCPS_8MS_ACQUISITION_CC_H
00019 #define GNSS_SDR_PCPS_8MS_ACQUISITION_CC_H
00020
00021 #include "channel_fsm.h"
00022 #include "gnss_sdr_fft.h"
00023 #include "gnss_synchro.h"
00024 #include <gnuradio/block.h>
00025 #include <gnuradio/gr_complex.h>
00026 #include <fstream>
00027 #include <memory>
00028 #include <string>
00029 #include <utility>
00030 #include <vector>
00031
00032 /** \addtogroup Acquisition
00033  * \{ */
00034 /** \addtogroup Acq_gnuradio_blocks
00035  * \{ */
00036
00037
00038 class galileo_pcps_8ms_acquisition_cc;
00039
00040 using galileo_pcps_8ms_acquisition_cc_sptr = gnss_shared_ptr<galileo_pcps_8ms_acquisition_cc>;
00041
00042 galileo_pcps_8ms_acquisition_cc_sptr
00043 galileo_pcps_8ms_make_acquisition_cc(uint32_t sampled_ms,
00044   uint32_t max_dwells,
00045   uint32_t doppler_max,
00046   int64_t fs_in,
00047   int32_t samples_per_ms,
00048   int32_t samples_per_code,
00049   bool dump,
00050   const std::string& dump_filename,
00051   bool enable_monitor_output);
00052
00053 /*!
00054 * \brief This class implements a Parallel Code Phase Search Acquisition for
00055 * Galileo E1 signals with coherent integration time = 8 ms (two codes)
00056 */
00057 class galileo_pcps_8ms_acquisition_cc : public gr::block
00058 {
00059 public:
00060     /*!
00061     * \brief Default destructor.
00062     */
00063     ~galileo_pcps_8ms_acquisition_cc();
00064
00065     /*!
00066     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00067     * to exchange synchronization data between acquisition and tracking blocks.
00068     * \param p_gnss_synchro Satellite information shared by the processing blocks.
00069     */
00070     inline void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro)
00071     {
00072         d_gnss_synchro = p_gnss_synchro;
00073     }
00074
00075     /*!
00076     * \brief Returns the maximum peak of grid search.
00077     */
00078     inline uint32_t mag() const
00079     {
00080         return d_mag;
00081     }
00082
00083     /*!

```

```

00084     * \brief Initializes acquisition algorithm.
00085     */
00086 void init();
00087
00088     /*!
00089     * \brief Sets local code for PCPS acquisition algorithm.
00090     * \param code - Pointer to the PRN code.
00091     */
00092 void set_local_code(std::complex<float>* code);
00093
00094     /*!
00095     * \brief Starts acquisition algorithm, turning from standby mode to
00096     * active mode
00097     * \param active - bool that activates/deactivates the block.
00098     */
00099 inline void set_active(bool active)
00100 {
00101     d_active = active;
00102 }
00103
00104     /*!
00105     * \brief If set to 1, ensures that acquisition starts at the
00106     * first available sample.
00107     * \param state - int=1 forces start of acquisition
00108     */
00109 void set_state(int32_t state);
00110
00111     /*!
00112     * \brief Set acquisition channel unique ID
00113     * \param channel - receiver channel.
00114     */
00115 inline void set_channel(uint32_t channel)
00116 {
00117     d_channel = channel;
00118 }
00119
00120     /*!
00121     * \brief Set channel fsm associated to this acquisition instance
00122     */
00123 inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm)
00124 {
00125     d_channel_fsm = std::move(channel_fsm);
00126 }
00127
00128     /*!
00129     * \brief Set statistics threshold of PCPS algorithm.
00130     * \param threshold - Threshold for signal detection (check \ref Navitec2012,
00131     * Algorithm 1, for a definition of this threshold).
00132     */
00133 inline void set_threshold(float threshold)
00134 {
00135     d_threshold = threshold;
00136 }
00137
00138     /*!
00139     * \brief Set maximum Doppler grid search
00140     * \param doppler_max - Maximum Doppler shift considered in the grid search [Hz].
00141     */
00142 inline void set_doppler_max(uint32_t doppler_max)
00143 {
00144     d_doppler_max = doppler_max;
00145 }
00146
00147     /*!
00148     * \brief Set Doppler steps for the grid search
00149     * \param doppler_step - Frequency bin of the search grid [Hz].
00150     */
00151 inline void set_doppler_step(uint32_t doppler_step)
00152 {
00153     d_doppler_step = doppler_step;
00154 }
00155
00156     /*!
00157     * \brief Parallel Code Phase Search Acquisition signal processing.
00158     */
00159 int general_work(int noutput_items, gr_vector_int& ninput_items,
00160     gr_vector_const_void_star& input_items,
00161     gr_vector_void_star& output_items);
00162
00163 private:
00164 friend galileo_pcps_8ms_acquisition_cc_sptr
00165 galileo_pcps_8ms_make_acquisition_cc(
00166     uint32_t sampled_ms,
00167     uint32_t max_dwells,
00168     uint32_t doppler_max,
00169     int64_t fs_in,
00170     int32_t samples_per_ms,

```



```

00171         int32_t samples_per_code,
00172         bool dump,
00173         const std::string& dump_filename,
00174         bool enable_monitor_output);
00175
00176     galileo_pcps_8ms_acquisition_cc(
00177         uint32_t sampled_ms,
00178         uint32_t max_dwells,
00179         uint32_t doppler_max,
00180         int64_t fs_in,
00181         int32_t samples_per_ms,
00182         int32_t samples_per_code,
00183         bool dump,
00184         const std::string& dump_filename,
00185         bool enable_monitor_output);
00186
00187     void calculate_magnitudes(
00188         gr_complex* fft_begin,
00189         int32_t doppler_shift,
00190         int32_t doppler_offset);
00191
00192     std::weak_ptr<ChannelFsm> d_channel_fsm;
00193     std::unique_ptr<gnss_fft_complex_fwd> d_fft_if;
00194     std::unique_ptr<gnss_fft_complex_rev> d_ifft;
00195
00196     std::vector<std::vector<gr_complex>> d_grid_doppler_wipeoffs;
00197     std::vector<gr_complex> d_fft_code_A;
00198     std::vector<gr_complex> d_fft_code_B;
00199     std::vector<float> d_magnitude;
00200
00201     std::string d_satellite_str;
00202     std::string d_dump_filename;
00203     std::ofstream d_dump_file;
00204
00205     Gnss_Synchro* d_gnss_synchro;
00206
00207     int64_t d_fs_in;
00208     uint64_t d_sample_counter;
00209
00210     float d_threshold;
00211     float d_doppler_freq;
00212     float d_mag;
00213     float d_input_power;
00214     float d_test_statistics;
00215     int32_t d_state;
00216     int32_t d_samples_per_ms;
00217     int32_t d_samples_per_code;
00218     uint32_t d_channel;
00219     uint32_t d_doppler_resolution;
00220     uint32_t d_doppler_max;
00221     uint32_t d_doppler_step;
00222     uint32_t d_sampled_ms;
00223     uint32_t d_max_dwells;
00224     uint32_t d_well_count;
00225     uint32_t d_fft_size;
00226     uint32_t d_num_doppler_bins;
00227     uint32_t d_code_phase;
00228
00229     bool d_active;
00230     bool d_dump;
00231     bool d_enable_monitor_output;
00232 };
00233
00234
00235 /** \} */
00236 /** \} */
00237 #endif // GNSS_SDR_PCPS_8MS_ACQUISITION_CC_H

```

## 11.59 pcps\_acquisition.h File Reference

This class implements a Parallel Code Phase Search Acquisition.

```

#include "acq_conf.h"
#include "channel_fsm.h"
#include "gnss_sdr_fft.h"
#include <armadillo>
#include <gnuradio/block.h>
#include <gnuradio/gr_complex.h>
#include <gnuradio/thread/thread.h>
#include <gnuradio/types.h>
#include <volk/volk_complex.h>

```



## Authors

- Javier Arribas, 2011. jarribas(at)cttc.es
- Luis Esteve, 2012. luis(at)epsilon-formacion.com
- Marc Molina, 2013. marc.molina.pena@gmail.com
- Cillian O'Driscoll, 2017. cillian(at)ieee.org
- Antonio Ramos, 2017. antonio.ramos@cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [pcps\\_acquisition.h](#).

---

## 11.60 pcps\_acquisition.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file pcps_acquisition.h
00003  * \brief This class implements a Parallel Code Phase Search Acquisition
00004  *
00005  * Acquisition strategy (Kay Borre book + CFAR threshold).
00006  * <ol>
00007  * <li> Compute the input signal power estimation
00008  * <li> Doppler serial search loop
00009  * <li> Perform the FFT-based circular convolution (parallel time search)
00010  * <li> Record the maximum peak and the associated synchronization parameters
00011  * <li> Compute the test statistics and compare to the threshold
00012  * <li> Declare positive or negative acquisition using a message queue
00013  * </ol>
00014  *
00015  * Kay Borre book: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen,
00016  * "A Software-Defined GPS and Galileo Receiver. A Single-Frequency
00017  * Approach", Birkhauser, 2007. pp 81-84
00018  *
00019  * \authors <ul>
00020  * <li> Javier Arribas, 2011. jarribas(at)cttc.es
00021  * <li> Luis Esteve, 2012. luis(at)epsilon-formacion.com
00022  * <li> Marc Molina, 2013. marc.molina.pena@gmail.com
00023  * <li> Cillian O'Driscoll, 2017. cillian(at)ieee.org
00024  * <li> Antonio Ramos, 2017. antonio.ramos@cttc.es
00025  * </ul>
00026  *
00027  * -----
00028  *
00029  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00030  * This file is part of GNSS-SDR.
00031  *
00032  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00033  * SPDX-License-Identifier: GPL-3.0-or-later
00034  *
00035  * -----
00036  */
00037
00038 #ifndef GNSS_SDR_PCPS_ACQUISITION_H
00039 #define GNSS_SDR_PCPS_ACQUISITION_H
00040
00041 #if ARMA_NO_BOUND_CHECKING
00042 #define ARMA_NO_DEBUG 1
00043 #endif
00044
00045 #include "acq_conf.h"
00046 #include "channel_fsm.h"
00047 #include "gnss_sdr_fft.h"
00048 #include <armadillo>
00049 #include <gnuradio/block.h>
00050 #include <gnuradio/gr_complex.h> // for gr_complex
00051 #include <gnuradio/thread/thread.h> // for scoped_lock
00052 #include <gnuradio/types.h> // for gr_vector_const_void_star
00053 #include <volk/volk_complex.h> // for lv_l6sc_t
00054 #include <volk_gnssdr/volk_gnssdr_alloc.h> // for volk_gnssdr::vector
00055 #include <complex>
00056 #include <stdint>
00057 #include <memory>
00058 #include <queue>
00059 #include <string>
00060 #include <utility>
00061
00062
00063 #if HAS_STD_SPAN
00064 #include <span>

```

```

00065 namespace own = std;
00066 #else
00067 #include <gsl/gsl-lite.hpp>
00068 namespace own = gsl;
00069 #endif
00070
00071 /** \addtogroup Acquisition
00072  * Classes for GNSS signal acquisition
00073  * \{ */
00074 /** \addtogroup Acq_gnuradio_blocks acquisition_gr_blocks
00075  * GNU Radio processing blocks for GNSS signal acquisition
00076  * \{ */
00077
00078
00079 class Gnss_Synchro;
00080 class pcps_acquisition;
00081
00082 using pcps_acquisition_sptr = gnss_shared_ptr<pcps_acquisition>;
00083
00084 pcps_acquisition_sptr pcps_make_acquisition(const Acq_Conf& conf_);
00085
00086 /*!
00087  * \brief This class implements a Parallel Code Phase Search Acquisition.
00088  *
00089  * Check \ref Navitec2012 "An Open Source Galileo E1 Software Receiver",
00090  * Algorithm 1, for a pseudocode description of this implementation.
00091  */
00092 class pcps_acquisition : public gr::block
00093 {
00094 public:
00095     ~pcps_acquisition() override = default;
00096
00097     /*!
00098      * \brief Initializes acquisition algorithm and reserves memory.
00099      */
00100     void init();
00101
00102     /*!
00103      * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00104      * to exchange synchronization data between acquisition and tracking blocks.
00105      * \param p_gnss_synchro Satellite information shared by the processing blocks.
00106      */
00107     inline void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro)
00108     {
00109         gr::thread::scoped_lock lock(d_setlock); // require mutex with work function called by the
scheduler
00110         d_gnss_synchro = p_gnss_synchro;
00111     }
00112
00113     /*!
00114      * \brief Sets local code for PCPS acquisition algorithm.
00115      * \param code - Pointer to the PRN code.
00116      */
00117     void set_local_code(std::complex<float>* code);
00118
00119     /*!
00120      * \brief If set to 1, ensures that acquisition starts at the
00121      * first available sample.
00122      * \param state - int=1 forces start of acquisition
00123      */
00124     void set_state(int32_t state);
00125
00126     void set_resampler_latency(uint32_t latency_samples);
00127
00128     /*!
00129      * \brief Returns the maximum peak of grid search.
00130      */
00131     inline uint32_t mag() const
00132     {
00133         return d_mag;
00134     }
00135
00136     /*!
00137      * \brief Starts acquisition algorithm, turning from standby mode to
00138      * active mode
00139      * \param active - bool that activates/deactivates the block.
00140      */
00141     inline void set_active(bool active)
00142     {
00143         gr::thread::scoped_lock lock(d_setlock); // require mutex with work function called by the
scheduler
00144         d_active = active;
00145     }
00146
00147     /*!
00148      * \brief Set acquisition channel unique ID
00149      * \param channel - receiver channel.

```

```

00150     */
00151     inline void set_channel(uint32_t channel)
00152     {
00153         d_channel = channel;
00154     }
00155
00156     /*!
00157     * \brief Set channel fsm associated to this acquisition instance
00158     */
00159     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm)
00160     {
00161         d_channel_fsm = std::move(channel_fsm);
00162     }
00163
00164     /*!
00165     * \brief Set statistics threshold of PCPS algorithm.
00166     * \param threshold - Threshold for signal detection (check \ref Navitec2012,
00167     * \param threshold - Threshold for signal detection (check \ref Navitec2012,
00168     * \param threshold - Threshold for signal detection (check \ref Navitec2012,
00169     */
00169     inline void set_threshold(float threshold)
00170     {
00171         gr::thread::scoped_lock lock(d_setlock); // require mutex with work function called by the
scheduler
00172         d_threshold = threshold;
00173     }
00174
00175     /*!
00176     * \brief Set maximum Doppler grid search
00177     * \param doppler_max - Maximum Doppler shift considered in the grid search [Hz].
00178     */
00179     inline void set_doppler_max(uint32_t doppler_max)
00180     {
00181         gr::thread::scoped_lock lock(d_setlock); // require mutex with work function called by the
scheduler
00182         d_acq_parameters.doppler_max = doppler_max;
00183     }
00184
00185     /*!
00186     * \brief Set Doppler steps for the grid search
00187     * \param doppler_step - Frequency bin of the search grid [Hz].
00188     */
00189     inline void set_doppler_step(uint32_t doppler_step)
00190     {
00191         gr::thread::scoped_lock lock(d_setlock); // require mutex with work function called by the
scheduler
00192         d_doppler_step = doppler_step;
00193     }
00194
00195     /*!
00196     * \brief Set Doppler center frequency for the grid search. It will refresh the Doppler grid.
00197     * \param doppler_center - Frequency center of the search grid [Hz].
00198     */
00199     void set_doppler_center(int32_t doppler_center);
00200
00201     /*!
00202     * \brief Parallel Code Phase Search Acquisition signal processing.
00203     */
00204     int general_work(int noutput_items, gr_vector_int& ninput_items,
00205                     gr_vector_const_void_star& input_items,
00206                     gr_vector_void_star& output_items) override;
00207
00208 private:
00209     friend pcps_acquisition_sptr pcps_make_acquisition(const Acq_Conf& conf_);
00210     explicit pcps_acquisition(const Acq_Conf& conf_);
00211
00212     void update_local_carrier(own::span<gr_complex> carrier_vector, float freq) const;
00213     void update_grid_doppler_wipeoffs();
00214     void update_grid_doppler_wipeoffs_step2();
00215     void acquisition_core(uint64_t samp_count);
00216     void send_negative_acquisition();
00217     void send_positive_acquisition();
00218     void dump_results(int32_t effective_fft_size);
00219     bool is_fdma();
00220     bool start() override;
00221     void calculate_threshold(void);
00222     float first_vs_second_peak_statistic(uint32_t& index, int32_t& doppler, uint32_t
num_doppler_bins, int32_t doppler_max, int32_t doppler_step);
00223     float max_to_input_power_statistic(uint32_t& index, int32_t& doppler, uint32_t num_doppler_bins,
int32_t doppler_max, int32_t doppler_step);
00224
00225     volk_gnssssdr::vector<volk_gnssssdr::vector<float>> d_magnitude_grid;
00226     volk_gnssssdr::vector<float> d_tmp_buffer;
00227     volk_gnssssdr::vector<std::complex<float>> d_input_signal;
00228     volk_gnssssdr::vector<volk_gnssssdr::vector<std::complex<float>>> d_grid_doppler_wipeoffs;
00229     volk_gnssssdr::vector<volk_gnssssdr::vector<std::complex<float>>> d_grid_doppler_wipeoffs_step_two;
00230     volk_gnssssdr::vector<std::complex<float>> d_fft_codes;
00231     volk_gnssssdr::vector<std::complex<float>> d_data_buffer;

```

```

00232     volk_gnssssdr::vector<lv_l6sc_t> d_data_buffer_sc;
00233
00234     std::unique_ptr<gnss_fft_complex_fwd> d_fft_if;
00235     std::unique_ptr<gnss_fft_complex_rev> d_ifft;
00236     std::weak_ptr<ChannelFsm> d_channel_fsm;
00237
00238     Acq_Conf d_acq_parameters;
00239     Gnss_Synchro* d_gnss_synchro;
00240     arma::fmat d_grid;
00241     arma::fmat d_narrow_grid;
00242
00243     std::queue<Gnss_Synchro> d_monitor_queue;
00244     std::string d_dump_filename;
00245
00246     int64_t d_dump_number;
00247     uint64_t d_sample_counter;
00248
00249     float d_threshold;
00250     float d_mag;
00251     float d_input_power;
00252     float d_test_statistics;
00253     float d_doppler_center_step_two;
00254
00255     int32_t d_state;
00256     int32_t d_positive_acq;
00257     int32_t d_doppler_center;
00258     int32_t d_doppler_bias;
00259     uint32_t d_channel;
00260     uint32_t d_samplesPerChip;
00261     uint32_t d_doppler_step;
00262     uint32_t d_num_noncoherent_integrations_counter;
00263     uint32_t d_fft_size;
00264     uint32_t d_consumed_samples;
00265     uint32_t d_num_doppler_bins;
00266     uint32_t d_num_doppler_bins_step2;
00267     uint32_t d_dump_channel;
00268     uint32_t d_buffer_count;
00269
00270     bool d_active;
00271     bool d_worker_active;
00272     bool d_cshort;
00273     bool d_step_two;
00274     bool d_use_CFAR_algorithm_flag;
00275     bool d_dump;
00276 };
00277
00278
00279 /** \} */
00280 /** \} */
00281 #endif // GNSS_SDR_PCPS_ACQUISITION_H

```

## 11.61 pcps\_acquisition\_fine\_doppler\_cc.h File Reference

This class implements a Parallel Code Phase Search Acquisition with multi-dwells and fine Doppler estimation for GPS L1 C/A signal.

```

#include "acq_conf.h"
#include "channel_fsm.h"
#include "gnss_sdr_fft.h"
#include "gnss_synchro.h"
#include <armadillo>
#include <gnuradio/block.h>
#include <gnuradio/gr_complex.h>
#include <volk_gnssssdr/volk_gnssssdr_alloc.h>
#include <stdint>
#include <fstream>
#include <memory>
#include <string>
#include <utility>

```



## Authors

- Javier Arribas, 2013. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [pcps\\_acquisition\\_fine\\_doppler\\_cc.h](#).

---

## 11.62 pcps\_acquisition\_fine\_doppler\_cc.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file pcps_acquisition_fine_doppler_cc.h
00003  * \brief This class implements a Parallel Code Phase Search Acquisition with multi-dwells and fine
Doppler estimation
00004  * for GPS L1 C/A signal
00005  *
00006  * Acquisition strategy (Kay Borre book).
00007  * <ol>
00008  * <li> Compute the input signal power estimation
00009  * <li> Doppler serial search loop
00010  * <li> Perform the FFT-based circular convolution (parallel time search)
00011  * <li> Record the maximum peak and the associated synchronization parameters
00012  * <li> Compute the test statistics and compare to the threshold
00013  * <li> Declare positive or negative acquisition using a message port
00014  * </ol>
00015  *
00016  * Kay Borre book: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen,
00017  * "A Software-Defined GPS and Galileo Receiver. A Single-Frequency
00018  * Approach", Birkhauser, 2007. pp 81-84
00019  *
00020  * \authors <ul>
00021  * <li> Javier Arribas, 2013. jarribas(at)cttc.es
00022  * </ul>
00023  *
00024  * -----
00025  *
00026  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00027  * This file is part of GNSS-SDR.
00028  *
00029  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00030  * SPDX-License-Identifier: GPL-3.0-or-later
00031  *
00032  * -----
00033  */
00034
00035 #ifndef GNSS_SDR_PCPS_ACQUISITION_FINE_DOPPLER_CC_H
00036 #define GNSS_SDR_PCPS_ACQUISITION_FINE_DOPPLER_CC_H
00037
00038 #if ARMA_NO_BOUND_CHECKING
00039 #define ARMA_NO_DEBUG 1
00040 #endif
00041
00042 #include "acq_conf.h"
00043 #include "channel_fsm.h"
00044 #include "gnss_sdr_fft.h"
00045 #include "gnss_synchro.h"
00046 #include <armadillo>
00047 #include <gnuradio/block.h>
00048 #include <gnuradio/gr_complex.h>
00049 #include <volk_gnssssdr/volk_gnssssdr_alloc.h> // for volk_gnssssdr::vector
00050 #include <stdint>
00051 #include <fstream>
00052 #include <memory>
00053 #include <string>
00054 #include <utility>
00055
00056
00057 /** \addtogroup Acquisition
00058  * \{ */
00059 /** \addtogroup Acq_gnuradio_blocks
00060  * \{ */
00061
00062
00063 class pcps_acquisition_fine_doppler_cc;
00064
00065 using pcps_acquisition_fine_doppler_cc_sptr = gnss_shared_ptr<pcps_acquisition_fine_doppler_cc>;
00066
00067 pcps_acquisition_fine_doppler_cc_sptr pcps_make_acquisition_fine_doppler_cc(const Acq_Conf& conf_);
00068
00069 /*!
00070  * \brief This class implements a Parallel Code Phase Search Acquisition.
00071  *

```



```

00072  */
00073  class pcps_acquisition_fine_doppler_cc : public gr::block
00074  {
00075  public:
00076      /*!
00077       * \brief Default destructor.
00078       */
00079      ~pcps_acquisition_fine_doppler_cc() = default;
00080
00081      /*!
00082       * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00083       * to exchange synchronization data between acquisition and tracking blocks.
00084       * \param p_gnss_synchro Satellite information shared by the processing blocks.
00085       */
00086      inline void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro)
00087      {
00088          d_gnss_synchro = p_gnss_synchro;
00089      }
00090
00091      /*!
00092       * \brief Returns the maximum peak of grid search.
00093       */
00094      inline unsigned int mag() const
00095      {
00096          return d_test_statistics;
00097      }
00098
00099      /*!
00100       * \brief Initializes acquisition algorithm.
00101       */
00102      void init();
00103
00104      /*!
00105       * \brief Sets local code for PCPS acquisition algorithm.
00106       * \param code - Pointer to the PRN code.
00107       */
00108      void set_local_code(std::complex<float>* code);
00109
00110      /*!
00111       * \brief Starts acquisition algorithm, turning from standby mode to
00112       * active mode
00113       * \param active - bool that activates/deactivates the block.
00114       */
00115      inline void set_active(bool active)
00116      {
00117          d_active = active;
00118      }
00119
00120      /*!
00121       * \brief Set acquisition channel unique ID
00122       * \param channel - receiver channel.
00123       */
00124      inline void set_channel(unsigned int channel)
00125      {
00126          d_channel = channel;
00127          d_dump_channel = d_channel;
00128      }
00129
00130      /*!
00131       * \brief Set channel fsm associated to this acquisition instance
00132       */
00133      inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm)
00134      {
00135          d_channel_fsm = std::move(channel_fsm);
00136      }
00137
00138      /*!
00139       * \brief Set statistics threshold of PCPS algorithm.
00140       * \param threshold - Threshold for signal detection (check \ref Navitec2012,
00141       * Algorithm 1, for a definition of this threshold).
00142       */
00143      inline void set_threshold(float threshold)
00144      {
00145          d_threshold = threshold;
00146      }
00147
00148      /*!
00149       * \brief Set maximum Doppler grid search
00150       * \param doppler_max - Maximum Doppler shift considered in the grid search [Hz].
00151       */
00152      inline void set_doppler_max(unsigned int doppler_max)
00153      {
00154          d_config_doppler_max = doppler_max;
00155      }
00156
00157      /*!
00158       * \brief Set Doppler steps for the grid search

```

```

00159     * \param doppler_step - Frequency bin of the search grid [Hz].
00160     */
00161     void set_doppler_step(unsigned int doppler_step);
00162
00163     /*!
00164     * \brief If set to 1, ensures that acquisition starts at the
00165     * first available sample.
00166     * \param state - int=1 forces start of acquisition
00167     */
00168     void set_state(int state);
00169
00170     /*!
00171     * \brief Obtains the next power of 2 greater or equal to the input parameter
00172     * \param n - Integer value to obtain the next power of 2.
00173     */
00174     unsigned int nextPowerOf2(unsigned int n);
00175
00176     void dump_results(int effective_fft_size);
00177
00178     void forecast(int noutput_items, gr_vector_int& ninput_items_required);
00179
00180     /*!
00181     * \brief Parallel Code Phase Search Acquisition signal processing.
00182     */
00183     int general_work(int noutput_items, gr_vector_int& ninput_items,
00184                     gr_vector_const_void_star& input_items,
00185                     gr_vector_void_star& output_items);
00186
00187 private:
00188     friend pcps_acquisition_fine_doppler_cc_sptr pcps_make_acquisition_fine_doppler_cc(const Acq_Conf&
00189 conf_);
00189     explicit pcps_acquisition_fine_doppler_cc(const Acq_Conf& conf_);
00190
00191     int compute_and_accumulate_grid(gr_vector_const_void_star& input_items);
00192     int estimate_Doppler();
00193     float estimate_input_power(gr_vector_const_void_star& input_items);
00194     float compute_CAF();
00195     void reset_grid();
00196     void update_carrier_wipeoff();
00197     bool start();
00198
00199     std::weak_ptr<ChannelFsm> d_channel_fsm;
00200     std::unique_ptr<gnss_fft_complex_fwd> d_fft_if;
00201     std::unique_ptr<gnss_fft_complex_rev> d_ifft;
00202
00203     volk_gnssssdr::vector<volk_gnssssdr::vector<std::complex<float>>> d_grid_doppler_wipeoffs;
00204     volk_gnssssdr::vector<volk_gnssssdr::vector<float> d_grid_data;
00205     volk_gnssssdr::vector<gr_complex> d_fft_codes;
00206     volk_gnssssdr::vector<gr_complex> d_l0_ms_buffer;
00207     volk_gnssssdr::vector<float> d_magnitude;
00208
00209     arma::fmat grid_;
00210
00211     std::string d_satellite_str;
00212     std::string d_dump_filename;
00213
00214     Gnss_Synchro* d_gnss_synchro;
00215
00216     Acq_Conf acq_parameters;
00217
00218     int64_t d_fs_in;
00219     int64_t d_dump_number;
00220     uint64_t d_sample_counter;
00221
00222     float d_threshold;
00223     float d_test_statistics;
00224
00225     int d_positive_acq;
00226     int d_state;
00227     int d_samples_per_ms;
00228     int d_max_dwells;
00229     int d_config_doppler_max;
00230     int d_num_doppler_points;
00231     int d_well_count;
00232     int d_n_samples_in_buffer;
00233     int d_fft_size;
00234     int d_gnuradio_forecast_samples;
00235     unsigned int d_doppler_step;
00236     unsigned int d_channel;
00237     unsigned int d_dump_channel;
00238
00239     bool d_active;
00240     bool d_dump;
00241 };
00242
00243
00244 /** \} */

```

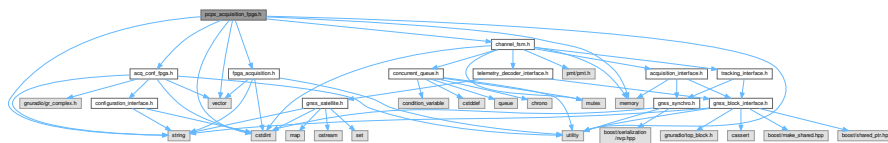
```
00245 /** \} */
00246 #endif // GNSS_SDR_PCPS_ACQUISITION_FINE_DOPPLER_CC_H
```

## 11.63 pcps\_acquisition\_fpga.h File Reference

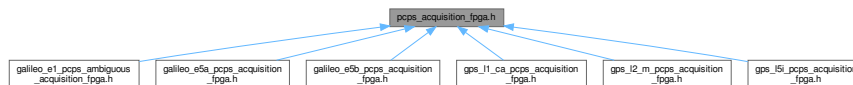
This class implements a Parallel Code Phase Search Acquisition for the FPGA.

```
#include "acq_conf_fpga.h"
#include "channel_fsm.h"
#include "fpga_acquisition.h"
#include <stdint>
#include <memory>
#include <string>
#include <utility>
#include <vector>
```

Include dependency graph for pcps\_acquisition\_fpga.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [pcps\\_acquisition\\_fpga](#)

*This class implements a Parallel Code Phase Search Acquisition that uses the FPGA.*

### Typedefs

- using [pcps\\_acquisition\\_fpga\\_sptr](#) = std::shared\_ptr<[pcps\\_acquisition\\_fpga](#)>

### Functions

- [pcps\\_acquisition\\_fpga\\_sptr pcps\\_make\\_acquisition\\_fpga](#) ([Acq\\_Conf\\_Fpga](#) \*conf\_, uint32\_t acq\_buff\_↔ num, std::vector< std::pair< uint32\_t, uint32\_t > > &downsampling\_filter\_specs, uint32\_t &max\_FFT\_size)

#### 11.63.1 Detailed Description

This class implements a Parallel Code Phase Search Acquisition for the FPGA.

Kay Borre book: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen, "A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach", Birkhauser, 2007. pp 81-84

#### Authors

- Marc Majoral, 2019. mmajoral(at)cttc.es
- Javier Arribas, 2019. jarribas(at)cttc.es

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [pcps\\_acquisition\\_fpga.h](#).

## 11.64 pcps\_acquisition\_fpga.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file pcps_acquisition_fpga.h
00003  * \brief This class implements a Parallel Code Phase Search Acquisition for the FPGA
00004  *
00005  *
00006  * Kay Borre book: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen,
00007  * "A Software-Defined GPS and Galileo Receiver. A Single-Frequency
00008  * Approach", Birkhauser, 2007. pp 81-84
00009  *
00010  * \authors <ul>
00011  *     <li> Marc Majoral, 2019. mmajoral(at)cttc.es
00012  *     <li> Javier Arribas, 2019. jarribas(at)cttc.es
00013  *     </ul>
00014  *
00015  * -----
00016  *
00017  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00018  * This file is part of GNSS-SDR.
00019  *
00020  * Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors)
00021  * SPDX-License-Identifier: GPL-3.0-or-later
00022  *
00023  * -----
00024  */
00025
00026 #ifndef GNSS_SDR_PCPS_ACQUISITION_FPGA_H
00027 #define GNSS_SDR_PCPS_ACQUISITION_FPGA_H
00028
00029 #include "acq_conf_fpga.h"
00030 #include "channel_fsm.h"
00031 #include "fpga_acquisition.h"
00032 #include <stdint> // for uint32_t
00033 #include <memory> // for shared_ptr
00034 #include <string> // for string
00035 #include <utility> // for for std::move, std::pair
00036 #include <vector> // for std::vector
00037
00038 /** \addtogroup Acquisition
00039  * \{ */
00040 /** \addtogroup Acq_gnuradio_blocks
00041  * \{ */
00042
00043
00044 class Gnss_Synchro;
00045
00046 class pcps_acquisition_fpga;
00047
00048 using pcps_acquisition_fpga_sptr = std::shared_ptr<pcps_acquisition_fpga>;
00049
00050 pcps_acquisition_fpga_sptr pcps_make_acquisition_fpga(Acq_Conf_Fpga* conf_, uint32_t acq_buff_num,
std::vector<std::pair<uint32_t, uint32_t>& downsampling_filter_specs, uint32_t& max_FFT_size);
00051
00052 /*!
00053  * \brief This class implements a Parallel Code Phase Search Acquisition that uses the FPGA.
00054  *
00055  * Check \ref Navitec2012 "An Open Source Galileo E1 Software Receiver",
00056  * Algorithm 1, for a pseudocode description of this implementation.
00057  */
00058 class pcps_acquisition_fpga
00059 {
00060 public:
00061     /*!
00062     * \brief Destructor
00063     */
00064     ~pcps_acquisition_fpga() = default;
00065
00066     /*!
00067     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00068     * to exchange synchronization data between acquisition and tracking blocks.
00069     * \param p_gnss_synchro Satellite information shared by the processing blocks.
00070     */
00071     inline void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro)
00072     {
00073         d_gnss_synchro = p_gnss_synchro;
00074     }
00075
00076     /*!
00077     * \brief Returns the maximum peak of grid search.
00078     */
00079     inline uint32_t mag() const
00080     {
00081         return d_mag;
00082     }

```

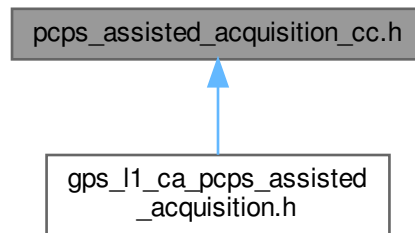
```

00083
00084     /*!
00085      * \brief Initializes acquisition algorithm.
00086      */
00087     void init();
00088
00089     /*!
00090      * \brief Sets local code for PCPS acquisition algorithm.
00091      */
00092     void set_local_code();
00093
00094     /*!
00095      * \brief If set to 1, ensures that acquisition starts at the
00096      * first available sample.
00097      * \param state - int=1 forces start of acquisition
00098      */
00099     void set_state(int32_t state);
00100
00101     /*!
00102      * \brief Starts acquisition algorithm, turning from standby mode to
00103      * active mode
00104      * \param active - bool that activates/deactivates the block.
00105      */
00106     void set_active(bool active);
00107
00108     /*!
00109      * \brief Set acquisition channel unique ID
00110      * \param channel - receiver channel.
00111      */
00112     inline void set_channel(uint32_t channel)
00113     {
00114         d_channel = channel;
00115     }
00116
00117     /*!
00118      * \brief Set channel fsm associated to this acquisition instance
00119      */
00120     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm)
00121     {
00122         d_channel_fsm = std::move(channel_fsm);
00123     }
00124
00125     /*!
00126      * \brief Set statistics threshold of PCPS algorithm.
00127      * \param threshold - Threshold for signal detection (check \ref Navitec2012,
00128      * Algorithm 1, for a definition of this threshold).
00129      */
00130     inline void set_threshold(float threshold)
00131     {
00132         d_threshold = threshold;
00133     }
00134
00135     /*!
00136      * \brief Set maximum Doppler grid search
00137      * \param doppler_max - Maximum Doppler shift considered in the grid search [Hz].
00138      */
00139     inline void set_doppler_max(uint32_t doppler_max)
00140     {
00141         d_doppler_max = doppler_max;
00142         d_acquisition_fpga->set_doppler_max(doppler_max);
00143     }
00144
00145     /*!
00146      * \brief Set Doppler steps for the grid search
00147      * \param doppler_step - Frequency bin of the search grid [Hz].
00148      */
00149     inline void set_doppler_step(uint32_t doppler_step)
00150     {
00151         d_doppler_step = doppler_step;
00152         d_acquisition_fpga->set_doppler_step(doppler_step);
00153     }
00154
00155     /*!
00156      * \brief Set Doppler center frequency for the grid search. It will refresh the Doppler grid.
00157      * \param doppler_center - Frequency center of the search grid [Hz].
00158      */
00159     void set_doppler_center(int32_t doppler_center);
00160
00161     /*!
00162      * \brief This function triggers a HW reset of the FPGA PL.
00163      */
00164     void reset_acquisition();
00165
00166     /*!
00167      * \brief stop the acquisition and the other FPGA modules.
00168      */
00169     void stop_acquisition();

```



This graph shows which files directly or indirectly include this file:



## Classes

- class [pcps\\_assisted\\_acquisition\\_cc](#)  
*This class implements a Parallel Code Phase Search Acquisition.*

## Typedefs

- using [pcps\\_assisted\\_acquisition\\_cc\\_sptr](#) = gnss\_shared\_ptr<[pcps\\_assisted\\_acquisition\\_cc](#)>

## Functions

- [pcps\\_assisted\\_acquisition\\_cc\\_sptr pcps\\_make\\_assisted\\_acquisition\\_cc](#) (int32\_t max\_dwells, uint32\_t sampled\_ms, int32\_t doppler\_max, int32\_t doppler\_min, int64\_t fs\_in, int32\_t samples\_per\_ms, bool dump, const std::string &dump\_filename, bool enable\_monitor\_output)

### 11.65.1 Detailed Description

This class implements a Parallel Code Phase Search Acquisition with assistance and multi-dwells. Acquisition strategy (Kay Borre book + CFAR threshold).

1. Compute the input signal power estimation
2. Doppler serial search loop
3. Perform the FFT-based circular convolution (parallel time search)
4. Record the maximum peak and the associated synchronization parameters
5. Compute the test statistics and compare to the threshold
6. Declare positive or negative acquisition using a message queue

Kay Borre book: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen, "A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach", Birkhauser, 2007. pp 81-84

## Authors

- Javier Arribas, 2013. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR. Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later Definition in file [pcps\\_assisted\\_acquisition\\_cc.h](#).

---

## 11.66 pcps\_assisted\_acquisition\_cc.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file pcps_assisted_acquisition_cc.h
00003  * \brief This class implements a Parallel Code Phase Search Acquisition with assistance and
      multi-dwells
00004  *
00005  * Acquisition strategy (Kay Borre book + CFAR threshold).
00006  * <ol>
00007  * <li> Compute the input signal power estimation
00008  * <li> Doppler serial search loop
00009  * <li> Perform the FFT-based circular convolution (parallel time search)
00010  * <li> Record the maximum peak and the associated synchronization parameters
00011  * <li> Compute the test statistics and compare to the threshold
00012  * <li> Declare positive or negative acquisition using a message queue
00013  * </ol>
00014  *
00015  * Kay Borre book: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen,
00016  * "A Software-Defined GPS and Galileo Receiver. A Single-Frequency
00017  * Approach", Birkhauser, 2007. pp 81-84
00018  *
00019  * \authors <ul>
00020  * <li> Javier Arribas, 2013. jarribas(at)cttc.es
00021  * </ul>
00022  *
00023  * -----
00024  *
00025  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00026  * This file is part of GNSS-SDR.
00027  *
00028  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00029  * SPDX-License-Identifier: GPL-3.0-or-later
00030  *
00031  * -----
00032  */
00033
00034 #ifndef GNSS_SDR_PCPS_ASSISTED_ACQUISITION_CC_H
00035 #define GNSS_SDR_PCPS_ASSISTED_ACQUISITION_CC_H
00036
00037 #include "channel_fsm.h"
00038 #include "gnss_sdr_fft.h"
00039 #include "gnss_synchro.h"
00040 #include <gnuradio/block.h>
00041 #include <gnuradio/gr_complex.h>
00042 #include <fstream>
00043 #include <memory>
00044 #include <string>
00045 #include <utility>
00046 #include <vector>
00047
00048 /** \addtogroup Acquisition
00049  * \{ */
00050 /** \addtogroup Acq_gnuradio_blocks
00051  * \{ */
00052
00053
00054 class pcps_assisted_acquisition_cc;
00055
00056 using pcps_assisted_acquisition_cc_sptr = gnss_shared_ptr<pcps_assisted_acquisition_cc>;
00057
00058 pcps_assisted_acquisition_cc_sptr pcps_make_assisted_acquisition_cc(
00059     int32_t max_dwells,
00060     uint32_t sampled_ms,
00061     int32_t doppler_max,
00062     int32_t doppler_min,
00063     int64_t fs_in,
00064     int32_t samples_per_ms,
00065     bool dump,
00066     const std::string& dump_filename,
00067     bool enable_monitor_output);
00068
00069 /*!
00070 * \brief This class implements a Parallel Code Phase Search Acquisition.
00071 *
00072 * Check \ref Navitec2012 "An Open Source Galileo E1 Software Receiver",
00073 * Algorithm 1, for a pseudocode description of this implementation.
00074 */
00075 class pcps_assisted_acquisition_cc : public gr::block
00076 {
00077 public:
00078     /*!
00079     * \brief Default destructor.
00080     */
00081     ~pcps_assisted_acquisition_cc();
00082

```



```

00083     /*!
00084     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00085     * to exchange synchronization data between acquisition and tracking blocks.
00086     * \param p_gnss_synchro Satellite information shared by the processing blocks.
00087     */
00088     inline void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro)
00089     {
00090         d_gnss_synchro = p_gnss_synchro;
00091     }
00092
00093     /*!
00094     * \brief Returns the maximum peak of grid search.
00095     */
00096     inline uint32_t mag() const
00097     {
00098         return d_test_statistics;
00099     }
00100
00101     /*!
00102     * \brief Initializes acquisition algorithm.
00103     */
00104     void init();
00105
00106     /*!
00107     * \brief Sets local code for PCPS acquisition algorithm.
00108     * \param code - Pointer to the PRN code.
00109     */
00110     void set_local_code(std::complex<float>* code);
00111
00112     /*!
00113     * \brief Starts acquisition algorithm, turning from standby mode to
00114     * active mode
00115     * \param active - bool that activates/deactivates the block.
00116     */
00117     inline void set_active(bool active)
00118     {
00119         d_active = active;
00120     }
00121
00122     /*!
00123     * \brief Set acquisition channel unique ID
00124     * \param channel - receiver channel.
00125     */
00126     inline void set_channel(uint32_t channel)
00127     {
00128         d_channel = channel;
00129     }
00130
00131     /*!
00132     * \brief Set channel fsm associated to this acquisition instance
00133     */
00134     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm)
00135     {
00136         d_channel_fsm = std::move(channel_fsm);
00137     }
00138
00139     /*!
00140     * \brief Set statistics threshold of PCPS algorithm.
00141     * \param threshold - Threshold for signal detection (check \ref Navitec2012,
00142     * Algorithm 1, for a definition of this threshold).
00143     */
00144     inline void set_threshold(float threshold)
00145     {
00146         d_threshold = threshold;
00147     }
00148
00149     inline void set_state(int32_t state)
00150     {
00151         d_state = state;
00152     }
00153
00154     /*!
00155     * \brief Set maximum Doppler grid search
00156     * \param doppler_max - Maximum Doppler shift considered in the grid search [Hz].
00157     */
00158     inline void set_doppler_max(uint32_t doppler_max)
00159     {
00160         d_doppler_max = doppler_max;
00161     }
00162
00163     /*!
00164     * \brief Set Doppler steps for the grid search
00165     * \param doppler_step - Frequency bin of the search grid [Hz].
00166     */
00167     void set_doppler_step(uint32_t doppler_step);
00168
00169     /*!

```

```

00170     * \brief Parallel Code Phase Search Acquisition signal processing.
00171     */
00172     int general_work(int noutput_items, gr_vector_int& ninput_items,
00173                     gr_vector_const_void_star& input_items,
00174                     gr_vector_void_star& output_items);
00175
00176     void forecast(int noutput_items, gr_vector_int& ninput_items_required);
00177
00178 private:
00179     friend pcps_assisted_acquisition_cc_sptr
00180     pcps_make_assisted_acquisition_cc(int32_t max_dwells, uint32_t sampled_ms,
00181                                       int32_t doppler_max, int32_t doppler_min, int64_t fs_in,
00182                                       int32_t samples_per_ms, bool dump,
00183                                       const std::string& dump_filename, bool enable_monitor_output);
00184
00185     pcps_assisted_acquisition_cc(int32_t max_dwells, uint32_t sampled_ms,
00186                                   int32_t doppler_max, int32_t doppler_min, int64_t fs_in,
00187                                   int32_t samples_per_ms, bool dump,
00188                                   const std::string& dump_filename, bool enable_monitor_output);
00189
00190     void calculate_magnitudes(gr_complex* fft_begin, int32_t doppler_shift,
00191                              int32_t doppler_offset);
00192
00193     int32_t compute_and_accumulate_grid(gr_vector_const_void_star& input_items);
00194     float estimate_input_power(gr_vector_const_void_star& input_items) const;
00195     float search_maximum();
00196     void get_assistance();
00197     void reset_grid();
00198     void redefine_grid();
00199
00200     std::weak_ptr<ChannelFsm> d_channel_fsm;
00201     std::unique_ptr<gnss_fft_complex_fwd> d_fft_if;
00202     std::unique_ptr<gnss_fft_complex_rev> d_ifft;
00203
00204     std::vector<std::vector<std::complex<float>>> d_grid_doppler_wipeoffs;
00205     std::vector<std::vector<float>> d_grid_data;
00206     std::vector<gr_complex> d_fft_codes;
00207
00208     std::string d_satellite_str;
00209     std::string d_dump_filename;
00210
00211     std::ofstream d_dump_file;
00212
00213     Gnss_Synchro* d_gnss_synchro;
00214
00215     int64_t d_fs_in;
00216     uint64_t d_sample_counter;
00217
00218     float d_threshold;
00219     float d_doppler_freq;
00220     float d_input_power;
00221     float d_test_statistics;
00222
00223     uint32_t d_doppler_resolution;
00224     uint32_t d_channel;
00225     uint32_t d_sampled_ms;
00226     uint32_t d_code_phase;
00227
00228     int32_t d_samples_per_ms;
00229
00230     uint32_t d_fft_size;
00231
00232     int32_t d_max_dwells;
00233     int32_t d_gnuradio_forecast_samples;
00234     int32_t d_doppler_max;
00235     int32_t d_doppler_min;
00236     int32_t d_config_doppler_max;
00237     int32_t d_config_doppler_min;
00238     int32_t d_num_doppler_points;
00239     int32_t d_doppler_step;
00240     int32_t d_state;
00241     int32_t d_well_count;
00242
00243     bool d_active;
00244     bool d_disable_assist;
00245     bool d_dump;
00246     bool d_enable_monitor_output;
00247 };
00248
00249
00250 /** \} */
00251 /** \} */
00252 #endif // GNSS_SDR_PCPS_ASSISTED_ACQUISITION_CC_H

```



### 11.67.1 Detailed Description

This class implements a Parallel Code Phase Search acquisition with Coherent [Channel](#) Combining With Sign Recovery scheme.

#### Author

Marc Molina, 2013. marc.molina.pena(at)gmail.com

D.Borio, C.O'Driscoll, G.Lachapelle, "Coherent, Noncoherent and Differentially Coherent Combining Techniques for Acquisition of New Composite GNSS Signals", IEEE Transactions On Aerospace and Electronic Systems vol. 45 no. 3, July 2009, section IV

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [pcps\\_cccwsr\\_acquisition\\_cc.h](#).

## 11.68 pcps\_cccwsr\_acquisition\_cc.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file pcps_cccwsr_acquisition_cc.h
00003  * \brief This class implements a Parallel Code Phase Search acquisition
00004  * with Coherent Channel Combining With Sign Recovery scheme.
00005  * \author Marc Molina, 2013. marc.molina.pena(at)gmail.com
00006  *
00007  * D.Borio, C.O'Driscoll, G.Lachapelle, "Coherent, Noncoherent and
00008  * Differentially Coherent Combining Techniques for Acquisition of
00009  * New Composite GNSS Signals", IEEE Transactions On Aerospace and
00010  * Electronic Systems vol. 45 no. 3, July 2009, section IV
00011  *
00012  * -----
00013  *
00014  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00015  * This file is part of GNSS-SDR.
00016  *
00017  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00018  * SPDX-License-Identifier: GPL-3.0-or-later
00019  *
00020  * -----
00021  */
00022
00023 #ifndef GNSS_SDR_PCPS_CCCWSR_ACQUISITION_CC_H
00024 #define GNSS_SDR_PCPS_CCCWSR_ACQUISITION_CC_H
00025
00026 #include "channel_fsm.h"
00027 #include "gnss_sdr_fft.h"
00028 #include "gnss_synchro.h"
00029 #include <gnuradio/block.h>
00030 #include <gnuradio/gr_complex.h>
00031 #include <fstream>
00032 #include <memory>
00033 #include <string>
00034 #include <utility>
00035 #include <vector>
00036
00037 /** \addtogroup Acquisition
00038  * \{ */
00039 /** \addtogroup Acq_gnuradio_blocks
00040  * \{ */
00041
00042
00043 class pcps_cccwsr_acquisition_cc;
00044
00045 using pcps_cccwsr_acquisition_cc_sptr = gnss_shared_ptr<pcps_cccwsr_acquisition_cc>;
00046
00047 pcps_cccwsr_acquisition_cc_sptr pcps_cccwsr_make_acquisition_cc(
00048     uint32_t sampled_ms,
00049     uint32_t max_dwells,
00050     uint32_t doppler_max,
00051     int64_t fs_in,
00052     int32_t samples_per_ms,
00053     int32_t samples_per_code,
00054     bool dump,
00055     const std::string& dump_filename,
00056     bool enable_monitor_output);
00057
00058 /*!
00059  * \brief This class implements a Parallel Code Phase Search Acquisition with
00060  * Coherent Channel Combining With Sign Recovery scheme.
00061  */
```

```

00062 class pcps_cccwsr_acquisition_cc : public gr::block
00063 {
00064 public:
00065     /*!
00066      * \brief Default destructor.
00067      */
00068     ~pcps_cccwsr_acquisition_cc();
00069
00070     /*!
00071      * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00072      * to exchange synchronization data between acquisition and tracking blocks.
00073      * \param p_gnss_synchro Satellite information shared by the processing blocks.
00074      */
00075     inline void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro)
00076     {
00077         d_gnss_synchro = p_gnss_synchro;
00078     }
00079
00080     /*!
00081      * \brief Returns the maximum peak of grid search.
00082      */
00083     inline uint32_t mag() const
00084     {
00085         return d_mag;
00086     }
00087
00088     /*!
00089      * \brief Initializes acquisition algorithm.
00090      */
00091     void init();
00092
00093     /*!
00094      * \brief Sets local code for CCCWSR acquisition algorithm.
00095      * \param data_code - Pointer to the data PRN code.
00096      * \param pilot_code - Pointer to the pilot PRN code.
00097      */
00098     void set_local_code(std::complex<float>* code_data, std::complex<float>* code_pilot);
00099
00100     /*!
00101      * \brief Starts acquisition algorithm, turning from standby mode to
00102      * active mode
00103      * \param active - bool that activates/deactivates the block.
00104      */
00105     inline void set_active(bool active)
00106     {
00107         d_active = active;
00108     }
00109
00110     /*!
00111      * \brief If set to 1, ensures that acquisition starts at the
00112      * first available sample.
00113      * \param state - int=1 forces start of acquisition
00114      */
00115     void set_state(int32_t state);
00116
00117     /*!
00118      * \brief Set acquisition channel unique ID
00119      * \param channel - receiver channel.
00120      */
00121     inline void set_channel(uint32_t channel)
00122     {
00123         d_channel = channel;
00124     }
00125
00126     /*!
00127      * \brief Set channel fsm associated to this acquisition instance
00128      */
00129     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm)
00130     {
00131         d_channel_fsm = std::move(channel_fsm);
00132     }
00133
00134     /*!
00135      * \brief Set statistics threshold of CCCWSR algorithm.
00136      * \param threshold - Threshold for signal detection (check \ref Navitec2012,
00137      * Algorithm 1, for a definition of this threshold).
00138      */
00139     inline void set_threshold(float threshold)
00140     {
00141         d_threshold = threshold;
00142     }
00143
00144     /*!
00145      * \brief Set maximum Doppler grid search
00146      * \param doppler_max - Maximum Doppler shift considered in the grid search [Hz].
00147      */
00148     inline void set_doppler_max(uint32_t doppler_max)

```

```

00149     {
00150         d_doppler_max = doppler_max;
00151     }
00152
00153     /*!
00154     * \brief Set Doppler steps for the grid search
00155     * \param doppler_step - Frequency bin of the search grid [Hz].
00156     */
00157     inline void set_doppler_step(uint32_t doppler_step)
00158     {
00159         d_doppler_step = doppler_step;
00160     }
00161
00162     /*!
00163     * \brief Coherent Channel Combining With Sign Recovery Acquisition signal processing.
00164     */
00165     int general_work(int noutput_items, gr_vector_int& ninput_items,
00166                     gr_vector_const_void_star& input_items,
00167                     gr_vector_void_star& output_items);
00168
00169 private:
00170     friend pcps_cccwsr_acquisition_cc_sptr
00171     pcps_cccwsr_make_acquisition_cc(uint32_t sampled_ms, uint32_t max_dwells,
00172                                     uint32_t doppler_max, int64_t fs_in,
00173                                     int32_t samples_per_ms, int32_t samples_per_code,
00174                                     bool dump, const std::string& dump_filename, bool enable_monitor_output);
00175
00176     pcps_cccwsr_acquisition_cc(uint32_t sampled_ms, uint32_t max_dwells,
00177                                 uint32_t doppler_max, int64_t fs_in,
00178                                 int32_t samples_per_ms, int32_t samples_per_code,
00179                                 bool dump, const std::string& dump_filename, bool enable_monitor_output);
00180
00181     void calculate_magnitudes(gr_complex* fft_begin, int32_t doppler_shift,
00182                              int32_t doppler_offset);
00183
00184     std::weak_ptr<ChannelFsm> d_channel_fsm;
00185
00186     std::unique_ptr<gnss_fft_complex_fwd> d_fft_if;
00187     std::unique_ptr<gnss_fft_complex_rev> d_ifft;
00188
00189     std::vector<std::vector<gr_complex>> d_grid_doppler_wipeoffs;
00190     std::vector<gr_complex> d_fft_code_data;
00191     std::vector<gr_complex> d_fft_code_pilot;
00192     std::vector<gr_complex> d_data_correlation;
00193     std::vector<gr_complex> d_pilot_correlation;
00194     std::vector<gr_complex> d_correlation_plus;
00195     std::vector<gr_complex> d_correlation_minus;
00196     std::vector<float> d_magnitude;
00197
00198     std::ofstream d_dump_file;
00199     std::string d_satellite_str;
00200     std::string d_dump_filename;
00201
00202     Gnss_Synchro* d_gnss_synchro;
00203
00204     int64_t d_fs_in;
00205     uint64_t d_sample_counter;
00206
00207     float d_threshold;
00208     float d_doppler_freq;
00209     float d_mag;
00210     float d_input_power;
00211     float d_test_statistics;
00212
00213     int32_t d_state;
00214     int32_t d_samples_per_ms;
00215     int32_t d_samples_per_code;
00216     uint32_t d_doppler_resolution;
00217     uint32_t d_doppler_max;
00218     uint32_t d_doppler_step;
00219     uint32_t d_sampled_ms;
00220     uint32_t d_max_dwells;
00221     uint32_t d_well_count;
00222     uint32_t d_fft_size;
00223     uint32_t d_num_doppler_bins;
00224     uint32_t d_code_phase;
00225     uint32_t d_channel;
00226
00227     bool d_active;
00228     bool d_dump;
00229     bool d_enable_monitor_output;
00230 };
00231
00232
00233 /** \} */
00234 /** \} */
00235 #endif // GNSS_SDR_PCPS_CCCWSR_ACQUISITION_CC_H

```

## 11.69 pcps\_openc1\_acquisition\_cc.h File Reference

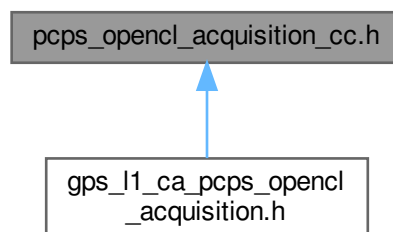
This class implements a Parallel Code Phase Search Acquisition using OpenCL to offload some functions to the GPU.

```
#include "channel_fsm.h"
#include "gnss_block_interface.h"
#include "gnss_sdr_fft.h"
#include "gnss_synchro.h"
#include "openc1/fft_internal.h"
#include <gnuradio/block.h>
#include <gnuradio/gr_complex.h>
#include "openc1/cl.hpp"
#include <stdint>
#include <fstream>
#include <memory>
#include <string>
#include <vector>
```

Include dependency graph for pcps\_openc1\_acquisition\_cc.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [pcps\\_openc1\\_acquisition\\_cc](#)

*This class implements a Parallel Code Phase Search Acquisition.*

### Typedefs

- using [pcps\\_openc1\\_acquisition\\_cc\\_sptr](#) = [gnss\\_shared\\_ptr](#)<[pcps\\_openc1\\_acquisition\\_cc](#)>

### Functions

- [pcps\\_openc1\\_acquisition\\_cc\\_sptr](#) **pcps\_make\_openc1\_acquisition\_cc** (uint32\_t sampled\_ms, uint32\_t max\_dwells, uint32\_t doppler\_max, int64\_t fs\_in, int samples\_per\_ms, int samples\_per\_code, bool bit\_transition\_flag, bool dump, const std::string &dump\_filename, bool enable\_monitor\_output)

### 11.69.1 Detailed Description

This class implements a Parallel Code Phase Search Acquisition using OpenCL to offload some functions to the GPU.

Acquisition strategy (Kay Borre book + CFAR threshold).

1. Compute the input signal power estimation
2. Doppler serial search loop
3. Perform the FFT-based circular convolution (parallel time search)
4. Record the maximum peak and the associated synchronization parameters
5. Compute the test statistics and compare to the threshold
6. Declare positive or negative acquisition using a message port

Kay Borre book: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen, "A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach", Birkhauser, 2007. pp 81-84

#### Authors

- Javier Arribas, 2011. jarribas(at)cttc.es
- Luis Esteve, 2012. luis(at)epsilon-formacion.com
- Marc Molina, 2013. [marc.molina.pena@gmail.com](mailto:marc.molina.pena@gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [pcps\\_openc1\\_acquisition\\_cc.h](#).

---

### 11.69.2 Macro Definition Documentation

#### 11.69.2.1 CL\_SILENCE\_DEPRECATION

```
#define CL_SILENCE_DEPRECATION
```

Definition at line 40 of file [pcps\\_openc1\\_acquisition\\_cc.h](#).

## 11.70 pcps\_openc1\_acquisition\_cc.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file pcps_openc1_acquisition_cc.h
00003  * \brief This class implements a Parallel Code Phase Search Acquisition
00004  * using OpenCL to offload some functions to the GPU.
00005  *
00006  * Acquisition strategy (Kay Borre book + CFAR threshold).
00007  * <ol>
00008  * <li> Compute the input signal power estimation
00009  * <li> Doppler serial search loop
00010  * <li> Perform the FFT-based circular convolution (parallel time search)
00011  * <li> Record the maximum peak and the associated synchronization parameters
00012  * <li> Compute the test statistics and compare to the threshold
00013  * <li> Declare positive or negative acquisition using a message port
00014  * </ol>
00015  *
00016  * Kay Borre book: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen,
00017  * "A Software-Defined GPS and Galileo Receiver. A Single-Frequency
00018  * Approach", Birkhauser, 2007. pp 81-84
00019  *
00020  * \authors <ul>
00021  * <li> Javier Arribas, 2011. jarribas(at)cttc.es
00022  * <li> Luis Esteve, 2012. luis(at)epsilon-formacion.com
00023  * <li> Marc Molina, 2013. marc.molina.pena@gmail.com
00024  * </ul>
00025  *
00026  * -----
00027  *
00028  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00029  * This file is part of GNSS-SDR.
00030  *
00031  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
```



```

00032  * SPDX-License-Identifier: GPL-3.0-or-later
00033  *
00034  * -----
00035  */
00036
00037 #ifndef GNSS_SDR_PCPS_OPENC1_ACQUISITION_CC_H
00038 #define GNSS_SDR_PCPS_OPENC1_ACQUISITION_CC_H
00039
00040 #define CL_SILENCE_DEPRECATION
00041 #include "channel_fsm.h"
00042 #include "gnss_block_interface.h"
00043 #include "gnss_sdr_fft.h"
00044 #include "gnss_synchro.h"
00045 #include "openc1/fft_internal.h"
00046 #include <gnuradio/block.h>
00047 #include <gnuradio/gr_complex.h>
00048 #include "openc1/cl.hpp"
00049 #include <stdint>
00050 #include <fstream>
00051 #include <memory> // for weak_ptr
00052 #include <string>
00053 #include <vector>
00054
00055 /** \addtogroup Acquisition
00056  * \{ */
00057 /** \addtogroup Acq_gnuradio_blocks
00058  * \{ */
00059
00060
00061 class pcps_openc1_acquisition_cc;
00062
00063 using pcps_openc1_acquisition_cc_sptr = gnss_shared_ptr<pcps_openc1_acquisition_cc>;
00064
00065 pcps_openc1_acquisition_cc_sptr pcps_make_openc1_acquisition_cc(
00066     uint32_t sampled_ms,
00067     uint32_t max_dwells,
00068     uint32_t doppler_max,
00069     int64_t fs_in,
00070     int samples_per_ms,
00071     int samples_per_code,
00072     bool bit_transition_flag,
00073     bool dump,
00074     const std::string& dump_filename,
00075     bool enable_monitor_output);
00076
00077 /*!
00078  * \brief This class implements a Parallel Code Phase Search Acquisition.
00079  *
00080  * Check \ref Navitex2012 "An Open Source Galileo E1 Software Receiver",
00081  * Algorithm 1, for a pseudocode description of this implementation.
00082  */
00083 class pcps_openc1_acquisition_cc : public gr::block
00084 {
00085 public:
00086     /*!
00087      * \brief Default destructor.
00088      */
00089     ~pcps_openc1_acquisition_cc();
00090
00091     /*!
00092      * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00093      * to exchange synchronization data between acquisition and tracking blocks.
00094      * \param p_gnss_synchro Satellite information shared by the processing blocks.
00095      */
00096     inline void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro)
00097     {
00098         d_gnss_synchro = p_gnss_synchro;
00099     }
00100
00101     /*!
00102      * \brief Returns the maximum peak of grid search.
00103      */
00104     inline uint32_t mag() const
00105     {
00106         return d_mag;
00107     }
00108
00109     /*!
00110      * \brief Initializes acquisition algorithm.
00111      */
00112     void init();
00113
00114     /*!
00115      * \brief Sets local code for PCPS acquisition algorithm.
00116      * \param code - Pointer to the PRN code.
00117      */
00118     void set_local_code(std::complex<float>* code);

```

```

00119
00120     /*!
00121     * \brief Starts acquisition algorithm, turning from standby mode to
00122     * active mode
00123     * \param active - bool that activates/deactivates the block.
00124     */
00125     inline void set_active(bool active)
00126     {
00127         d_active = active;
00128     }
00129
00130     /*!
00131     * \brief If set to 1, ensures that acquisition starts at the
00132     * first available sample.
00133     * \param state - int=1 forces start of acquisition
00134     */
00135     void set_state(int state);
00136
00137     /*!
00138     * \brief Set acquisition channel unique ID
00139     * \param channel - receiver channel.
00140     */
00141     inline void set_channel(uint32_t channel)
00142     {
00143         d_channel = channel;
00144     }
00145
00146     /*!
00147     * \brief Set channel fsm associated to this acquisition instance
00148     */
00149     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm)
00150     {
00151         d_channel_fsm = channel_fsm;
00152     }
00153
00154     /*!
00155     * \brief Set statistics threshold of PCPS algorithm.
00156     * \param threshold - Threshold for signal detection (check \ref Navitec2012,
00157     * Algorithm 1, for a definition of this threshold).
00158     */
00159     inline void set_threshold(float threshold)
00160     {
00161         d_threshold = threshold;
00162     }
00163
00164     /*!
00165     * \brief Set maximum Doppler grid search
00166     * \param doppler_max - Maximum Doppler shift considered in the grid search [Hz].
00167     */
00168     inline void set_doppler_max(uint32_t doppler_max)
00169     {
00170         d_doppler_max = doppler_max;
00171     }
00172
00173     /*!
00174     * \brief Set Doppler steps for the grid search
00175     * \param doppler_step - Frequency bin of the search grid [Hz].
00176     */
00177     inline void set_doppler_step(uint32_t doppler_step)
00178     {
00179         d_doppler_step = doppler_step;
00180     }
00181
00182     inline bool opencl_ready() const
00183     {
00184         bool ready = false;
00185         if (d_opencl == 0)
00186         {
00187             ready = true;
00188         }
00189         return ready;
00190     }
00191
00192     void acquisition_core_volk();
00193
00194     void acquisition_core_opencl();
00195
00196     /*!
00197     * \brief Parallel Code Phase Search Acquisition signal processing.
00198     */
00199     int general_work(int noutput_items, gr_vector_int& ninput_items,
00200                     gr_vector_const_void_star& input_items,
00201                     gr_vector_void_star& output_items);
00202
00203 private:
00204     friend pcps_opencl_acquisition_cc_sptr
00205     pcps_make_opencl_acquisition_cc(uint32_t sampled_ms, uint32_t max_dwells,

```

```

00206         uint32_t doppler_max, int64_t fs_in,
00207         int samples_per_ms, int samples_per_code,
00208         bool bit_transition_flag,
00209         bool dump,
00210         const std::string& dump_filename,
00211         bool enable_monitor_output);
00212
00213     pcps_openc1_acquisition_cc(uint32_t sampled_ms, uint32_t max_dwells,
00214         uint32_t doppler_max, int64_t fs_in,
00215         int samples_per_ms, int samples_per_code,
00216         bool bit_transition_flag,
00217         bool dump,
00218         const std::string& dump_filename,
00219         bool enable_monitor_output);
00220
00221     void calculate_magnitudes(gr_complex* fft_begin, int doppler_shift,
00222         int doppler_offset);
00223
00224     int init_openc1_environment(const std::string& kernel_filename);
00225
00226     cl::Platform d_cl_platform;
00227     cl::Device d_cl_device;
00228     cl::Context d_cl_context;
00229     cl::Program d_cl_program;
00230     cl::Buffer* d_cl_buffer_in;
00231     cl::Buffer* d_cl_buffer_fft_codes;
00232     cl::Buffer* d_cl_buffer_1;
00233     cl::Buffer* d_cl_buffer_2;
00234     cl::Buffer* d_cl_buffer_magnitude;
00235     cl::Buffer** d_cl_buffer_grid_doppler_wipeoffs;
00236     cl::CommandQueue* d_cl_queue;
00237     clFFT_Plan d_cl_fft_plan;
00238     cl_int d_cl_fft_batch_size;
00239
00240     std::weak_ptr<ChannelFsm> d_channel_fsm;
00241
00242     std::unique_ptr<gnss_fft_complex_fwd> d_fft_if;
00243     std::unique_ptr<gnss_fft_complex_rev> d_ifft;
00244
00245     std::vector<std::vector<gr_complex>> d_grid_doppler_wipeoffs;
00246     std::vector<std::vector<gr_complex>> d_in_buffer;
00247     std::vector<gr_complex> d_fft_codes;
00248     std::vector<gr_complex> d_zero_vector;
00249     std::vector<uint64_t> d_sample_counter_buffer;
00250     std::vector<float> d_magnitude;
00251
00252     std::string d_dump_filename;
00253     std::string d_satellite_str;
00254
00255     std::ofstream d_dump_file;
00256
00257     Gnss_Synchro* d_gnss_synchro;
00258
00259     int64_t d_fs_in;
00260     uint64_t d_sample_counter;
00261
00262     int* d_max_doppler_indexes;
00263
00264     float d_threshold;
00265     float d_doppler_freq;
00266     float d_mag;
00267     float d_input_power;
00268     float d_test_statistics;
00269
00270     int d_samples_per_ms;
00271     int d_samples_per_code;
00272     int d_state;
00273     int d_openc1;
00274
00275     uint32_t d_doppler_resolution;
00276     uint32_t d_doppler_max;
00277     uint32_t d_doppler_step;
00278     uint32_t d_sampled_ms;
00279     uint32_t d_max_dwells;
00280     uint32_t d_well_count;
00281     uint32_t d_fft_size;
00282     uint32_t d_fft_size_pow2;
00283     uint32_t d_num_doppler_bins;
00284     uint32_t d_code_phase;
00285     uint32_t d_channel;
00286     uint32_t d_in_dwell_count;
00287
00288     bool d_bit_transition_flag;
00289     bool d_active;
00290     bool d_core_working;
00291     bool d_dump;
00292     bool d_enable_monitor_output;

```

```

00293 };
00294
00295
00296 /** \} */
00297 /** \} */
00298 #endif // GNSS_SDR_PCPS_OPENCL_ACQUISITION_CC_H

```

## 11.71 pcps\_quicksync\_acquisition\_cc.h File Reference

This class implements a Parallel Code Phase Search Acquisition with the QuickSync Algorithm.

```

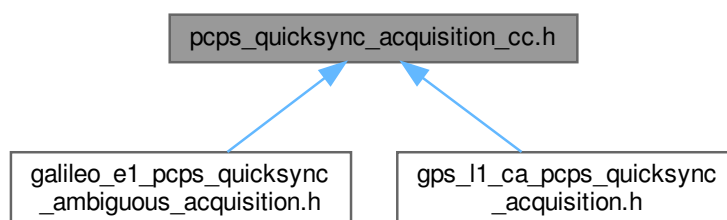
#include "channel_fsm.h"
#include "gnss_sdr_fft.h"
#include "gnss_synchro.h"
#include <gnuradio/block.h>
#include <gnuradio/gr_complex.h>
#include <algorithm>
#include <cassert>
#include <fstream>
#include <functional>
#include <memory>
#include <string>
#include <utility>
#include <vector>

```

Include dependency graph for pcps\_quicksync\_acquisition\_cc.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [pcps\\_quicksync\\_acquisition\\_cc](#)

*This class implements a Parallel Code Phase Search Acquisition with the implementation of the Sparse QuickSync Algorithm.*

### Typedefs

- using [pcps\\_quicksync\\_acquisition\\_cc\\_sptr](#) = `gnss_shared_ptr<pcps_quicksync_acquisition_cc>`

## Functions

- pcps\_quicksync\_acquisition\_cc\_sptr **pcps\_quicksync\_make\_acquisition\_cc** (uint32\_t folding\_factor, uint32\_t sampled\_ms, uint32\_t max\_dwells, uint32\_t doppler\_max, int64\_t fs\_in, int32\_t samples\_per\_ms, int32\_t samples\_per\_code, bool bit\_transition\_flag, bool dump, const std::string &dump\_filename, bool enable\_monitor\_output)

### 11.71.1 Detailed Description

This class implements a Parallel Code Phase Search Acquisition with the QuickSync Algorithm. Acquisition strategy (Kay Borre book CFAR + threshold).

1. Compute the input signal power estimation
2. Doppler serial search loop
3. Perform folding of the incoming signal and local generated code
4. Perform the FFT-based circular convolution (parallel time search)
5. Record the maximum peak and the associated synchronization parameters
6. Compute the test statistics and compare to the threshold
7. Declare positive or negative acquisition using a message port
8. Obtain the adequate acquisition parameters by correlating the incoming signal shifted by the possible folded delays

Kay Borre book: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen, "A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach", Birkha user, 2007. pp 81-84

#### Date

Jun2 2014

#### Author

Damian Miralles Sanchez, [dmiralles2009@gmail.com](mailto:dmiralles2009@gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [pcps\\_quicksync\\_acquisition\\_cc.h](#).

---

## 11.72 pcps\_quicksync\_acquisition\_cc.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file pcps_quicksync_acquisition_cc.h
00003  * \brief This class implements a Parallel Code Phase Search Acquisition with the
00004  * QuickSync Algorithm
00005  *
00006  * Acquisition strategy (Kay Borre book CFAR + threshold).
00007  * <ol>
00008  * <li> Compute the input signal power estimation
00009  * <li> Doppler serial search loop
00010  * <li> Perform folding of the incoming signal and local generated code
00011  * <li> Perform the FFT-based circular convolution (parallel time search)
00012  * <li> Record the maximum peak and the associated synchronization parameters
00013  * <li> Compute the test statistics and compare to the threshold
00014  * <li> Declare positive or negative acquisition using a message port
00015  * <li> Obtain the adequate acquisition parameters by correlating the incoming
00016  * signal shifted by the possible folded delays
00017  * </ol>
00018  *
00019  * Kay Borre book: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen,
00020  * "A Software-Defined GPS and Galileo Receiver. A Single-Frequency
00021  * Approach", Birkha user, 2007. pp 81-84
00022  *
00023  * \date Jun2 2014
00024  * \author Damian Miralles Sanchez, dmiralles2009@gmail.com
```

```

00025  *
00026  * -----
00027  *
00028  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00029  * This file is part of GNSS-SDR.
00030  *
00031  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00032  * SPDX-License-Identifier: GPL-3.0-or-later
00033  *
00034  * -----
00035  */
00036
00037 #ifndef GNSS_SDR_PCPS_QUICKSYNC_ACQUISITION_CC_H
00038 #define GNSS_SDR_PCPS_QUICKSYNC_ACQUISITION_CC_H
00039
00040 #include "channel_fsm.h"
00041 #include "gnss_sdr_fft.h"
00042 #include "gnss_synchro.h"
00043 #include <gnuradio/block.h>
00044 #include <gnuradio/gr_complex.h>
00045 #include <algorithm>
00046 #include <cassert>
00047 #include <fstream>
00048 #include <functional>
00049 #include <memory> // for weak_ptr
00050 #include <string>
00051 #include <utility>
00052 #include <vector>
00053
00054 /** \addtogroup Acquisition
00055  * \{ */
00056 /** \addtogroup Acq_gnuradio_blocks
00057  * \{ */
00058
00059
00060 class pcps_quicksync_acquisition_cc;
00061
00062 using pcps_quicksync_acquisition_cc_sptr = gnss_shared_ptr<pcps_quicksync_acquisition_cc>;
00063
00064 pcps_quicksync_acquisition_cc_sptr pcps_quicksync_make_acquisition_cc(
00065     uint32_t folding_factor,
00066     uint32_t sampled_ms,
00067     uint32_t max_dwells,
00068     uint32_t doppler_max,
00069     int64_t fs_in,
00070     int32_t samples_per_ms,
00071     int32_t samples_per_code,
00072     bool bit_transition_flag,
00073     bool dump,
00074     const std::string& dump_filename,
00075     bool enable_monitor_output);
00076
00077 /*!
00078  * \brief This class implements a Parallel Code Phase Search Acquisition with
00079  * the implementation of the Sparse QuickSync Algorithm.
00080  *
00081  * Check \ref Navitec2012 "Faster GPS via the Sparse Fourier Transform",
00082  * for details of its implementation and functionality.
00083  */
00084 class pcps_quicksync_acquisition_cc : public gr::block
00085 {
00086 public:
00087     /*!
00088      * \brief Default destructor.
00089      */
00090     ~pcps_quicksync_acquisition_cc();
00091
00092     /*!
00093      * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00094      * to exchange synchronization data between acquisition and tracking blocks.
00095      * \param p_gnss_synchro Satellite information shared by the processing blocks.
00096      */
00097     inline void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro)
00098     {
00099         d_gnss_synchro = p_gnss_synchro;
00100     }
00101
00102     /*!
00103      * \brief Returns the maximum peak of grid search.
00104      */
00105     inline uint32_t mag() const
00106     {
00107         return d_mag;
00108     }
00109
00110     /*!
00111      * \brief Initializes acquisition algorithm.

```

```

00112     */
00113     void init();
00114
00115     /*!
00116     * \brief Sets local code for PCPS acquisition algorithm.
00117     * \param code - Pointer to the PRN code.
00118     */
00119     void set_local_code(std::complex<float>* code);
00120
00121     /*!
00122     * \brief Starts acquisition algorithm, turning from standby mode to
00123     * active mode
00124     * \param active - bool that activates/deactivates the block.
00125     */
00126     inline void set_active(bool active)
00127     {
00128         d_active = active;
00129     }
00130
00131     /*!
00132     * \brief If set to 1, ensures that acquisition starts at the
00133     * first available sample.
00134     * \param state - int=1 forces start of acquisition
00135     */
00136     void set_state(int32_t state);
00137
00138     /*!
00139     * \brief Set acquisition channel unique ID
00140     * \param channel - receiver channel.
00141     */
00142     inline void set_channel(uint32_t channel)
00143     {
00144         d_channel = channel;
00145     }
00146
00147     /*!
00148     * \brief Set channel fsm associated to this acquisition instance
00149     */
00150     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm)
00151     {
00152         d_channel_fsm = std::move(channel_fsm);
00153     }
00154
00155     /*!
00156     * \brief Set statistics threshold of PCPS algorithm.
00157     * \param threshold - Threshold for signal detection (check \ref Navitec2012,
00158     * Algorithm 1, for a definition of this threshold).
00159     */
00160     inline void set_threshold(float threshold)
00161     {
00162         d_threshold = threshold;
00163     }
00164
00165     /*!
00166     * \brief Set maximum Doppler grid search
00167     * \param doppler_max - Maximum Doppler shift considered in the grid search [Hz].
00168     */
00169     inline void set_doppler_max(uint32_t doppler_max)
00170     {
00171         d_doppler_max = doppler_max;
00172     }
00173
00174     /*!
00175     * \brief Set Doppler steps for the grid search
00176     * \param doppler_step - Frequency bin of the search grid [Hz].
00177     */
00178     inline void set_doppler_step(uint32_t doppler_step)
00179     {
00180         d_doppler_step = doppler_step;
00181     }
00182
00183     /*!
00184     * \brief Parallel Code Phase Search Acquisition signal processing.
00185     */
00186     int general_work(int noutput_items, gr_vector_int& ninput_items,
00187         gr_vector_const_void_star& input_items,
00188         gr_vector_void_star& output_items);
00189 private:
00190     friend pcps_quicksync_acquisition_cc_sptr
00191     pcps_quicksync_make_acquisition_cc(uint32_t folding_factor,
00192         uint32_t sampled_ms, uint32_t max_dwells,
00193         uint32_t doppler_max, int64_t fs_in,
00194         int32_t samples_per_ms, int32_t samples_per_code,
00195         bool bit_transition_flag,
00196         bool dump,
00197         const std::string& dump_filename,

```

```

00199         bool enable_monitor_output);
00200
00201     pcps_quicksync_acquisition_cc(uint32_t folding_factor,
00202         uint32_t sampled_ms, uint32_t max_dwells,
00203         uint32_t doppler_max, int64_t fs_in,
00204         int32_t samples_per_ms, int32_t samples_per_code,
00205         bool bit_transition_flag,
00206         bool dump,
00207         const std::string& dump_filename,
00208         bool enable_monitor_output);
00209
00210     void calculate_magnitudes(gr_complex* fft_begin, int32_t doppler_shift,
00211         int32_t doppler_offset);
00212
00213     std::weak_ptr<ChannelFsm> d_channel_fsm;
00214
00215     std::unique_ptr<gnss_fft_complex_fwd> d_fft_if;
00216     std::unique_ptr<gnss_fft_complex_rev> d_ifft;
00217
00218     std::vector<std::vector<gr_complex>> d_grid_doppler_wipeoffs;
00219     std::vector<gr_complex> d_code;
00220     std::vector<gr_complex> d_fft_codes;
00221     std::vector<gr_complex> d_signal_folded;
00222     std::vector<gr_complex> d_code_folded;
00223     std::vector<float> d_magnitude;
00224     std::vector<float> d_corr_output_f;
00225     std::vector<float> d_magnitude_folded;
00226     std::vector<uint32_t> d_possible_delay;
00227
00228     std::string d_dump_filename;
00229     std::string d_satellite_str;
00230
00231     std::ofstream d_dump_file;
00232
00233     Gnss_Synchro* d_gnss_synchro;
00234
00235     int64_t d_fs_in;
00236     uint64_t d_sample_counter;
00237
00238     float d_noise_floor_power;
00239     float d_threshold;
00240     float d_doppler_freq;
00241     float d_mag;
00242     float d_input_power;
00243     float d_test_statistics;
00244     int32_t d_samples_per_ms;
00245     int32_t d_samples_per_code;
00246     int32_t d_state;
00247     uint32_t d_channel;
00248     uint32_t d_folding_factor; // also referred in the paper as 'p'
00249     uint32_t d_doppler_resolution;
00250     uint32_t d_doppler_max;
00251     uint32_t d_doppler_step;
00252     uint32_t d_sampled_ms;
00253     uint32_t d_max_dwells;
00254     uint32_t d_well_count;
00255     uint32_t d_fft_size;
00256     uint32_t d_num_doppler_bins;
00257     uint32_t d_code_phase;
00258
00259     bool d_bit_transition_flag;
00260     bool d_active;
00261     bool d_dump;
00262     bool d_enable_monitor_output;
00263 };
00264
00265 /** \} */
00266 /** \} */
00267 /** \} */
00268 #endif // GNSS_SDR_PCPS_QUICKSYNC_ACQUISITION_CC_H

```

## 11.73 pcps\_tong\_acquisition\_cc.h File Reference

This class implements a Parallel Code Phase Search Acquisition with Tong algorithm.

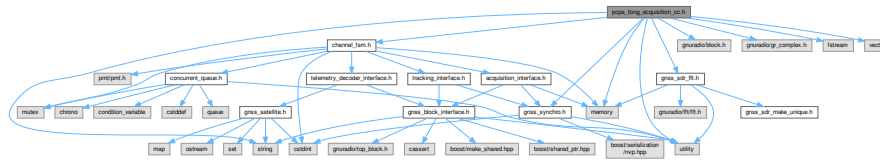
```

#include "channel_fsm.h"
#include "gnss_sdr_fft.h"
#include "gnss_synchro.h"
#include <gnuradio/block.h>
#include <gnuradio/gr_complex.h>
#include <fstream>
#include <memory>

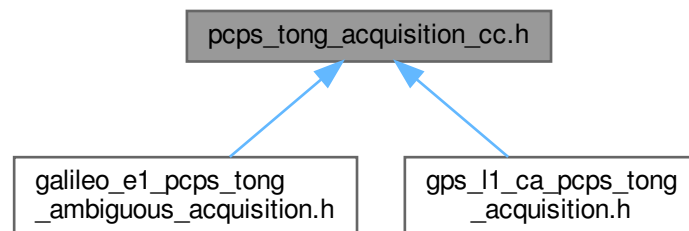
```



```
#include <string>
#include <utility>
#include <vector>
Include dependency graph for pcps_tong_acquisition_cc.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class `pcps_tong_acquisition_cc`

*This class implements a Parallel Code Phase Search Acquisition with Tong algorithm.*

## Typedefs

- using `pcps_tong_acquisition_cc_sptr = gnss_shared_ptr<pcps_tong_acquisition_cc>`

## Functions

- `pccps_tong_acquisition_cc` **`pccps_tong_make_acquisition_cc`** (`uint32_t` `sampled_ms`, `uint32_t` `doppler_max`, `int64_t` `fs_in`, `int32_t` `samples_per_ms`, `int32_t` `samples_per_code`, `uint32_t` `tong_init_val`, `uint32_t` `tong_max_val`, `uint32_t` `tong_max_dwells`, `bool` `dump`, `const std::string &dump_filename`, `bool` `enable_monitor_output`)

### 11.73.1 Detailed Description

This class implements a Parallel Code Phase Search Acquisition with Tong algorithm.

**Author**

Marc Molina, 2013. marc.molina.pena(at)gmail.com

Acquisition strategy (Kaplan book + CFAR threshold).

1. Compute the input signal power estimation.
2. Doppler serial search loop.

3. Perform the FFT-based circular convolution (parallel time search).
4. Compute the tests statistics for all the cells.
5. Accumulate the grid of tests statistics with the previous grids.
6. Record the maximum peak and the associated synchronization parameters.
7. Compare the maximum averaged test statistics with a threshold.
8. If the test statistics exceeds the threshold, increment the Tong counter.
9. Otherwise, decrement the Tong counter.
10. If the Tong counter is equal to a given maximum value, declare positive
11. acquisition. If the Tong counter is equa to zero, declare negative
12. acquisition. Otherwise, process the next block.

Kaplan book: D.Kaplan, J.Hegarty, "Understanding GPS. Principles and Applications", Artech House, 2006, pp 223-227

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [pcps\\_tong\\_acquisition\\_cc.h](#).

## 11.74 pcps\_tong\_acquisition\_cc.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file pcps_tong_acquisition_cc.h
00003  * \brief This class implements a Parallel Code Phase Search Acquisition with
00004  * Tong algorithm.
00005  * \author Marc Molina, 2013. marc.molina.pena(at)gmail.com
00006  *
00007  * Acquisition strategy (Kaplan book + CFAR threshold).
00008  * <ol>
00009  * <li> Compute the input signal power estimation.
00010  * <li> Doppler serial search loop.
00011  * <li> Perform the FFT-based circular convolution (parallel time search).
00012  * <li> Compute the tests statistics for all the cells.
00013  * <li> Accumulate the grid of tests statistics with the previous grids.
00014  * <li> Record the maximum peak and the associated synchronization parameters.
00015  * <li> Compare the maximum averaged test statistics with a threshold.
00016  * <li> If the test statistics exceeds the threshold, increment the Tong counter.
00017  * <li> Otherwise, decrement the Tong counter.
00018  * <li> If the Tong counter is equal to a given maximum value, declare positive
00019  * <li> acquisition. If the Tong counter is equa to zero, declare negative
00020  * <li> acquisition. Otherwise, process the next block.
00021  * </ol>
00022  *
00023  * Kaplan book: D.Kaplan, J.Hegarty, "Understanding GPS. Principles
00024  * and Applications", Artech House, 2006, pp 223-227
00025  *
00026  * -----
00027  *
00028  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00029  * This file is part of GNSS-SDR.
00030  *
00031  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00032  * SPDX-License-Identifier: GPL-3.0-or-later
00033  *
00034  * -----
00035  */
00036
00037 #ifndef GNSS_SDR_PCPS_TONG_ACQUISITION_CC_H
00038 #define GNSS_SDR_PCPS_TONG_ACQUISITION_CC_H
00039
00040 #include "channel_fsm.h"
00041 #include "gnss_sdr_fft.h"
00042 #include "gnss_synchro.h"
00043 #include <gnuradio/block.h>
00044 #include <gnuradio/gr_complex.h>
00045 #include <fstream>
00046 #include <memory> // for weak_ptr
00047 #include <string>
00048 #include <utility>
00049 #include <vector>
```

```

00050
00051 /** \addtogroup Acquisition
00052  * \{ */
00053 /** \addtogroup Acq_gnuradio_blocks
00054  * \{ */
00055
00056
00057 class pcps_tong_acquisition_cc;
00058
00059 using pcps_tong_acquisition_cc_sptr = gnss_shared_ptr<pcps_tong_acquisition_cc>;
00060
00061 pcps_tong_acquisition_cc_sptr pcps_tong_make_acquisition_cc(
00062     uint32_t sampled_ms,
00063     uint32_t doppler_max,
00064     int64_t fs_in,
00065     int32_t samples_per_ms,
00066     int32_t samples_per_code,
00067     uint32_t tong_init_val,
00068     uint32_t tong_max_val,
00069     uint32_t tong_max_dwells,
00070     bool dump,
00071     const std::string& dump_filename,
00072     bool enable_monitor_output);
00073
00074 /*!
00075  * \brief This class implements a Parallel Code Phase Search Acquisition with
00076  * Tong algorithm.
00077  */
00078 class pcps_tong_acquisition_cc : public gr::block
00079 {
00080 public:
00081     /*!
00082     * \brief Default destructor.
00083     */
00084     ~pcps_tong_acquisition_cc();
00085
00086     /*!
00087     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00088     * to exchange synchronization data between acquisition and tracking blocks.
00089     * \param p_gnss_synchro Satellite information shared by the processing blocks.
00090     */
00091     inline void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro)
00092     {
00093         d_gnss_synchro = p_gnss_synchro;
00094     }
00095
00096     /*!
00097     * \brief Returns the maximum peak of grid search.
00098     */
00099     inline uint32_t mag() const
00100     {
00101         return d_mag;
00102     }
00103
00104     /*!
00105     * \brief Initializes acquisition algorithm.
00106     */
00107     void init();
00108
00109     /*!
00110     * \brief Sets local code for TONG acquisition algorithm.
00111     * \param code - Pointer to the PRN code.
00112     */
00113     void set_local_code(std::complex<float>* code);
00114
00115     /*!
00116     * \brief Starts acquisition algorithm, turning from standby mode to
00117     * active mode
00118     * \param active - bool that activates/deactivates the block.
00119     */
00120     inline void set_active(bool active)
00121     {
00122         d_active = active;
00123     }
00124
00125     /*!
00126     * \brief If set to 1, ensures that acquisition starts at the
00127     * first available sample.
00128     * \param state - int=1 forces start of acquisition
00129     */
00130     void set_state(int32_t state);
00131
00132     /*!
00133     * \brief Set acquisition channel unique ID
00134     * \param channel - receiver channel.
00135     */
00136     inline void set_channel(uint32_t channel)

```

```

00137     {
00138         d_channel = channel;
00139     }
00140
00141     /*!
00142     * \brief Set channel fsm associated to this acquisition instance
00143     */
00144     inline void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm)
00145     {
00146         d_channel_fsm = std::move(channel_fsm);
00147     }
00148
00149     /*!
00150     * \brief Set statistics threshold of TONG algorithm.
00151     * \param threshold - Threshold for signal detection (check \ref Navitec2012,
00152     * Algorithm 1, for a definition of this threshold).
00153     */
00154     inline void set_threshold(float threshold)
00155     {
00156         d_threshold = threshold;
00157     }
00158
00159     /*!
00160     * \brief Set maximum Doppler grid search
00161     * \param doppler_max - Maximum Doppler shift considered in the grid search [Hz].
00162     */
00163     inline void set_doppler_max(uint32_t doppler_max)
00164     {
00165         d_doppler_max = doppler_max;
00166     }
00167
00168     /*!
00169     * \brief Set Doppler steps for the grid search
00170     * \param doppler_step - Frequency bin of the search grid [Hz].
00171     */
00172     inline void set_doppler_step(uint32_t doppler_step)
00173     {
00174         d_doppler_step = doppler_step;
00175     }
00176
00177     /*!
00178     * \brief Parallel Code Phase Search Acquisition signal processing.
00179     */
00180     int general_work(int noutput_items, gr_vector_int& ninput_items,
00181                     gr_vector_const_void_star& input_items,
00182                     gr_vector_void_star& output_items);
00183
00184 private:
00185     friend pcps_tong_acquisition_cc_sptr
00186     pcps_tong_make_acquisition_cc(uint32_t sampled_ms, uint32_t doppler_max,
00187                                   int64_t fs_in, int32_t samples_per_ms,
00188                                   int32_t samples_per_code, uint32_t tong_init_val,
00189                                   uint32_t tong_max_val, uint32_t tong_max_dwells,
00190                                   bool dump, const std::string& dump_filename, bool enable_monitor_output);
00191
00192     pcps_tong_acquisition_cc(uint32_t sampled_ms, uint32_t doppler_max,
00193                               int64_t fs_in, int32_t samples_per_ms,
00194                               int32_t samples_per_code, uint32_t tong_init_val,
00195                               uint32_t tong_max_val, uint32_t tong_max_dwells,
00196                               bool dump, const std::string& dump_filename, bool enable_monitor_output);
00197
00198     void calculate_magnitudes(gr_complex* fft_begin, int32_t doppler_shift,
00199                               int32_t doppler_offset);
00200
00201     std::weak_ptr<ChannelFsm> d_channel_fsm;
00202     std::unique_ptr<gnss_fft_complex_fwd> d_fft_if;
00203     std::unique_ptr<gnss_fft_complex_rev> d_ifft;
00204
00205     std::vector<std::vector<gr_complex>> d_grid_doppler_wipeoffs;
00206     std::vector<std::vector<float>> d_grid_data;
00207     std::vector<gr_complex> d_fft_codes;
00208     std::vector<float> d_magnitude;
00209
00210     std::string d_satellite_str;
00211     std::string d_dump_filename;
00212
00213     std::ofstream d_dump_file;
00214
00215     Gnss_Synchro* d_gnss_synchro;
00216
00217     int64_t d_fs_in;
00218     uint64_t d_sample_counter;
00219
00220     float d_threshold;
00221     float d_doppler_freq;
00222     float d_mag;
00223     float d_input_power;

```

```

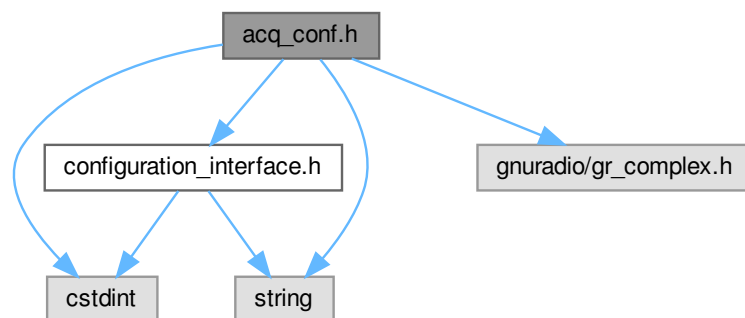
00224     float d_test_statistics;
00225     int32_t d_state;
00226     int32_t d_samples_per_ms;
00227     int32_t d_samples_per_code;
00228     uint32_t d_channel;
00229     uint32_t d_doppler_resolution;
00230     uint32_t d_doppler_max;
00231     uint32_t d_doppler_step;
00232     uint32_t d_sampled_ms;
00233     uint32_t d_dwell_count;
00234     uint32_t d_tong_init_val;
00235     uint32_t d_tong_max_val;
00236     uint32_t d_tong_max_dwells;
00237     uint32_t d_tong_count;
00238     uint32_t d_fft_size;
00239     uint32_t d_num_doppler_bins;
00240     uint32_t d_code_phase;
00241
00242     bool d_active;
00243     bool d_dump;
00244     bool d_enable_monitor_output;
00245 };
00246
00247
00248 /** \} */
00249 /** \} */
00250 #endif // GNSS_SDR_PCPS_TONG_ACQUISITION_CC_H

```

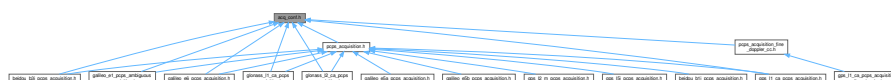
Class that contains all the configuration parameters for generic acquisition block based on the PCPS algorithm.

```
#include <string>
```

Include dependency graph for acq\_conf.h:



This graph shows which files directly or indirectly include this file:



- class Acq Conf

### 11.75.1 Detailed Description

Class that contains all the configuration parameters for generic acquisition block based on the PCPS algorithm.

Author

Carles Fernandez, 2018. cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [acq\\_conf.h](#).

---

## 11.76 acq\_conf.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file acq_conf.h
00003  * \brief Class that contains all the configuration parameters for generic
00004  * acquisition block based on the PCPS algorithm.
00005  * \author Carles Fernandez, 2018. cfernandez(at)cttc.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_ACQ_CONF_H
00019 #define GNSS_SDR_ACQ_CONF_H
00020
00021 #include "configuration_interface.h"
00022 #include <gnuradio/gr_complex.h>
00023 #include <stdint>
00024 #include <string>
00025
00026 /** \addtogroup Acquisition
00027  * \{ */
00028 /** \addtogroup acquisition_libs acquisition_libs
00029  * Library with utilities for GNSS signal acquisition
00030  * \{ */
00031
00032
00033 class Acq_Conf
00034 {
00035 public:
00036     Acq_Conf() = default;
00037
00038     void SetFromConfiguration(const ConfigurationInterface *configuration, const std::string &role,
00039                             double chip_rate, double opt_freq);
00040
00041     /* PCPS Acquisition configuration */
00042     std::string item_type{"gr_complex"};
00043     std::string dump_filename;
00044
00045     int64_t fs_in{4000000LL};
00046     int64_t resampled_fs{0LL};
00047
00048     size_t it_size{sizeof(gr_complex)};
00049
00050     float doppler_step{250.0};
00051     float samples_per_ms{0.0};
00052     float doppler_step2{125.0};
00053     float pfa{0.0};
00054     float pfa2{0.0};
00055     float samples_per_code{0.0};
00056     float resampler_ratio{1.0};
00057
00058     uint32_t sampled_ms{1U};
00059     uint32_t ms_per_code{1U};
00060     uint32_t samples_per_chip{2U};
00061     uint32_t chips_per_second{1023000U};
00062     uint32_t max_dwells{1U};
00063     uint32_t num_doppler_bins_step2{4U};
00064     uint32_t resampler_latency_samples{0U};
00065     uint32_t dump_channel{0U};
00066     int32_t doppler_max{5000};
00067     int32_t doppler_min{-5000};

```

```

00067
00068     bool bit_transition_flag{false};
00069     bool use_CFAR_algorithm_flag{true};
00070     bool dump{false};
00071     bool blocking{true};
00072     bool blocking_on_standby{false}; // enable it only for unit testing to avoid sample consume on
idle status
00073     bool make_2_steps{false};
00074     bool use_automatic_resampler{false};
00075     bool enable_monitor_output{false};
00076
00077 private:
00078     void SetDerivedParams();
00079
00080     void ConfigureAutomaticResampler(double opt_freq);
00081 };
00082
00083
00084 /** \} */
00085 /** \} */
00086 #endif // GNSS_SDR_ACQ_CONF_H

```

## 11.77 acq\_conf\_fpga.h File Reference

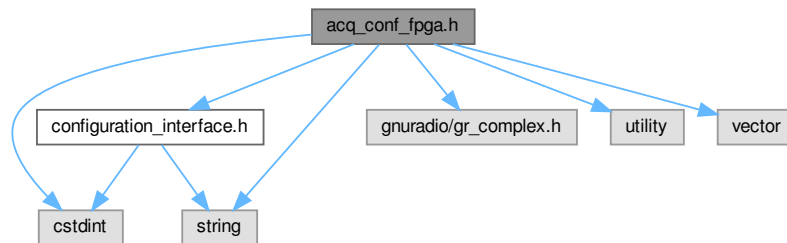
Class that contains all the configuration parameters for generic acquisition block based on the PCPS algorithm running in the FPGA.

```

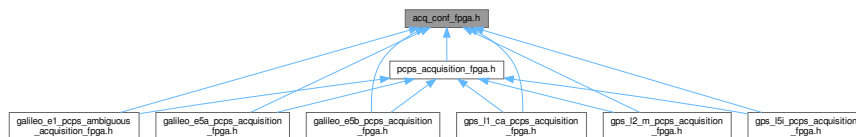
#include "configuration_interface.h"
#include <gnuradio/gr_complex.h>
#include <stdint>
#include <string>
#include <utility>
#include <vector>

```

Include dependency graph for acq\_conf\_fpga.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Acq\\_Conf\\_Fpga](#)

### 11.77.1 Detailed Description

Class that contains all the configuration parameters for generic acquisition block based on the PCPS algorithm running in the FPGA.

#### Author

Marc Majoral, 2022. mmajoral(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [acq\\_conf\\_fpga.h](#).

---

## 11.78 acq\_conf\_fpga.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file acq_conf_fpga.h
00003  * \brief Class that contains all the configuration parameters for generic
00004  * acquisition block based on the PCPS algorithm running in the FPGA.
00005  * \author Marc Majoral, 2022. mmajoral(at)cttc.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_ACQ_CONF_FPGA_H
00019 #define GNSS_SDR_ACQ_CONF_FPGA_H
00020
00021 #include "configuration_interface.h"
00022 #include <gnuradio/gr_complex.h>
00023 #include <stdint>
00024 #include <string>
00025 #include <utility> // for std::move, std::pair
00026 #include <vector> // for std::vector
00027
00028 /** \addtogroup Acquisition
00029  * \{ */
00030 /** \addtogroup acquisition_libs acquisition_libs
00031  * Library with utilities for GNSS signal acquisition
00032  * \{ */
00033
00034
00035 class Acq_Conf_Fpga
00036 {
00037 public:
00038     Acq_Conf_Fpga() = default;
00039
00040     void SetFromConfiguration(const ConfigurationInterface *configuration, const std::string &role,
00041                             uint32_t blk_exp, double code_chips_per_sec, double num_chips_per_code);
00042
00043     bool ConfigureAutomaticResampler(std::vector<std::pair<uint32_t, uint32_t>
00044                                     downsampling_filter_specs, uint32_t max_FFT_size, double opt_freq);
00045
00046     bool Is_acq_config_valid(uint32_t max_FFT_size);
00047
00048     /* PCPS Acquisition configuration */
00049     std::string device_name = "uio0";
00050     uint32_t *all_fft_codes = NULL; // pointer to memory that contains all the code ffts
00051     double code_rate_cps;
00052     double code_length_chips;
00053     int64_t fs_in{4000000LL};
00054     int64_t resampled_fs{4000000LL};
00055     float doppler_step{250.0};
00056     float doppler_step2{125.0};
00057     uint32_t num_doppler_bins_step2{4U};
00058     int32_t doppler_max{5000};
00059     uint32_t downsampling_filter_num{0U};
00060     uint32_t downsampling_factor{1U};
00061     uint32_t downsampling_filter_delay{0U};
00062     uint32_t total_block_exp{13U};
00063     uint32_t excludelimit{5U};
00064     uint32_t max_num_acqs{2U};
00065     uint32_t fft_size{1U};

```



```

00064     uint32_t code_length{16000U};
00065     bool make_2_steps{false};
00066     bool enable_zero_padding{false};
00067     bool repeat_satellite{false};
00068
00069 private:
00070     const std::string acquisition_device_name = "acquisition_S00_AXI"; // UIO device name
00071 };
00072
00073
00074 /** \} */
00075 /** \} */
00076 #endif // GNSS_SDR_ACQ_CONF_FPGA_H

```

## 11.79 fpga\_acquisition.h File Reference

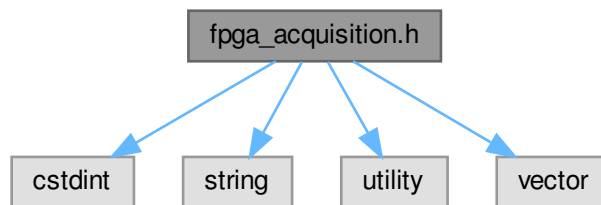
Highly optimized FPGA vector correlator class.

```

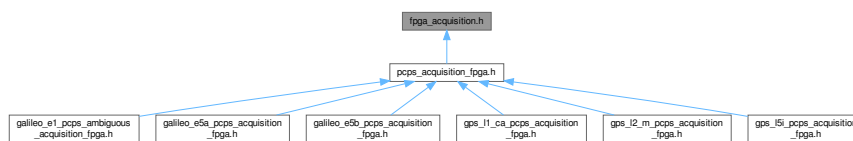
#include <cstdint>
#include <string>
#include <utility>
#include <vector>

```

Include dependency graph for fpga\_acquisition.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [Fpga\\_Acquisition](#)  
Class that implements carrier wipe-off and correlators.

### 11.79.1 Detailed Description

Highly optimized FPGA vector correlator class.

## Authors

- Marc Majoral, 2019. mmajoral(at)cttc.cat

Class that controls and executes a highly optimized acquisition HW accelerator in the FPGA  
 GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [fpga\\_acquisition.h](#).

## 11.80 fpga\_acquisition.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file fpga_acquisition.h
00003  * \brief Highly optimized FPGA vector correlator class
00004  * \authors <ul>
00005  *         <li> Marc Majoral, 2019. mmajoral(at)cttc.cat
00006  *         </ul>
00007  *
00008  * Class that controls and executes a highly optimized acquisition HW
00009  * accelerator in the FPGA
00010  *
00011  * -----
00012  *
00013  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00014  * This file is part of GNSS-SDR.
00015  *
00016  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00017  * SPDX-License-Identifier: GPL-3.0-or-later
00018  *
00019  * -----
00020  */
00021
00022 #ifndef GNSS_SDR_FPGA_ACQUISITION_H
00023 #define GNSS_SDR_FPGA_ACQUISITION_H
00024
00025 #include <cstdint>
00026 #include <string>
00027 #include <utility> // for std::move, std::pair
00028 #include <vector>   // for std::vector
00029
00030 /** \addtogroup Acquisition
00031  * \{ */
00032 /** \addtogroup acquisition_libs
00033  * \{ */
00034
00035
00036 /*!
00037 * \brief Class that implements carrier wipe-off and correlators.
00038 */
00039 class Fpga_Acquisition
00040 {
00041 public:
00042     /*!
00043     * \brief Constructor
00044     */
00045     Fpga_Acquisition(
00046         std::string device_name,
00047         uint32_t select_queue,
00048         std::vector<std::pair<uint32_t, uint32_t>> &downsampling_filter_specs,
00049         uint32_t &max_FFT_size);
00050
00051     /*!
00052     * \brief Destructor
00053     */
00054     ~Fpga_Acquisition() = default;
00055
00056     /*!
00057     * \brief Initialize acquisition parameters
00058     */
00059     // void init(uint32_t samples_per_code, uint32_t code_length, int64_t resampled_fs, uint32_t
00060     *all_fft_codes);
00061     void init(uint32_t nsamples, uint32_t doppler_max, uint32_t d_fft_size,
00062         int64_t resampled_fs, uint32_t downsampling_filter_num, uint32_t excludelimit, uint32_t
00063     *all_fft_codes);
00064
00065     /*!
00066     * \brief Select the code with the chosen PRN
00067     */
00068     bool set_local_code(uint32_t PRN);
00069
00070     /*!
00071     * \brief Configure the doppler sweep parameters in the FPGA

```

```

00070     */
00071 void set_doppler_sweep(uint32_t num_sweeps, uint32_t doppler_step, int32_t doppler_min);
00072
00073 /*!
00074  * \brief Run the acquisition process in the FPGA
00075  */
00076 void run_acquisition();
00077
00078 /*!
00079  * \brief Read the results of the acquisition process
00080  */
00081 void read_acquisition_results(
00082     uint32_t *max_index,
00083     float *firstpeak,
00084     float *secondpeak,
00085     uint64_t *initial_sample,
00086     float *power_sum,
00087     uint32_t *doppler_index,
00088     uint32_t *total_blk_exp);
00089
00090 /*!
00091  * \brief Set maximum Doppler grid search
00092  * \param doppler_max - Maximum Doppler shift considered in the grid search [Hz].
00093  */
00094 void set_doppler_max(uint32_t doppler_max)
00095 {
00096     d_doppler_max = doppler_max;
00097 }
00098
00099 /*!
00100  * \brief Set Doppler steps for the grid search
00101  * \param doppler_step - Frequency bin of the search grid [Hz].
00102  */
00103 void set_doppler_step(uint32_t doppler_step)
00104 {
00105     d_doppler_step = doppler_step;
00106 }
00107
00108 /*!
00109  * \brief Reset the FPGA PL.
00110  */
00111 void reset_acquisition();
00112
00113 /*!
00114  * \brief stop the acquisition and the FPGA modules.
00115  */
00116 void stop_acquisition();
00117
00118 /*!
00119  * \brief Set the block exponent of the FFT in the FPGA.
00120  */
00121 void set_block_exp(uint32_t total_block_exp);
00122
00123 /*!
00124  * \brief Write the PRN code in the FPGA
00125  */
00126 void write_local_code(void);
00127
00128 /*!
00129  * \brief Write the acquisition parameters into the FPGA
00130  */
00131 void configure_acquisition(void);
00132
00133 /*!
00134  * \brief Open the device driver
00135  */
00136 void open_device();
00137
00138 /*!
00139  * \brief Close the device driver
00140  */
00141 void close_device();
00142
00143 private:
00144     // FPGA IP Core version
00145     static const uint32_t FPGA_ACO_IP_VERSION_1 = 0x0001; // FPGA IP core version
00146
00147     // FPGA register addresses
00148
00149     // write-only registers
00150     static const uint32_t FREQ_BAND_DOWNSAMPLE_REG_ADDR = 0; // Select frequency band and
downsampling filter
00151     static const uint32_t FFT_LENGTH_REG_ADDR = 1; // Length of the FFT
00152     static const uint32_t CORR_NSAMPLES_REG_ADDR = 2; // Correlation length
00153     static const uint32_t DOPPLER_MIN_REG_ADDR = 3; // Doppler min
00154     static const uint32_t DOPPLER_STEP_REG_ADDR = 4; // Doppler step
00155     static const uint32_t NUM_DOPPLER_SEARCH_STEPS_REG_ADDR = 5; // Number of Doppler search steps

```

```

00156     static const uint32_t PROG_MEM_ADDR = 6;                // Access to the memory storing the
PRN code of the target satellite.
00157     static const uint32_t LOG2_FFT_LENGTH_REG_ADDR = 7;      // Log2(FFT_LENGTH)
00158     static const uint32_t ACQ_COMMAND_FLAGS_REG_ADDR = 8;    // Flags that reset, start, and stop
the acquisition process.
00159     static const uint32_t CLEAR_MEM_REG_ADDR = 9;            // Flag that resets the write
address of the PRN code memory.
00160     static const uint32_t MAX_FFT_SCALING_FACTOR_REG_ADDR = 11; // Reference FFT scaling factor
00161     static const uint32_t EXCL_LIM_REG_ADDR = 12;            // Exclude Limit value for the
second FFT peak search process
00162
00163     // read-write registers
00164     static const uint32_t TEST_REG_ADDR = 15;
00165
00166     // read-only registers
00167     static const uint32_t RESULT_VALID_REG_ADDR = 0;          // Flag that indicates a
valid result
00168     static const uint32_t SAMPLESTAMP_LSW_REG_ADDR = 1;       // Sample stamp LSW
00169     static const uint32_t SAMPLESTAMP_MSW_REG_ADDR = 2;       // Sample stamp MSW
00170     static const uint32_t MAG_SQ_FIRST_PEAK_REG_ADDR = 3;      // magnitude squared of the
first peak
00171     static const uint32_t MAG_SQ_SECOND_PEAK_REG_ADDR = 4;    // magnitude squared of the
second peak
00172     static const uint32_t ACQ_DELAY_SAMPLES_REG_ADDR = 5;     // acquisition delay in
samples
00173     static const uint32_t DOPPLER_INDEX_REG_ADDR = 7;         // Doppler index
00174     static const uint32_t FFT_SCALING_FACTOR_REG_ADDR = 8;    // Scaling factor applied by
the FFT
00175     static const uint32_t MAX_FFT_SIZE_REG_ADDR = 9;          // Maximum FFT size
supported by the FPGA
00176     static const uint32_t DOWNSAMPLING_FILTER_DEC_FACTORS_REG_ADDR = 10; // Available decimation
factors
00177     static const uint32_t DOWNSAMPLING_FILTER_LATENCIES_REG_ADDR = 11; // Available downsampling
filter latencies
00178     static const uint32_t FPGA_IP_CORE_VERSION_REG_ADDR = 14; // FPGA acquisition IP core
version
00179
00180     // FPGA register parameters
00181     static const uint32_t FPGA_PAGE_SIZE = 0x1000;           // default page size for the
multicorrelator memory map
00182     static const uint32_t LAUNCH_ACQUISITION = 1;            // command to launch the acquisition
process
00183     static const uint32_t RESET_ACQUISITION = 2;             // command to reset the acquisition
and the FPGA Modules
00184     static const uint32_t STOP_ACQUISITION = 4;              // command to stop the acquisition
and the FPGA modules
00185     static const uint32_t TEST_REG_SANITY_CHECK = 0x55AA;    // value to check the presence of
the test register (to detect the hw)
00186     static const uint32_t LOCAL_CODE_CLEAR_MEM = 0x10000000; // command to clear the internal
memory of the multicorrelator
00187     static const uint32_t MEM_LOCAL_CODE_WR_ENABLE = 0x0C000000; // command to enable the ENA and WR
pins of the internal memory of the multicorrelator
00188     static const uint32_t POW_2_2 = 4;                       // 2^2 (used for the conversion of
floating point numbers to integers)
00189     static const uint32_t POW_2_31 = 2147483648;             // 2^31 (used for the conversion of
floating point numbers to integers)
00190     static const uint32_t MAX_FILTERS_AVAILABLE = 2;         // maximum number of downsampling
filters available in the FPGA by default
00191     static const uint32_t DEFAULT_MAX_FFT_SIZE = 32768;      // default maximum FFT size
supported by the FPGA
00192     static const uint32_t ACQ_BUFF_0 = 0;                    // FPGA Acquisition IP buffer
containing L1/E1 frequency band samples by default.
00193     static const uint32_t ACQ_BUFF_1 = 0;                    // FPGA Acquisition IP buffer
containing L2 or L5/E5a frequency band samples by default.
00194
00195     // bit manipulation
00196     static const uint32_t RSHIFT_4_BITS = 0x4;
00197     static const uint32_t RSHIFT_8_BITS = 0x8;
00198     static const uint32_t BIT_MASK_4 = 0xF;
00199     static const uint32_t BIT_MASK_8 = 0xFF;
00200
00201     // Downsampling default constants
00202     const uint32_t DEFAULT_DOWNSAMPLING_FILTER_DELAY = 40;   // default downsampling filter delay (for
FPGA Acquisition IP core versions earlier than FPGA_ACQ_IP_VERSION_1)
00203     const uint32_t DEFAULT_DOWNSAMPLING_FACTOR = 4;          // default downsampling factor (for FPGA
Acquisition IP core versions earlier than FPGA_ACQ_IP_VERSION_1)
00204
00205     // private methods
00206     void fpga_acquisition_test_register(void);
00207     void read_ipcore_info(std::vector<std::pair<uint32_t, uint32_t>> &downsampling_filter_specs,
uint32_t &max_fft_size);
00208
00209     std::vector<std::pair<uint32_t, uint32_t>> d_downsampling_filter_specs;
00210     std::string d_device_name;                                // HW device name
00211     int64_t d_resampled_fs;                                    // sampling frequency
00212     volatile uint32_t *d_map_base;                             // driver memory map
00213     uint32_t *d_all_fft_codes;                                // memory that contains all the code ffts

```

```

00214     int32_t d_fd;                                // driver descriptor
00215     uint32_t d_fft_size;                          // number of samples including padding
00216     uint32_t d_excludelimit;
00217     uint32_t d_nsamples;                          // number of samples not including padding
00218     uint32_t d_filter_num;                        // Selected downsampling filter
00219     uint32_t d_downsampling_factor;               // downsampling_factor
00220     uint32_t d_downsampling_filter_delay;         // Impulse response delay of the downsampling filter
00221     uint32_t d_select_queue;                      // queue selection
00222     uint32_t d_doppler_max;                       // max doppler
00223     uint32_t d_doppler_step;                     // doppler step
00224     uint32_t d_PRN;                              // PRN
00225     uint32_t d_IP_core_version;                   // FPGA acquisition IP core version
00226 };
00227
00228
00229 /** \} */
00230 /** \} */
00231 #endif // GNSS_SDR_FPGA_ACQUISITION_H

```

## 11.81 channel.h File Reference

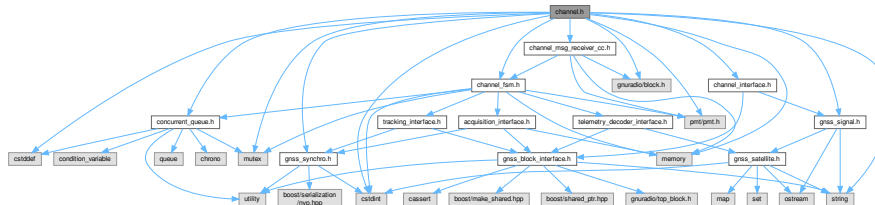
Interface of a GNSS channel.

```

#include "channel_fsm.h"
#include "channel_interface.h"
#include "channel_msg_receiver_cc.h"
#include "concurrent_queue.h"
#include "gnss_signal.h"
#include "gnss_synchro.h"
#include <gnuradio/block.h>
#include <pmt/pmt.h>
#include <stddef>
#include <stdint>
#include <memory>
#include <mutex>
#include <string>

```

Include dependency graph for channel.h:



### Classes

- class [Channel](#)

*This class represents a GNSS channel. It wraps an [AcquisitionInterface](#), a [TrackingInterface](#) and a [TelemetryDecoderInterface](#), and handles their interaction through a Finite State Machine.*

### 11.81.1 Detailed Description

Interface of a GNSS channel.

Author

Carlos Aviles, 2010. carlos.avilesr(at)gmail.com Luis Esteve, 2011. luis(at)epsilon-formacion.com

It holds blocks for acquisition, tracking, navigation data extraction and pseudorange calculation. GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR. Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later Definition in file [channel.h](#).

## 11.82 channel.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file channel.h
00003  * \brief Interface of a GNSS channel.
00004  * \author Carlos Aviles, 2010. carlos.avilesr(at)gmail.com
00005  *         Luis Esteve, 2011. luis(at)epsilon-formacion.com
00006  *
00007  * It holds blocks for acquisition, tracking,
00008  * navigation data extraction and pseudorange calculation.
00009  *
00010  * -----
00011  *
00012  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00013  * This file is part of GNSS-SDR.
00014  *
00015  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00016  * SPDX-License-Identifier: GPL-3.0-or-later
00017  *
00018  * -----
00019  */
00020
00021 #ifndef GNSS_SDR_CHANNEL_H
00022 #define GNSS_SDR_CHANNEL_H
00023
00024 #include "channel_fsm.h"
00025 #include "channel_interface.h"
00026 #include "channel_msg_receiver_cc.h"
00027 #include "concurrent_queue.h"
00028 #include "gnss_signal.h"
00029 #include "gnss_synchro.h"
00030 #include <gnuradio/block.h>
00031 #include <pmt/pmt.h>
00032 #include <cstdlib>
00033 #include <stdint>
00034 #include <memory>
00035 #include <mutex>
00036 #include <string>
00037
00038 /** \addtogroup Channel
00039  * Classes containing a GNSS channel.
00040  * \{ */
00041 /** \addtogroup Channel_adapters channel_adapters
00042  * Classes that wrap an AcquisitionInterface,
00043  * a TrackingInterface and a TelemetryDecoderInterface, and handles
00044  * their interaction.
00045  * \{ */
00046
00047
00048 class ConfigurationInterface;
00049 class AcquisitionInterface;
00050 class TrackingInterface;
00051 class TelemetryDecoderInterface;
00052
00053
00054 /*!
00055 * \brief This class represents a GNSS channel. It wraps an AcquisitionInterface,
00056 * a TrackingInterface and a TelemetryDecoderInterface, and handles
00057 * their interaction through a Finite State Machine
00058 *
00059 */
00060 class Channel : public ChannelInterface
00061 {
00062 public:
00063     /** Constructor
00064      * Channel(const ConfigurationInterface* configuration,
00065      *          uint32_t channel,
00066      *          std::shared_ptr<AcquisitionInterface> acq,
00067      *          std::shared_ptr<TrackingInterface> trk,
00068      *          std::shared_ptr<TelemetryDecoderInterface> nav,
00069      *          const std::string& role,
00070      *          const std::string& signal_str,
00071      *          Concurrent_Queue<pmt::pmt_t>* queue);
00072
00073      * ~Channel() = default; //!< Destructor
00074
00075      * void connect(gr::top_block_sptr top_block) override; //!< Connects the tracking block to the
00076      * top_block and to the telemetry
00077      * void disconnect(gr::top_block_sptr top_block) override;
00078      * gr::basic_block_sptr get_left_block() override;
00079      * gr::basic_block_sptr get_left_block_trk() override; //!< Gets the GNU Radio tracking block input
00080      * pointer
00081      * gr::basic_block_sptr get_right_block_trk() override; //!< Gets the GNU Radio tracking block
00082      * output pointer
00083      * gr::basic_block_sptr get_left_block_acq() override; //!< Gets the GNU Radio acquisition block

```

```

    input pointer
00081     gr::basic_block_sptr get_right_block_acq() override; //!< Gets the GNU Radio acquisition block
    output pointer
00082     gr::basic_block_sptr get_right_block() override;      //!< Gets the GNU Radio channel block output
    pointer
00083
00084     inline std::string role() override { return role_; }
00085     inline std::string implementation() override { return std::string("Channel"); } //!< Returns
    "Channel"
00086     inline size_t item_size() override { return 2 * sizeof(float); }
00087     Gnss_Signal get_signal() override;
00088     void start_acquisition() override;                    //!< Start the State Machine
00089     void stop_channel() override;                          //!< Stop the State Machine
00090     void set_signal(const Gnss_Signal& gnss_signal_) override; //!< Sets the channel GNSS signal
00091
00092     void assist_acquisition_doppler(double Carrier_Doppler_hz) override;
00093
00094     inline std::shared_ptr<AcquisitionInterface> acquisition() const { return acq_; }
00095     inline std::shared_ptr<TrackingInterface> tracking() const { return trk_; }
00096     inline std::shared_ptr<TelemetryDecoderInterface> telemetry() const { return nav_; }
00097
00098 private:
00099     bool glonass_dll_pll_c_aid_tracking_check() const;
00100     std::shared_ptr<ChannelFsm> channel_fsm_;
00101     std::shared_ptr<AcquisitionInterface> acq_;
00102     std::shared_ptr<TrackingInterface> trk_;
00103     std::shared_ptr<TelemetryDecoderInterface> nav_;
00104     channel_msg_receiver_cc_sptr channel_msg_rx_;
00105     Gnss_Synchro gnss_synchro_{};
00106     Gnss_Signal gnss_signal_;
00107     std::string role_;
00108     std::mutex mx_;
00109     uint32_t channel_;
00110     int glonass_extend_correlation_ms_;
00111     bool connected_;
00112     bool repeat_;
00113     bool flag_enable_fpga_;
00114 };
00115
00116
00117 /** @} */
00118 /** @} */
00119 #endif // GNSS_SDR_CHANNEL_H

```

## 11.83 channel\_fsm.h File Reference

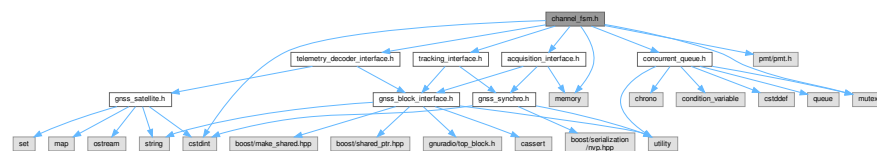
Interface of the State Machine for channel.

```

#include "acquisition_interface.h"
#include "concurrent_queue.h"
#include "telemetry_decoder_interface.h"
#include "tracking_interface.h"
#include <pmt/pmt.h>
#include <stdint>
#include <memory>
#include <mutex>

```

Include dependency graph for channel\_fsm.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [ChannelFsm](#)

*This class implements a State Machine for channel.*

### 11.83.1 Detailed Description

Interface of the State Machine for channel.

#### Authors

Javier Arribas, 2019. [javiarribas@gmail.com](mailto:javiarribas@gmail.com) Antonio Ramos, 2017. [antonio.ramos\(at\)cttc.es](mailto:antonio.ramos(at)cttc.es) Luis Esteve, 2011. [luis\(at\)epsilon-formacion.com](mailto:luis(at)epsilon-formacion.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [channel\\_fsm.h](#).

---

## 11.84 channel\_fsm.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file channel_fsm.h
00003  * \brief Interface of the State Machine for channel
00004  * \authors Javier Arribas, 2019. javiarribas@gmail.com
00005  *           Antonio Ramos, 2017. antonio.ramos(at)cttc.es
00006  *           Luis Esteve, 2011. luis(at)epsilon-formacion.com
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019 #ifndef GNSS_SDR_CHANNEL_FSM_H
00020 #define GNSS_SDR_CHANNEL_FSM_H
00021
00022 #include "acquisition_interface.h"
00023 #include "concurrent_queue.h"
00024 #include "telemetry_decoder_interface.h"
00025 #include "tracking_interface.h"
00026 #include <pmt/pmt.h>
00027 #include <cstdint>
00028 #include <memory>
00029 #include <mutex>
00030
00031 /** \addtogroup Channel
00032  * \{ */
00033 /** \addtogroup Channel_libs channel_libs
00034  * Library with utilities for a GNSS Channel.
00035  * \{ */
00036
00037
00038 /*!
00039  * \brief This class implements a State Machine for channel
00040  */
00041 class ChannelFsm
00042 {
00043 public:
00044     ChannelFsm();
00045     virtual ~ChannelFsm() = default;
00046     explicit ChannelFsm(std::shared_ptr<AcquisitionInterface> acquisition);
00047
00048     void set_acquisition(std::shared_ptr<AcquisitionInterface> acquisition);
00049     void set_tracking(std::shared_ptr<TrackingInterface> tracking);
00050     void set_telemetry(std::shared_ptr<TelemetryDecoderInterface> telemetry);
00051     void set_queue(Concurrent_Queue<pmt::pmt_t>* queue);
00052     void set_channel(uint32_t channel);
00053     void start_acquisition();
00054
00055     // FSM EVENTS
00056     bool Event_start_acquisition();
00057     bool Event_start_acquisition_fpga();
00058     bool Event_stop_channel();
```



```

00059     bool Event_failed_tracking_standby();
00060     virtual bool Event_valid_acquisition();
00061     virtual bool Event_failed_acquisition_repeat();
00062     virtual bool Event_failed_acquisition_no_repeat();
00063
00064 private:
00065     void start_tracking();
00066     void stop_acquisition();
00067     void stop_tracking();
00068     void request_satellite();
00069     void notify_stop_tracking();
00070
00071     std::shared_ptr<AcquisitionInterface> acq_;
00072     std::shared_ptr<TrackingInterface> trk_;
00073     std::shared_ptr<TelemetryDecoderInterface> nav_;
00074
00075     std::mutex mx_;
00076
00077     Concurrent_Queue<pmt::pmt_t> queue_;
00078
00079     uint32_t channel_;
00080     uint32_t state_;
00081 };
00082
00083
00084 /** \} */
00085 /** \} */
00086 #endif // GNSS_SDR_CHANNEL_FSM_H

```

## 11.85 channel\_msg\_receiver\_cc.h File Reference

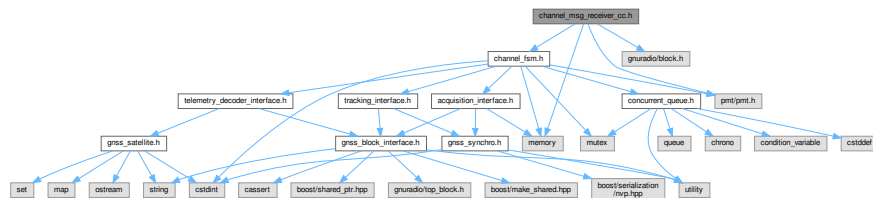
GNU Radio block that receives asynchronous channel messages from acquisition and tracking blocks.

```

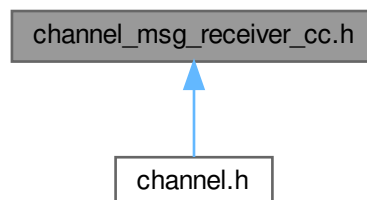
#include "channel_fsm.h"
#include <gnuradio/block.h>
#include <pmt/pmt.h>
#include <memory>

```

Include dependency graph for channel\_msg\_receiver\_cc.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [channel\\_msg\\_receiver\\_cc](#)

*GNU Radio block that receives asynchronous channel messages from acquisition and tracking blocks.*

## Typedefs

- using [channel\\_msg\\_receiver\\_cc\\_sptr](#) = `gnss_shared_ptr<channel\_msg\_receiver\_cc>`

## Functions

- `channel_msg_receiver_cc_sptr channel_msg_receiver_make_cc (std::shared_ptr< ChannelFsm > channel_fsm, bool repeat)`

### 11.85.1 Detailed Description

GNU Radio block that receives asynchronous channel messages from acquisition and tracking blocks.

#### Author

Javier Arribas, 2016. [jarribas\(at\)cttc.es](mailto:jarribas(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [channel\\_msg\\_receiver\\_cc.h](#).

---

## 11.86 channel\_msg\_receiver\_cc.h

[Go to the documentation of this file.](#)

```
00001  /*!
00002  * \file channel_msg_receiver_cc.h
00003  * \brief GNU Radio block that receives asynchronous channel messages from acquisition and tracking
00004  * \author Javier Arribas, 2016. jarribas(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_CHANNEL_MSG_RECEIVER_CC_H
00018 #define GNSS_SDR_CHANNEL_MSG_RECEIVER_CC_H
00019
00020 #include "channel_fsm.h"
00021 #include <gnuradio/block.h>
00022 #include <pmt/pmt.h>
00023 #include <memory>
00024
00025 /** \addtogroup Channel
00026  * \{ */
00027 /** \addtogroup Channel_libs
00028  * \{ */
00029
00030
00031 class channel_msg_receiver_cc;
00032
00033 using channel_msg_receiver_cc_sptr = gnss_shared_ptr<channel_msg_receiver_cc>;
00034
00035 channel_msg_receiver_cc_sptr channel_msg_receiver_make_cc(std::shared_ptr<ChannelFsm> channel_fsm,
00036 bool repeat);
00037
00038 /*!
00039 * \brief GNU Radio block that receives asynchronous channel messages from acquisition and tracking
00040 * \{
00041 * \{
00042 public:
00043     ~channel_msg_receiver_cc() = default; //!< Default destructor
```

```

00044
00045 private:
00046     friend channel_msg_receiver_cc_sptr channel_msg_receiver_make_cc(std::shared_ptr<ChannelFsm>
        channel_fsm, bool repeat);
00047     channel_msg_receiver_cc(std::shared_ptr<ChannelFsm> channel_fsm, bool repeat);
00048     void msg_handler_channel_events(const pmt::pmt_t& msg);
00049     std::shared_ptr<ChannelFsm> d_channel_fsm;
00050     bool d_repeat; // todo: change FSM to include repeat value
00051 };
00052
00053
00054 /** \} */
00055 /** \} */
00056 #endif // GNSS_SDR_CHANNEL_MSG_RECEIVER_CC_H

```

## 11.87 array\_signal\_conditioner.h File Reference

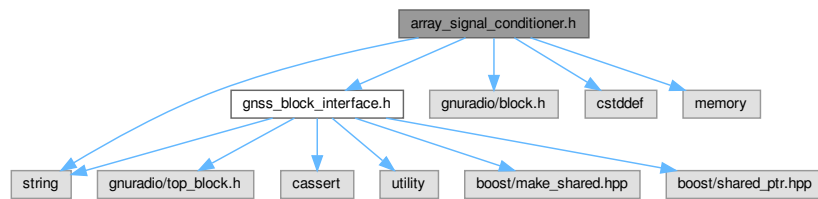
It wraps blocks to change data type, filter and resample input data, adapted to array receiver.

```

#include "gnss_block_interface.h"
#include <gnuradio/block.h>
#include <cstdlib>
#include <memory>
#include <string>

```

Include dependency graph for array\_signal\_conditioner.h:



### Classes

- class [ArraySignalConditioner](#)

*This class wraps blocks to change data\_type\_adapter, input\_filter and resampler to be applied to the input flow of sampled signal.*

### 11.87.1 Detailed Description

It wraps blocks to change data type, filter and resample input data, adapted to array receiver.

#### Author

Javier Arribas jarribas (at) cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [array\\_signal\\_conditioner.h](#).

---

## 11.88 array\_signal\_conditioner.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file array_signal_conditioner.h
00003  * \brief It wraps blocks to change data type, filter and resample input data, adapted to array
        receiver
00004  * \author Javier Arribas jarribas (at) cttc.es
00005  *
00006  *

```

```

00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_ARRAY_SIGNAL_CONDITIONER_H
00019 #define GNSS_SDR_ARRAY_SIGNAL_CONDITIONER_H
00020
00021
00022 #include "gnss_block_interface.h"
00023 #include <gnuradio/block.h>
00024 #include <cstdint>
00025 #include <memory>
00026 #include <string>
00027
00028 /** \addtogroup Signal_Conditioner
00029  * \{ */
00030 /** \addtogroup Signal_Conditioner_adapters
00031  * \{ */
00032
00033
00034 class ConfigurationInterface;
00035
00036
00037 /*!
00038  * \brief This class wraps blocks to change data_type_adapter, input_filter and resampler
00039  * to be applied to the input flow of sampled signal.
00040  */
00041 class ArraySignalConditioner : public GNSSBlockInterface
00042 {
00043 public:
00044     /** Constructor
00045      * ArraySignalConditioner(std::shared_ptr<GNSSBlockInterface> data_type_adapt,
00046      * std::shared_ptr<GNSSBlockInterface> in_filt,
00047      * std::shared_ptr<GNSSBlockInterface> res,
00048      * std::string role);
00049
00050      * \{ Destructor
00051      * ~ArraySignalConditioner() = default;
00052
00053      * void connect(gr::top_block_sptr top_block) override;
00054      * void disconnect(gr::top_block_sptr top_block) override;
00055      * gr::basic_block_sptr get_left_block() override;
00056      * gr::basic_block_sptr get_right_block() override;
00057
00058      * inline std::string role() override { return role_; }
00059      * \{ Returns "Array_Signal_Conditioner"
00060      * inline std::string implementation() override { return "Array_Signal_Conditioner"; }
00061      * inline size_t item_size() override { return data_type_adapter->item_size(); }
00062
00063      * inline std::shared_ptr<GNSSBlockInterface> data_type_adapter() { return data_type_adapter_; }
00064      * inline std::shared_ptr<GNSSBlockInterface> input_filter() { return in_filt_; }
00065      * inline std::shared_ptr<GNSSBlockInterface> resampler() { return res_; }
00066
00067 private:
00068     std::shared_ptr<GNSSBlockInterface> data_type_adapter_;
00069     std::shared_ptr<GNSSBlockInterface> in_filt_;
00070     std::shared_ptr<GNSSBlockInterface> res_;
00071     std::string role_;
00072     bool connected_;
00073 };
00074
00075
00076 /** \} */
00077 /** \} */
00078 #endif // GNSS_SDR_SIGNAL_CONDITIONER_H

```

## 11.89 signal\_conditioner.h File Reference

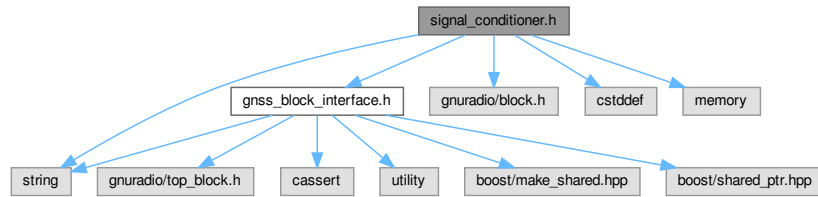
It wraps blocks to change data type, filter and resample input data.

```

#include "gnss_block_interface.h"
#include <gnuradio/block.h>
#include <cstdint>
#include <memory>
#include <string>

```

Include dependency graph for signal\_conditioner.h:



## Classes

- class [SignalConditioner](#)

*This class wraps blocks to change data\_type\_adapter, input\_filter and resampler to be applied to the input flow of sampled signal.*

### 11.89.1 Detailed Description

It wraps blocks to change data type, filter and resample input data.

#### Author

Luis Esteve, 2012. [luis\(at\)epsilon-formacion.com](mailto:luis(at)epsilon-formacion.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [signal\\_conditioner.h](#).

---

## 11.90 signal\_conditioner.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file signal_conditioner.h
00003  * \brief It wraps blocks to change data type, filter and resample input data.
00004  * \author Luis Esteve, 2012. luis\(at\)epsilon-formacion.com
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_SIGNAL_CONDITIONER_H
00019 #define GNSS_SDR_SIGNAL_CONDITIONER_H
00020
00021 #include "gnss_block_interface.h"
00022 #include <gnuradio/block.h>
00023 #include <cstdint>
00024 #include <memory>
00025 #include <string>
00026
00027 /** \addtogroup SignalConditioner Signal Conditioner
00028  * Signal Conditioner wrapper block
00029  * \{ */
00030 /** \addtogroup SignalConditioner_adapters conditioner_adapters
00031  * Wrap a Signal Conditioner with a GNSSBlockInterface
00032  * \{ */
00033
00034
00035 /*!
00036  * \brief This class wraps blocks to change data_type_adapter, input_filter and resampler
00037  * to be applied to the input flow of sampled signal.
  
```

```

00038  */
00039  class SignalConditioner : public GNSSBlockInterface
00040  {
00041  public:
00042      ///! Constructor
00043      SignalConditioner(std::shared_ptr<GNSSBlockInterface> data_type_adapt,
00044                      std::shared_ptr<GNSSBlockInterface> in_filt,
00045                      std::shared_ptr<GNSSBlockInterface> res,
00046                      std::string role);
00047
00048      ///! Destructor
00049      ~SignalConditioner() = default;
00050
00051      void connect(gr::top_block_sptr top_block) override;
00052      void disconnect(gr::top_block_sptr top_block) override;
00053      gr::basic_block_sptr get_left_block() override;
00054      gr::basic_block_sptr get_right_block() override;
00055
00056      inline std::string role() override { return role_; }
00057
00058      inline std::string implementation() override { return "Signal_Conditioner"; } //!< Returns
"Signal_Conditioner"
00059
00060      inline size_t item_size() override { return data_type_adapt->item_size(); }
00061
00062      inline std::shared_ptr<GNSSBlockInterface> data_type_adapter() { return data_type_adapt_; }
00063      inline std::shared_ptr<GNSSBlockInterface> input_filter() { return in_filt_; }
00064      inline std::shared_ptr<GNSSBlockInterface> resampler() { return res_; }
00065
00066  private:
00067      std::shared_ptr<GNSSBlockInterface> data_type_adapt_;
00068      std::shared_ptr<GNSSBlockInterface> in_filt_;
00069      std::shared_ptr<GNSSBlockInterface> res_;
00070      std::string role_;
00071      bool connected_;
00072  };
00073
00074
00075  /** \} */
00076  /** \} */
00077  #endif // GNSS_SDR_SIGNAL_CONDITIONER_H

```

## 11.91 byte\_to\_short.h File Reference

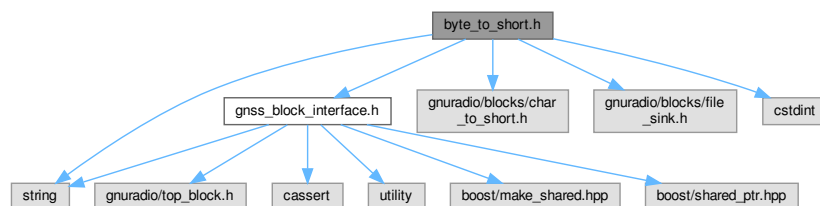
Adapts an 8-bits sample stream (IF) to a short int stream (IF)

```

#include "gnss_block_interface.h"
#include <gnuradio/blocks/char_to_short.h>
#include <gnuradio/blocks/file_sink.h>
#include <cstdint>
#include <string>

```

Include dependency graph for byte\_to\_short.h:



### Classes

- class [ByteToShort](#)

*Adapts an 8-bits sample stream (IF) to a short int stream (IF)*

### 11.91.1 Detailed Description

Adapts an 8-bits sample stream (IF) to a short int stream (IF)

Author

Carles Fernandez Prades, cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [byte\\_to\\_short.h](#).

---

## 11.92 byte\_to\_short.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file byte_to_short.h
00003  * \brief Adapts an 8-bits sample stream (IF) to a short int stream (IF)
00004  * \author Carles Fernandez Prades, cfernandez(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_BYTE_TO_SHORT_H
00018 #define GNSS_SDR_BYTE_TO_SHORT_H
00019
00020 #include "gnss_block_interface.h"
00021 #include <gnuradio/blocks/char_to_short.h>
00022 #include <gnuradio/blocks/file_sink.h>
00023 #include <stdint>
00024 #include <string>
00025
00026 /** \addtogroup Data_Type Data Type Adapters
00027  * Classes for data type conversion
00028  * \{ */
00029 /** \addtogroup Data_type_adapters data_type_adapters
00030  * Wrap GNU Radio data type adapter blocks with a GNSSBlockInterface
00031  * \{ */
00032
00033
00034 class ConfigurationInterface;
00035
00036 /*!
00037 * \brief Adapts an 8-bits sample stream (IF) to a short int stream (IF)
00038 *
00039 */
00040 class ByteToShort : public GNSSBlockInterface
00041 {
00042 public:
00043     ByteToShort(const ConfigurationInterface* configuration,
00044                 std::string role, unsigned int in_streams,
00045                 unsigned int out_streams);
00046
00047     ~ByteToShort() = default;
00048
00049     inline std::string role() override
00050     {
00051         return role_;
00052     }
00053
00054     //! Returns "Byte_To_Short"
00055     inline std::string implementation() override
00056     {
00057         return "Byte_To_Short";
00058     }
00059
00060     inline size_t item_size() override
00061     {
00062         return sizeof(int8_t);
00063     }
00064
00065     void connect(gr::top_block_sptr top_block) override;
00066     void disconnect(gr::top_block_sptr top_block) override;
00067     gr::basic_block_sptr get_left_block() override;

```

```

00068     gr::basic_block_sptr get_right_block() override;
00069
00070 private:
00071     gr::blocks::char_to_short::sptr gr_char_to_short_;
00072     gr::blocks::file_sink::sptr file_sink_;
00073     std::string dump_filename_;
00074     std::string input_item_type_;
00075     std::string output_item_type_;
00076     std::string role_;
00077     unsigned int in_streams_;
00078     unsigned int out_streams_;
00079     bool dump_;
00080 };
00081
00082
00083 /** \} */
00084 /** \} */
00085 #endif // GNSS_SDR_BYTE_TO_SHORT_H

```

## 11.93 cshort\_to\_grcomplex.h File Reference

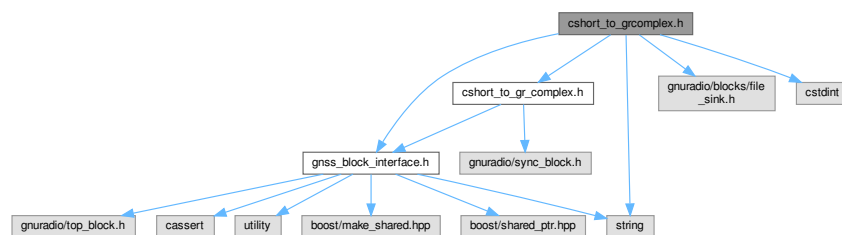
Adapts an 16-bits complex sample stream to a float complex stream.

```

#include "cshort_to_gr_complex.h"
#include "gnss_block_interface.h"
#include <gnuradio/blocks/file_sink.h>
#include <stdint>
#include <string>

```

Include dependency graph for cshort\_to\_grcomplex.h:



### Classes

- class [CshortToGrComplex](#)

*Adapts an 16-bits complex sample stream to a float complex stream.*

### 11.93.1 Detailed Description

Adapts an 16-bits complex sample stream to a float complex stream.

#### Author

Carles Fernandez Prades, 2014 cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [cshort\\_to\\_grcomplex.h](#).

---

## 11.94 cshort\_to\_grcomplex.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file cshort_to_grcomplex.h
00003  * \brief Adapts an 16-bits complex sample stream to a float complex stream
00004  * \author Carles Fernandez Prades, 2014 cfernandez(at)cttc.es

```



```

00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_CSHORT_TO_GRCOMPLEX_H
00018 #define GNSS_SDR_CSHORT_TO_GRCOMPLEX_H
00019
00020 #include "cshort_to_gr_complex.h"
00021 #include "gnss_block_interface.h"
00022 #include <gnuradio/blocks/file_sink.h>
00023 #include <stdint>
00024 #include <string>
00025
00026 /** \addtogroup Data_Type Data Type Adapters
00027  * Classes for data type conversion
00028  * \{ */
00029 /** \addtogroup Data_type_adapters data_type_adapters
00030  * Wrap GNU Radio data type adapter blocks with a GNSSBlockInterface
00031  * \{ */
00032
00033
00034 class ConfigurationInterface;
00035
00036 /*!
00037  * \brief Adapts an 16-bits complex sample stream to a float complex stream
00038  *
00039  */
00040 class CshortToGrComplex : public GNSSBlockInterface
00041 {
00042 public:
00043     CshortToGrComplex(const ConfigurationInterface* configuration,
00044         std::string role, unsigned int in_streams,
00045         unsigned int out_streams);
00046
00047     ~CshortToGrComplex() = default;
00048
00049     inline std::string role() override
00050     {
00051         return role_;
00052     }
00053
00054     /*! Returns "Cshort_To_Gr_Complex"
00055     inline std::string implementation() override
00056     {
00057         return "Cshort_To_Gr_Complex";
00058     }
00059
00060     inline size_t item_size() override
00061     {
00062         return 2 * sizeof(float);
00063     }
00064
00065     void connect(gr::top_block_sptr top_block) override;
00066     void disconnect(gr::top_block_sptr top_block) override;
00067     gr::basic_block_sptr get_left_block() override;
00068     gr::basic_block_sptr get_right_block() override;
00069
00070 private:
00071     cshort_to_gr_complex_sptr cshort_to_gr_complex_;
00072     gr::blocks::file_sink::sptr file_sink_;
00073     std::string dump_filename_;
00074     std::string role_;
00075     unsigned int in_streams_;
00076     unsigned int out_streams_;
00077     bool dump_;
00078 };
00079
00080
00081 /** \} */
00082 /** \} */
00083 #endif // GNSS_SDR_CSHORT_TO_GRCOMPLEX_H

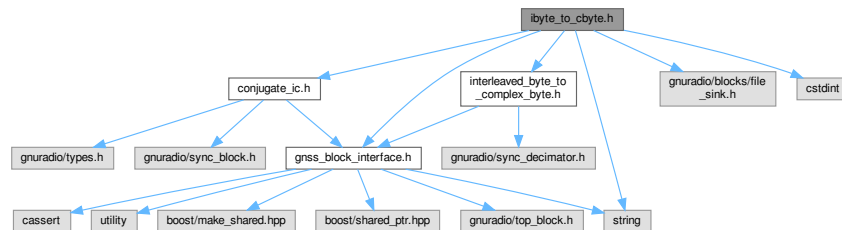
```

## 11.95 ibyte\_to\_cbyte.h File Reference

Adapts an I/Q interleaved byte (unsigned char) sample stream into a `std::complex<unsigned char>` stream.

```
#include "conjugate_ic.h"
#include "gnss_block_interface.h"
#include "interleaved_byte_to_complex_byte.h"
#include <gnuradio/blocks/file_sink.h>
#include <stdint>
#include <string>
```

Include dependency graph for `ibyte_to_cbyte.h`:



## Classes

- class [IByteToCByte](#)

## 11.95.1 Detailed Description

Adapts an I/Q interleaved byte (unsigned char) sample stream into a `std::complex<unsigned char>` stream.

### Author

Carles Fernandez Prades, [cfernandez\(at\)cttc.es](mailto:cfernandez(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [ibyte\\_to\\_cbyte.h](#).

---

## 11.96 ibyte\_to\_cbyte.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file ibyte_to_cbyte.h
00003  * \brief \brief Adapts an I/Q interleaved byte (unsigned char) sample stream
00004  * into a std::complex<unsigned char> stream
00005  * \author Carles Fernandez Prades, cfernandez(at)cttc.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_IBYTE_TO_CBYTE_H
00019 #define GNSS_SDR_IBYTE_TO_CBYTE_H
00020
00021 #include "conjugate_ic.h"
00022 #include "gnss_block_interface.h"
00023 #include "interleaved_byte_to_complex_byte.h"
00024 #include <gnuradio/blocks/file_sink.h>
00025 #include <stdint>
00026 #include <string>
00027
00028 /** \addtogroup Data_Type
00029  * \{ */
00030 /** \addtogroup Data_type_adapters
```

```

00031  * \{ */
00032
00033
00034 class ConfigurationInterface;
00035
00036 /*!
00037  * \brief Adapts an I/Q interleaved byte (unsigned char) sample stream
00038  * into a std::complex<unsigned char> stream
00039  */
00040 class IbyteToCbyte : public GNSSBlockInterface
00041 {
00042 public:
00043     IbyteToCbyte(const ConfigurationInterface* configuration,
00044                 const std::string& role, unsigned int in_streams,
00045                 unsigned int out_streams);
00046
00047     ~IbyteToCbyte() = default;
00048
00049     inline std::string role() override
00050     {
00051         return role_;
00052     }
00053
00054     /*! Returns "Ibyte_To_Cbyte"
00055     inline std::string implementation() override
00056     {
00057         return "Ibyte_To_Cbyte";
00058     }
00059
00060     inline size_t item_size() override
00061     {
00062         return 2 * sizeof(int8_t);
00063     }
00064
00065     void connect(gr::top_block_sptr top_block) override;
00066     void disconnect(gr::top_block_sptr top_block) override;
00067     gr::basic_block_sptr get_left_block() override;
00068     gr::basic_block_sptr get_right_block() override;
00069
00070 private:
00071     interleaved_byte_to_complex_byte_sptr ibyte_to_cbyte_;
00072     conjugate_ic_sptr conjugate_ic_;
00073     gr::blocks::file_sink::sptr file_sink_;
00074     std::string dump_filename_;
00075     std::string input_item_type_;
00076     std::string output_item_type_;
00077     std::string role_;
00078     unsigned int in_streams_;
00079     unsigned int out_streams_;
00080     bool inverted_spectrum;
00081     bool dump_;
00082 };
00083
00084
00085 /** \} */
00086 /** \} */
00087 #endif // GNSS_SDR_IBYTE_TO_CBYTE_H

```

## 11.97 ibyte\_to\_complex.h File Reference

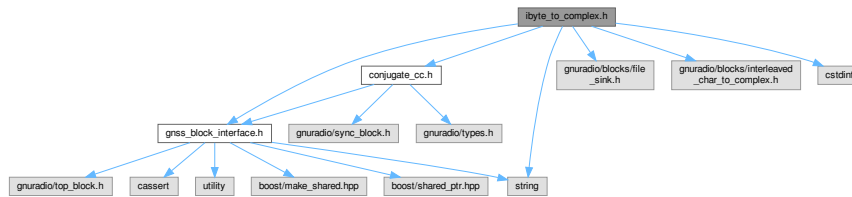
Adapts an I/Q interleaved byte integer sample stream to a gr\_complex (float) stream.

```

#include "conjugate_cc.h"
#include "gnss_block_interface.h"
#include <gnuradio/blocks/file_sink.h>
#include <gnuradio/blocks/interleaved_char_to_complex.h>
#include <cstdint>
#include <string>

```

Include dependency graph for `ibyte_to_complex.h`:



## Classes

- class [IbyteToComplex](#)

*Adapts an I/Q interleaved byte integer sample stream to a `gr_complex` (float) stream.*

### 11.97.1 Detailed Description

Adapts an I/Q interleaved byte integer sample stream to a `gr_complex` (float) stream.

#### Author

Javier Arribas, [jarribas\(at\)cttc.es](mailto:jarribas(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [ibyte\\_to\\_complex.h](#).

---

## 11.98 ibyte\_to\_complex.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file ibyte_to_complex.h
00003  * \brief Adapts an I/Q interleaved byte integer sample stream to a gr_complex (float) stream
00004  * \author Javier Arribas, jarribas(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_IBYTE_TO_COMPLEX_H
00018 #define GNSS_SDR_IBYTE_TO_COMPLEX_H
00019
00020 #include "conjugate_cc.h"
00021 #include "gnss_block_interface.h"
00022 #include <gnuradio/blocks/file_sink.h>
00023 #include <gnuradio/blocks/interleaved_char_to_complex.h>
00024 #include <cstdint>
00025 #include <string>
00026
00027 /** \addtogroup Data_Type
00028  * \{ */
00029 /** \addtogroup Data_type_adapters
00030  * \{ */
00031
00032
00033 class ConfigurationInterface;
00034
00035 /*!
00036 * \brief Adapts an I/Q interleaved byte integer sample stream to a gr_complex (float) stream
00037 *
00038 */
00039 class IbyteToComplex : public GNSSBlockInterface
00040 {

```

```

00041 public:
00042     IbyteToComplex(const ConfigurationInterface* configuration,
00043         const std::string& role, unsigned int in_streams,
00044         unsigned int out_streams);
00045
00046     ~IbyteToComplex() = default;
00047
00048     inline std::string role() override
00049     {
00050         return role_;
00051     }
00052
00053     //! Returns "Ibyte_To_Complex"
00054     inline std::string implementation() override
00055     {
00056         return "Ibyte_To_Complex";
00057     }
00058
00059     inline size_t item_size() override
00060     {
00061         return 2 * sizeof(int8_t);
00062     }
00063
00064     void connect(gr::top_block_sptr top_block) override;
00065     void disconnect(gr::top_block_sptr top_block) override;
00066     gr::basic_block_sptr get_left_block() override;
00067     gr::basic_block_sptr get_right_block() override;
00068
00069 private:
00070     gr::blocks::interleaved_char_to_complex::sptr gr_interleaved_char_to_complex_;
00071     conjugate_cc_sptr conjugate_cc_;
00072     gr::blocks::file_sink::sptr file_sink_;
00073     std::string dump_filename_;
00074     std::string input_item_type_;
00075     std::string output_item_type_;
00076     std::string role_;
00077     unsigned int in_streams_;
00078     unsigned int out_streams_;
00079     bool inverted_spectrum;
00080     bool dump_;
00081 };
00082
00083
00084 /** \} */
00085 /** \} */
00086 #endif // GNSS_SDR_IBYTE_TO_COMPLEX_H

```

## 11.99 ibyte\_to\_cshort.h File Reference

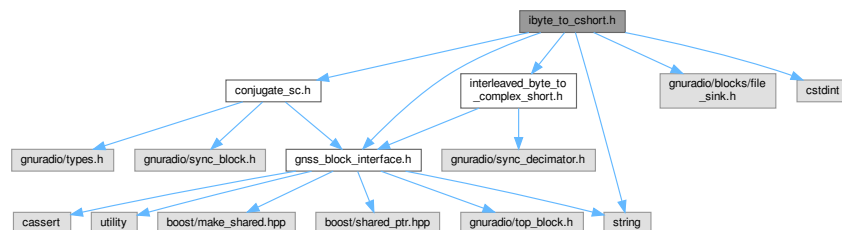
Adapts a short interleaved sample stream into a `std::complex<short>` stream.

```

#include "conjugate_sc.h"
#include "gnss_block_interface.h"
#include "interleaved_byte_to_complex_short.h"
#include <gnuradio/blocks/file_sink.h>
#include <cstdint>
#include <string>

```

Include dependency graph for `ibyte_to_cshort.h`:



### Classes

- class [IbyteToCshort](#)

*Adapts a short integer (16 bits) interleaved sample stream into a `std::complex<short>` stream.*

### 11.99.1 Detailed Description

Adapts a short interleaved sample stream into a `std::complex<short>` stream.

Author

Carles Fernandez-Prades, cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [ibyte\\_to\\_cshort.h](#).

---

## 11.100 ibyte\_to\_cshort.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002 * \file ibyte_to_cshort.h
00003 * \brief Adapts a short interleaved sample stream into a std::complex<short> stream
00004 * \author Carles Fernandez-Prades, cfernandez(at)cttc.es
00005 *
00006 * -----
00007 *
00008 * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009 * This file is part of GNSS-SDR.
00010 *
00011 * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012 * SPDX-License-Identifier: GPL-3.0-or-later
00013 *
00014 * -----
00015 */
00016
00017 #ifndef GNSS_SDR_IBYTE_TO_CSHORT_H
00018 #define GNSS_SDR_IBYTE_TO_CSHORT_H
00019
00020 #include "conjugate_sc.h"
00021 #include "gnss_block_interface.h"
00022 #include "interleaved_byte_to_complex_short.h"
00023 #include <gnuradio/blocks/file_sink.h>
00024 #include <stdint>
00025 #include <string>
00026
00027 /** \addtogroup Data_Type
00028 * \{ */
00029 /** \addtogroup Data_type_adapters
00030 * \{ */
00031
00032
00033 class ConfigurationInterface;
00034
00035 /*!
00036 * \brief Adapts a short integer (16 bits) interleaved sample stream into a std::complex<short> stream
00037 *
00038 */
00039 class IbyteToCshort : public GNSSBlockInterface
00040 {
00041 public:
00042     IbyteToCshort(const ConfigurationInterface* configuration,
00043                   const std::string& role, unsigned int in_streams,
00044                   unsigned int out_streams);
00045
00046     ~IbyteToCshort() = default;
00047
00048     inline std::string role() override
00049     {
00050         return role_;
00051     }
00052
00053     ///! Returns "Ibyte_To_Cshort"
00054     inline std::string implementation() override
00055     {
00056         return "Ibyte_To_Cshort";
00057     }
00058
00059     inline size_t item_size() override
00060     {
00061         return 2 * sizeof(int8_t);
00062     }
00063

```

```

00064     void connect(gr::top_block_sptr top_block) override;
00065     void disconnect(gr::top_block_sptr top_block) override;
00066     gr::basic_block_sptr get_left_block() override;
00067     gr::basic_block_sptr get_right_block() override;
00068
00069 private:
00070     interleaved_byte_to_complex_short_sptr interleaved_byte_to_complex_short_;
00071     conjugate_sc_sptr conjugate_sc_;
00072     gr::blocks::file_sink::sptr file_sink_;
00073     std::string dump_filename_;
00074     std::string input_item_type_;
00075     std::string output_item_type_;
00076     std::string role_;
00077     unsigned int in_streams_;
00078     unsigned int out_streams_;
00079     bool inverted_spectrum;
00080     bool dump_;
00081 };
00082
00083
00084 /** \} */
00085 /** \} */
00086 #endif // GNSS_SDR_IBYTE_TO_CSHORT_H

```

## 11.101 ishort\_to\_complex.h File Reference

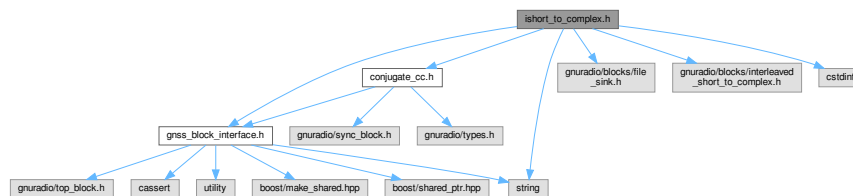
Adapts an I/Q interleaved short integer sample stream to a `gr_complex` (float) stream.

```

#include "conjugate_cc.h"
#include "gnss_block_interface.h"
#include <gnuradio/blocks/file_sink.h>
#include <gnuradio/blocks/interleaved_short_to_complex.h>
#include <cstdint>
#include <string>

```

Include dependency graph for `ishort_to_complex.h`:



### Classes

- class [IshortToComplex](#)

*Adapts an I/Q interleaved short integer sample stream to a `gr_complex` (float) stream.*

### 11.101.1 Detailed Description

Adapts an I/Q interleaved short integer sample stream to a `gr_complex` (float) stream.

#### Author

Javier Arribas, [jarribas\(at\)cttc.es](mailto:jarribas(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [ishort\\_to\\_complex.h](#).

---

## 11.102 ishort\_to\_complex.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file ishort_to_complex.h
00003  * \brief Adapts an I/Q interleaved short integer sample stream to a gr_complex (float) stream
00004  * \author Javier Arribas, jarribas(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_ISHORT_TO_COMPLEX_H
00018 #define GNSS_SDR_ISHORT_TO_COMPLEX_H
00019
00020 #include "conjugate_cc.h"
00021 #include "gnss_block_interface.h"
00022 #include <gnuradio/blocks/file_sink.h>
00023 #include <gnuradio/blocks/interleaved_short_to_complex.h>
00024 #include <stdint>
00025 #include <string>
00026
00027 /** \addtogroup Data_Type
00028  * \{ */
00029 /** \addtogroup Data_type_adapters
00030  * \{ */
00031
00032
00033 class ConfigurationInterface;
00034
00035 /*!
00036 * \brief Adapts an I/Q interleaved short integer sample stream to a gr_complex (float) stream
00037 *
00038 */
00039 class IshortToComplex : public GNSSBlockInterface
00040 {
00041 public:
00042     IshortToComplex(const ConfigurationInterface* configuration,
00043                     const std::string& role, unsigned int in_streams,
00044                     unsigned int out_streams);
00045
00046     ~IshortToComplex() = default;
00047
00048     inline std::string role() override
00049     {
00050         return role_;
00051     }
00052
00053     ///! Returns "Ishort_To_Complex"
00054     inline std::string implementation() override
00055     {
00056         return "Ishort_To_Complex";
00057     }
00058
00059     inline size_t item_size() override
00060     {
00061         return 2 * sizeof(int16_t);
00062     }
00063
00064     void connect(gr::top_block_sptr top_block) override;
00065     void disconnect(gr::top_block_sptr top_block) override;
00066     gr::basic_block_sptr get_left_block() override;
00067     gr::basic_block_sptr get_right_block() override;
00068
00069 private:
00070     gr::blocks::interleaved_short_to_complex::sptr gr_interleaved_short_to_complex_;
00071     conjugate_cc_sptr conjugate_cc_;
00072     gr::blocks::file_sink::sptr file_sink_;
00073     std::string dump_filename_;
00074     std::string input_item_type_;
00075     std::string output_item_type_;
00076     std::string role_;
00077     unsigned int in_streams_;
00078     unsigned int out_streams_;
00079     bool inverted_spectrum;
00080     bool dump_;
00081 };
00082
00083
00084 /** \} */
00085 /** \} */
00086 #endif // GNSS_SDR_ISHORT_TO_COMPLEX_H

```

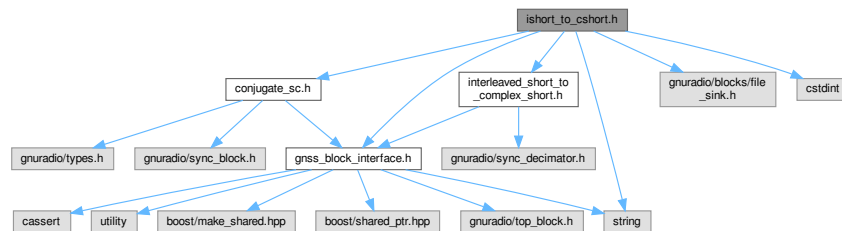


## 11.103 ishort\_to\_cshort.h File Reference

Adapts a short interleaved sample stream into a `std::complex<short>` stream.

```
#include "conjugate_sc.h"
#include "gnss_block_interface.h"
#include "interleaved_short_to_complex_short.h"
#include <gnuradio/blocks/file_sink.h>
#include <cstdint>
#include <string>
```

Include dependency graph for `ishort_to_cshort.h`:



### Classes

- class [IshortToCshort](#)

*Adapts a short integer (16 bits) interleaved sample stream into a `std::complex<short>` stream.*

### 11.103.1 Detailed Description

Adapts a short interleaved sample stream into a `std::complex<short>` stream.

#### Author

Carles Fernandez-Prades, [cfernandez\(at\)cttc.es](mailto:cfernandez(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [ishort\\_to\\_cshort.h](#).

---

## 11.104 ishort\_to\_cshort.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file ishort_to_cshort.h
00003  * \brief Adapts a short interleaved sample stream into a std::complex<short> stream
00004  * \author Carles Fernandez-Prades, cfernandez(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_ISHORT_TO_CSHORT_H
00018 #define GNSS_SDR_ISHORT_TO_CSHORT_H
00019
00020 #include "conjugate_sc.h"
00021 #include "gnss_block_interface.h"
00022 #include "interleaved_short_to_complex_short.h"
00023 #include <gnuradio/blocks/file_sink.h>
00024 #include <cstdint>
```

```

00025 #include <string>
00026
00027 /** \addtogroup Data_Type
00028 * \{ */
00029 /** \addtogroup Data_type_adapters
00030 * \{ */
00031
00032
00033 class ConfigurationInterface;
00034
00035 /*!
00036 * \brief Adapts a short integer (16 bits) interleaved sample stream into a std::complex<short> stream
00037 *
00038 */
00039 class IshortToCshort : public GNSSBlockInterface
00040 {
00041 public:
00042     IshortToCshort(const ConfigurationInterface* configuration,
00043         const std::string& role, unsigned int in_streams,
00044         unsigned int out_streams);
00045
00046     ~IshortToCshort() = default;
00047
00048     inline std::string role() override
00049     {
00050         return role_;
00051     }
00052
00053     /*! Returns "Ishort_To_Cshort"
00054     inline std::string implementation() override
00055     {
00056         return "Ishort_To_Cshort";
00057     }
00058
00059     inline size_t item_size() override
00060     {
00061         return 2 * sizeof(int16_t);
00062     }
00063
00064     void connect(gr::top_block_sptr top_block) override;
00065     void disconnect(gr::top_block_sptr top_block) override;
00066     gr::basic_block_sptr get_left_block() override;
00067     gr::basic_block_sptr get_right_block() override;
00068
00069 private:
00070     interleaved_short_to_complex_short_sptr interleaved_short_to_complex_short_;
00071     conjugate_sc_sptr conjugate_sc_;
00072     gr::blocks::file_sink::sptr file_sink_;
00073     std::string dump_filename_;
00074     std::string input_item_type_;
00075     std::string output_item_type_;
00076     std::string role_;
00077     unsigned int in_streams_;
00078     unsigned int out_streams_;
00079     bool inverted_spectrum;
00080     bool dump_;
00081 };
00082
00083
00084 /** \} */
00085 /** \} */
00086 #endif // GNSS_SDR_ISHORT_TO_CSHORT_H

```

## 11.105 cshort\_to\_gr\_complex.h File Reference

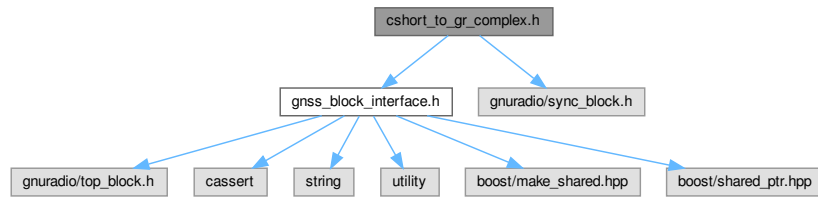
Adapts a complex short (16 + 16 bits) sample stream into a std::complex<float> stream (32 + 32 bits)

```

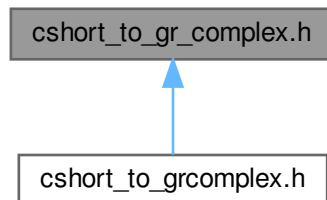
#include "gnss_block_interface.h"
#include <gnuradio/sync_block.h>

```

Include dependency graph for cshort\_to\_gr\_complex.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [cshort\\_to\\_gr\\_complex](#)

*This class adapts a short (16-bits) interleaved sample stream into a `std::complex<float>` stream.*

## Typedefs

- using [cshort\\_to\\_gr\\_complex\\_sptr](#) = `gnss_shared_ptr<cshort\_to\_gr\_complex>`

## Functions

- `cshort_to_gr_complex_sptr make_cshort_to_gr_complex ()`

### 11.105.1 Detailed Description

Adapts a complex short (16 + 16 bits) sample stream into a `std::complex<float>` stream (32 + 32 bits)

#### Author

Carles Fernandez Prades, 2014 cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [cshort\\_to\\_gr\\_complex.h](#).

---

## 11.106 cshort\_to\_gr\_complex.h

[Go to the documentation of this file.](#)

00001 /\*!

```

00002  * \file cshort_to_gr_complex.h
00003  * \brief Adapts a complex short (16 + 16 bits) sample stream into a
00004  *        std::complex<float> stream (32 + 32 bits)
00005  * \author Carles Fernandez Prades, 2014 cfernandez(at)cttc.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_CSHORT_TO_GR_COMPLEX_H
00019 #define GNSS_SDR_CSHORT_TO_GR_COMPLEX_H
00020
00021 #include "gnss_block_interface.h"
00022 #include <gnuradio/sync_block.h>
00023
00024 /** \addtogroup Data_Type
00025  *  \{ */
00026 /** \addtogroup data_type_gnuradio_blocks
00027  *  \{ */
00028
00029
00030 class cshort_to_gr_complex;
00031
00032 using cshort_to_gr_complex_sptr = gnss_shared_ptr<csshort_to_gr_complex>;
00033
00034 cshort_to_gr_complex_sptr make_cshort_to_gr_complex();
00035
00036 /*!
00037  * \brief This class adapts a short (16-bits) interleaved sample stream
00038  * into a std::complex<float> stream
00039  */
00040 class cshort_to_gr_complex : public gr::sync_block
00041 {
00042 public:
00043     int work(int noutput_items,
00044             gr_vector_const_void_star &input_items,
00045             gr_vector_void_star &output_items);
00046
00047 private:
00048     friend cshort_to_gr_complex_sptr make_cshort_to_gr_complex();
00049     cshort_to_gr_complex();
00050 };
00051
00052
00053 /** \} */
00054 /** \} */
00055 #endif // GNSS_SDR_CSHORT_TO_GR_COMPLEX_H

```

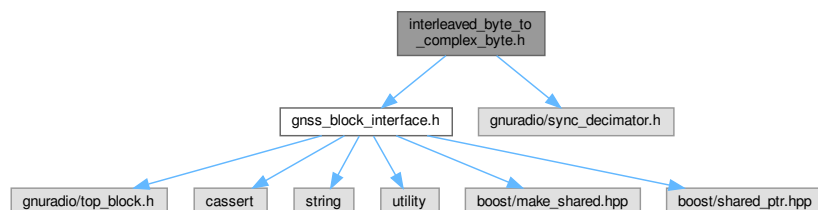
## 11.107 interleaved\_byte\_to\_complex\_byte.h File Reference

Adapts an 8-bits interleaved sample stream into a 16-bits complex stream.

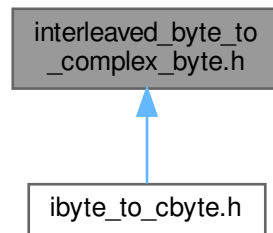
```
#include "gnss_block_interface.h"
```

```
#include <gnuradio/sync_decimator.h>
```

Include dependency graph for interleaved\_byte\_to\_complex\_byte.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [interleaved\\_byte\\_to\\_complex\\_byte](#)

*This class adapts an 8-bits interleaved sample stream into a 16-bits complex stream (std::complex<unsigned char>)*

## Typedefs

- using [interleaved\\_byte\\_to\\_complex\\_byte\\_sptr](#) = gnss\_shared\_ptr<[interleaved\\_byte\\_to\\_complex\\_byte](#)>

## Functions

- interleaved\_byte\_to\_complex\_byte\_sptr **make\_interleaved\_byte\_to\_complex\_byte** ()

### 11.107.1 Detailed Description

Adapts an 8-bits interleaved sample stream into a 16-bits complex stream.

#### Author

Carles Fernandez Prades, cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [interleaved\\_byte\\_to\\_complex\\_byte.h](#).

---

## 11.108 interleaved\_byte\_to\_complex\_byte.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file interleaved_byte_to_complex_byte.h
00003  * \brief Adapts an 8-bits interleaved sample stream into a 16-bits complex stream
00004  * \author Carles Fernandez Prades, cfernandez(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_INTERLEAVED_BYTE_TO_COMPLEX_BYTE_H
00018 #define GNSS_SDR_INTERLEAVED_BYTE_TO_COMPLEX_BYTE_H
00019
00020 #include "gnss_block_interface.h"
  
```

```

00021 #include <gnuradio/sync_decimator.h>
00022
00023 /** \addtogroup Data_Type
00024 * \{ */
00025 /** \addtogroup data_type_gnuradio_blocks data_type_gr_blocks
00026 * GNU Radio Blocks for data type conversion
00027 * \{ */
00028
00029
00030 class interleaved_byte_to_complex_byte;
00031
00032 using interleaved_byte_to_complex_byte_sptr = gnss_shared_ptr<interleaved_byte_to_complex_byte>;
00033
00034 interleaved_byte_to_complex_byte_sptr make_interleaved_byte_to_complex_byte();
00035
00036 /*!
00037 * \brief This class adapts an 8-bits interleaved sample stream
00038 * into a 16-bits complex stream (std::complex<unsigned char>)
00039 */
00040 class interleaved_byte_to_complex_byte : public gr::sync_decimator
00041 {
00042 public:
00043     int work(int noutput_items,
00044             gr_vector_const_void_star &input_items,
00045             gr_vector_void_star &output_items);
00046 private:
00047     friend interleaved_byte_to_complex_byte_sptr make_interleaved_byte_to_complex_byte();
00048     interleaved_byte_to_complex_byte();
00049 };
00050
00051
00052
00053 /** \} */
00054 /** \} */
00055 #endif // GNSS_SDR_INTERLEAVED_BYTE_TO_COMPLEX_BYTE_H

```

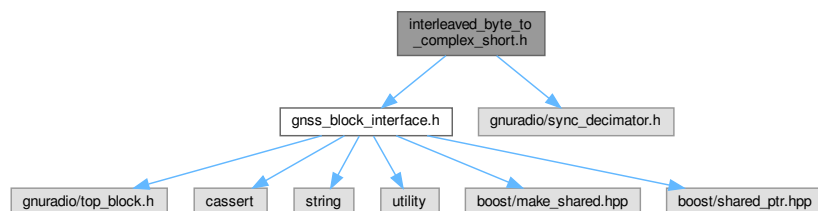
## 11.109 interleaved\_byte\_to\_complex\_short.h File Reference

Adapts a byte (8-bits) interleaved sample stream into a `std::complex<short>` stream.

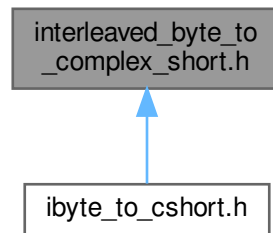
```
#include "gnss_block_interface.h"
```

```
#include <gnuradio/sync_decimator.h>
```

Include dependency graph for `interleaved_byte_to_complex_short.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [interleaved\\_byte\\_to\\_complex\\_short](#)

*This class adapts a short (16-bits) interleaved sample stream into a `std::complex<short>` stream.*

## Typedefs

- using [interleaved\\_byte\\_to\\_complex\\_short\\_sptr](#) = `gnss_shared_ptr<interleaved_byte_to_complex_short>`

## Functions

- `interleaved_byte_to_complex_short_sptr make_interleaved_byte_to_complex_short ()`

### 11.109.1 Detailed Description

Adapts a byte (8-bits) interleaved sample stream into a `std::complex<short>` stream.

#### Author

Javier Arribas (jarribas(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [interleaved\\_byte\\_to\\_complex\\_short.h](#).

---

## 11.110 interleaved\_byte\_to\_complex\_short.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file interleaved_byte_to_complex_short.h
00003  * \brief Adapts a byte (8-bits) interleaved sample stream into a std::complex<short> stream
00004  * \author Javier Arribas (jarribas(at)cttc.es)
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_INTERLEAVED_BYTE_TO_COMPLEX_SHORT_H
00018 #define GNSS_SDR_INTERLEAVED_BYTE_TO_COMPLEX_SHORT_H
00019
00020 #include "gnss_block_interface.h"
  
```

```

00021 #include <gnuradio/sync_decimator.h>
00022
00023
00024 /** \addtogroup Data_Type
00025  * \{ */
00026 /** \addtogroup data_type_gnuradio_blocks
00027  * \{ */
00028
00029
00030 class interleaved_byte_to_complex_short;
00031
00032 using interleaved_byte_to_complex_short_sptr = gnss_shared_ptr<interleaved_byte_to_complex_short>;
00033
00034 interleaved_byte_to_complex_short_sptr make_interleaved_byte_to_complex_short();
00035
00036 /*!
00037  * \brief This class adapts a short (16-bits) interleaved sample stream
00038  * into a std::complex<short> stream
00039  */
00040 class interleaved_byte_to_complex_short : public gr::sync_decimator
00041 {
00042 public:
00043     int work(int noutput_items,
00044             gr_vector_const_void_star &input_items,
00045             gr_vector_void_star &output_items);
00046 private:
00047     friend interleaved_byte_to_complex_short_sptr make_interleaved_byte_to_complex_short();
00048     interleaved_byte_to_complex_short();
00049 };
00050
00051
00052
00053 /** \} */
00054 /** \} */
00055 #endif // GNSS_SDR_INTERLEAVED_BYTE_TO_COMPLEX_SHORT_H

```

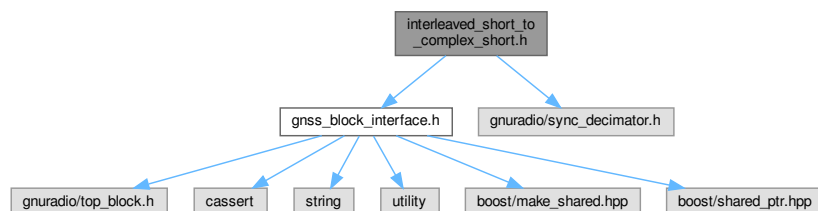
## 11.111 interleaved\_short\_to\_complex\_short.h File Reference

Adapts a short (16-bits) interleaved sample stream into a std::complex<short> stream.

```
#include "gnss_block_interface.h"
```

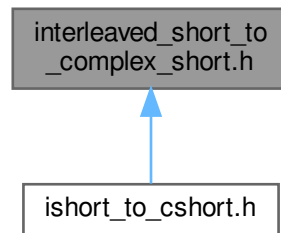
```
#include <gnuradio/sync_decimator.h>
```

Include dependency graph for interleaved\_short\_to\_complex\_short.h:





This graph shows which files directly or indirectly include this file:



## Classes

- class [interleaved\\_short\\_to\\_complex\\_short](#)

*This class adapts a short (16-bits) interleaved sample stream into a `std::complex<short>` stream.*

## Typedefs

- using [interleaved\\_short\\_to\\_complex\\_short\\_sptr](#) = `gnss_shared_ptr<interleaved_short_to_complex_short>`

## Functions

- `interleaved_short_to_complex_short_sptr make_interleaved_short_to_complex_short ()`

### 11.111.1 Detailed Description

Adapts a short (16-bits) interleaved sample stream into a `std::complex<short>` stream.

#### Author

Carles Fernandez Prades, [cfernandez\(at\)cttc.es](mailto:cfernandez(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [interleaved\\_short\\_to\\_complex\\_short.h](#).

---

## 11.112 interleaved\_short\_to\_complex\_short.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file interleaved_short_to_complex_short.h
00003  * \brief Adapts a short (16-bits) interleaved sample stream into a std::complex<short> stream
00004  * \author Carles Fernandez Prades, cfernandez(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_INTERLEAVED_SHORT_TO_COMPLEX_SHORT_H
00018 #define GNSS_SDR_INTERLEAVED_SHORT_TO_COMPLEX_SHORT_H
00019
00020 #include "gnss_block_interface.h"
  
```

```

00021 #include <gnuradio/sync_decimator.h>
00022
00023 /** \addtogroup Data_Type
00024 * \{ */
00025 /** \addtogroup data_type_gnuradio_blocks
00026 * \{ */
00027
00028
00029 class interleaved_short_to_complex_short;
00030
00031 using interleaved_short_to_complex_short_sptr = gnss_shared_ptr<interleaved_short_to_complex_short>;
00032
00033 interleaved_short_to_complex_short_sptr make_interleaved_short_to_complex_short();
00034
00035 /*!
00036 * \brief This class adapts a short (16-bits) interleaved sample stream
00037 * into a std::complex<short> stream
00038 */
00039 class interleaved_short_to_complex_short : public gr::sync_decimator
00040 {
00041 public:
00042     int work(int noutput_items,
00043             gr_vector_const_void_star &input_items,
00044             gr_vector_void_star &output_items);
00045 private:
00046     friend interleaved_short_to_complex_short_sptr make_interleaved_short_to_complex_short();
00047     interleaved_short_to_complex_short();
00048 };
00049
00050
00051
00052 /** \} */
00053 /** \} */
00054 #endif // GNSS_SDR_INTERLEAVED_SHORT_TO_COMPLEX_SHORT_H

```

### 11.113 beamformer\_filter.h File Reference

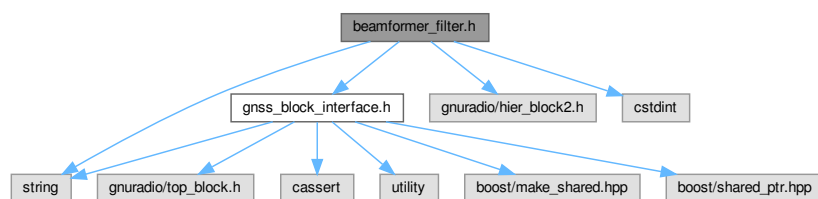
Interface of an adapter of a digital beamformer.

```

#include "gnss_block_interface.h"
#include <gnuradio/hier_block2.h>
#include <stdint>
#include <string>

```

Include dependency graph for beamformer\_filter.h:



#### Classes

- class [BeamformerFilter](#)

Interface of an adapter of a digital beamformer block to a [GNSSBlockInterface](#).

#### 11.113.1 Detailed Description

Interface of an adapter of a digital beamformer.

##### Author

Javier Arribas jarribas (at) cttc.es

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [beamformer\\_filter.h](#).

## 11.114 beamformer\_filter.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file beamformer_filter.h
00003  * \brief Interface of an adapter of a digital beamformer
00004  * \author Javier Arribas jarribas (at) cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_BEAMFORMER_FILTER_H
00019 #define GNSS_SDR_BEAMFORMER_FILTER_H
00020
00021 #include "gnss_block_interface.h"
00022 #include <gnuradio/hier_block2.h>
00023 #include <stdint>
00024 #include <string>
00025
00026 /** \addtogroup Input_Filter
00027  * \{ */
00028 /** \addtogroup Input_filter_adapters
00029  * \{ */
00030
00031
00032 class ConfigurationInterface;
00033
00034 /*!
00035  * \brief Interface of an adapter of a digital beamformer block
00036  * to a GNSSBlockInterface
00037  */
00038 class BeamformerFilter : public GNSSBlockInterface
00039 {
00040 public:
00041     BeamformerFilter(const ConfigurationInterface* configuration,
00042                     const std::string& role, unsigned int in_stream,
00043                     unsigned int out_stream);
00044
00045     ~BeamformerFilter() = default;
00046
00047     inline std::string role() override
00048     {
00049         return role_;
00050     }
00051
00052     //! returns "Beamformer_Filte"
00053     inline std::string implementation() override
00054     {
00055         return "Beamformer_Filter";
00056     }
00057
00058     inline size_t item_size() override
00059     {
00060         return item_size_;
00061     }
00062
00063     void connect(gr::top_block_sptr top_block) override;
00064     void disconnect(gr::top_block_sptr top_block) override;
00065     gr::basic_block_sptr get_left_block() override;
00066     gr::basic_block_sptr get_right_block() override;
00067
00068 private:
00069     gr::block_sptr beamformer_;
00070     gr::block_sptr file_sink_;
00071     std::string role_;
00072     std::string item_type_;
00073     std::string dump_filename_;
00074     size_t item_size_;
00075     unsigned int in_stream_;
00076     unsigned int out_stream_;
00077     bool dump_;
00078 };
00079
00080
00081 /** \} */
00082 /** \} */
00083 #endif // GNSS_SDR_BEAMFORMER_FILTER_H

```

## 11.115 fir\_filter.h File Reference

Adapts a gnuradio gr\_fir\_filter designed with pm\_remez.

```
#include "byte_x2_to_complex_byte.h"
#include "complex_byte_to_float_x2.h"
#include "cshort_to_float_x2.h"
#include "gnss_block_interface.h"
#include "short_x2_to_cshort.h"
#include <gnuradio/blocks/file_sink.h>
#include <gnuradio/blocks/float_to_char.h>
#include <gnuradio/blocks/float_to_complex.h>
#include <gnuradio/blocks/float_to_short.h>
#include <gnuradio/gr_complex.h>
#include <gnuradio/filter/fir_filter_ccf.h>
#include <gnuradio/filter/fir_filter_fff.h>
#include <cmath>
#include <string>
#include <vector>
```

Include dependency graph for fir\_filter.h:



### Classes

- class [FirFilter](#)

*This class adapts a GNU Radio gr\_fir\_filter designed with pm\_remez.*

### 11.115.1 Detailed Description

Adapts a gnuradio gr\_fir\_filter designed with pm\_remez.

#### Author

Luis Esteve, 2012. [luis\(at\)epsilon-formation.com](mailto:luis(at)epsilon-formation.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [fir\\_filter.h](#).

---

## 11.116 fir\_filter.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file fir_filter.h
00003  * \brief Adapts a gnuradio gr_fir_filter designed with pm_remez
00004  * \author Luis Esteve, 2012. luis\(at\)epsilon-formation.com
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_FIR_FILTER_H
00019 #define GNSS_SDR_FIR_FILTER_H
```

```

00020
00021 #include "byte_x2_to_complex_byte.h"
00022 #include "complex_byte_to_float_x2.h"
00023 #include "cshort_to_float_x2.h"
00024 #include "gnss_block_interface.h"
00025 #include "short_x2_to_cshort.h"
00026 #include <gnuradio/blocks/file_sink.h>
00027 #include <gnuradio/blocks/float_to_char.h>
00028 #include <gnuradio/blocks/float_to_complex.h>
00029 #include <gnuradio/blocks/float_to_short.h>
00030 #include <gnuradio/gr_complex.h>
00031 #ifdef GR_GREATER_38
00032 #include <gnuradio/filter/fir_filter_blk.h>
00033 #else
00034 #include <gnuradio/filter/fir_filter_ccf.h>
00035 #include <gnuradio/filter/fir_filter_fff.h>
00036 #endif
00037 #include <cmath>
00038 #include <string>
00039 #include <vector>
00040
00041 /** \addtogroup Input_Filter Input Filter
00042  * Classes for input signal filtering
00043  * \{ */
00044 /** \addtogroup Input_filter_adapters input_filter_adapters
00045  * Classes that wrap GNU Radio input filters with a GNSSBlockInterface
00046  * \{ */
00047
00048
00049 class ConfigurationInterface;
00050
00051 /*!
00052  * \brief This class adapts a GNU Radio gr_fir_filter designed with pm_remez
00053  *
00054  * See Parks-McClellan FIR filter design,
00055  * https://en.wikipedia.org/wiki/Parks-McClellan\_filter\_design\_algorithm
00056  * Calculates the optimal (in the Chebyshev/minimax sense) FIR filter impulse response
00057  * given a set of band edges, the desired response on those bands, and the weight given
00058  * to the error in those bands.
00059  */
00059 class FirFilter : public GNSSBlockInterface
00060 {
00061 public:
00062     /** Constructor
00063     FirFilter(const ConfigurationInterface* configuration,
00064             std::string role,
00065             unsigned int in_streams,
00066             unsigned int out_streams);
00067
00068     /** Destructor
00069     ~FirFilter() = default;
00070
00071     inline std::string role() override
00072     {
00073         return role_;
00074     }
00075
00076     /** Returns "Fir_Filter"
00077     inline std::string implementation() override
00078     {
00079         return "Fir_Filter";
00080     }
00081
00082     inline size_t item_size() override
00083     {
00084         return item_size_;
00085     }
00086
00087     void connect(gr::top_block_sptr top_block) override;
00088     void disconnect(gr::top_block_sptr top_block) override;
00089     gr::basic_block_sptr get_left_block() override;
00090     gr::basic_block_sptr get_right_block() override;
00091 private:
00092     void init();
00093
00094     gr::filter::fir_filter_ccf::sptr fir_filter_ccf_;
00095     gr::filter::fir_filter_fff::sptr fir_filter_fff_1_;
00096     gr::filter::fir_filter_fff::sptr fir_filter_fff_2_;
00097     gr::blocks::float_to_complex::sptr float_to_complex_;
00098     gr::blocks::float_to_short::sptr float_to_short_1_;
00099     gr::blocks::float_to_short::sptr float_to_short_2_;
00100     short_x2_to_cshort_sptr short_x2_to_cshort_;
00101     complex_byte_to_float_x2_sptr cbyte_to_float_x2_;
00102     byte_x2_to_complex_byte_sptr char_x2_cbyte_;
00103     cshort_to_float_x2_sptr cshort_to_float_x2_;
00104     gr::blocks::float_to_char::sptr float_to_char_1_;

```

```

00106     gr::blocks::float_to_char::sptr float_to_char_2_;
00107     gr::blocks::file_sink::sptr file_sink_;
00108     const ConfigurationInterface* config_;
00109     std::vector<float> taps_;
00110     std::string dump_filename_;
00111     std::string input_item_type_;
00112     std::string output_item_type_;
00113     std::string taps_item_type_;
00114     std::string role_;
00115     size_t item_size_;
00116     unsigned int in_streams_;
00117     unsigned int out_streams_;
00118     bool dump_;
00119 };
00120
00121
00122 /** \} */
00123 /** \} */
00124 #endif // GNSS_SDR_FIR_FILTER_H

```

## 11.117 freq\_xlating\_fir\_filter.h File Reference

Adapts a gnuradio `gr_freq_xlating_fir_filter` designed with `gr_remez`.

```

#include "complex_float_to_complex_byte.h"
#include "gnss_block_interface.h"
#include "short_x2_to_cshort.h"
#include <gnuradio/filter/freq_xlating_fir_filter_ccf.h>
#include <gnuradio/filter/freq_xlating_fir_filter_fcf.h>
#include <gnuradio/filter/freq_xlating_fir_filter_scf.h>
#include <gnuradio/blocks/char_to_short.h>
#include <gnuradio/blocks/complex_to_float.h>
#include <gnuradio/blocks/file_sink.h>
#include <gnuradio/blocks/float_to_short.h>
#include <string>
#include <vector>

```

Include dependency graph for `freq_xlating_fir_filter.h`:



### Classes

- class [FreqXlatingFirFilter](#)

*This class adapts a gnuradio `gr_freq_xlating_fir_filter` designed with `pm_remez`.*

### 11.117.1 Detailed Description

Adapts a gnuradio `gr_freq_xlating_fir_filter` designed with `gr_remez`.

#### Author

Luis Esteve, 2012. [luis\(at\)epsilon-formation.com](mailto:luis(at)epsilon-formation.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [freq\\_xlating\\_fir\\_filter.h](#).

---

## 11.118 freq\_xlating\_fir\_filter.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file freq_xlating_fir_filter.h
00003  * \brief Adapts a gnuradio gr_freq_xlating_fir_filter designed with gr_remez
00004  * \author Luis Esteve, 2012. luis(at)epsilon-formacion.com
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_FREQ_XLATING_FIR_FILTER_H
00019 #define GNSS_SDR_FREQ_XLATING_FIR_FILTER_H
00020
00021 #include "complex_float_to_complex_byte.h"
00022 #include "gnss_block_interface.h"
00023 #include "short_x2_to_cshort.h"
00024 #ifdef GR_GREATER_38
00025 #include <gnuradio/filter/freq_xlating_fir_filter.h>
00026 #else
00027 #include <gnuradio/filter/freq_xlating_fir_filter_ccf.h>
00028 #include <gnuradio/filter/freq_xlating_fir_filter_fcf.h>
00029 #include <gnuradio/filter/freq_xlating_fir_filter_scf.h>
00030 #endif
00031 #include <gnuradio/blocks/char_to_short.h>
00032 #include <gnuradio/blocks/complex_to_float.h>
00033 #include <gnuradio/blocks/file_sink.h>
00034 #include <gnuradio/blocks/float_to_short.h>
00035 #include <string>
00036 #include <vector>
00037
00038 /** \addtogroup Input_Filter
00039  * \{ */
00040 /** \addtogroup Input_filter_adapters
00041  * \{ */
00042
00043
00044 class ConfigurationInterface;
00045
00046 /*!
00047  * \brief This class adapts a gnuradio gr_freq_xlating_fir_filter designed with pm_remez
00048  *
00049  * Construct a FIR filter with the given taps and a composite frequency
00050  * translation that shifts intermediate_freq_ down to zero Hz. The frequency
00051  * translation logically comes before the filtering operation.
00052  *
00053  * See Parks-McClellan FIR filter design,
00054  * https://en.wikipedia.org/wiki/Parks-McClellan\_filter\_design\_algorithm
00055  * Calculates the optimal (in the Chebyshev/minimax sense) FIR filter impulse response
00056  * given a set of band edges, the desired response on those bands, and the weight given
00057  * to the error in those bands.
00058  */
00058 class FreqXlatingFirFilter : public GNSSBlockInterface
00059 {
00060 public:
00061     FreqXlatingFirFilter(const ConfigurationInterface* configuration,
00062         std::string role, unsigned int in_streams,
00063         unsigned int out_streams);
00064
00065     ~FreqXlatingFirFilter() = default;
00066
00067     inline std::string role() override
00068     {
00069         return role_;
00070     }
00071
00072     /*! Returns "Freq_Xlating_Fir_Filter"
00073     inline std::string implementation() override
00074     {
00075         return "Freq_Xlating_Fir_Filter";
00076     }
00077
00078     inline size_t item_size() override
00079     {
00080         return input_size_;
00081     }
00082
00083     void connect(gr::top_block_sptr top_block) override;
00084     void disconnect(gr::top_block_sptr top_block) override;
00085     gr::basic_block_sptr get_left_block() override;
00086     gr::basic_block_sptr get_right_block() override;

```

```

00087
00088 private:
00089     gr::filter::freq_xlating_fir_filter_ccf::sptr freq_xlating_fir_filter_ccf_;
00090     gr::filter::freq_xlating_fir_filter_fcf::sptr freq_xlating_fir_filter_fcf_;
00091     gr::filter::freq_xlating_fir_filter_scf::sptr freq_xlating_fir_filter_scf_;
00092     gr::blocks::complex_to_float::sptr complex_to_float_;
00093     gr::blocks::char_to_short::sptr gr_char_to_short_;
00094     gr::blocks::float_to_short::sptr float_to_short_1_;
00095     gr::blocks::float_to_short::sptr float_to_short_2_;
00096     short_x2_to_cshort_sptr short_x2_to_cshort_;
00097     complex_float_to_complex_byte_sptr complex_to_complex_byte_;
00098     gr::blocks::file_sink::sptr file_sink_;
00099     std::vector<float> taps_;
00100     std::string dump_filename_;
00101     std::string input_item_type_;
00102     std::string output_item_type_;
00103     std::string taps_item_type_;
00104     std::string role_;
00105     size_t input_size_;
00106     double intermediate_freq_;
00107     double sampling_freq_;
00108     int decimation_factor_;
00109     unsigned int in_streams_;
00110     unsigned int out_streams_;
00111     bool dump_;
00112 };
00113
00114
00115 /** \} */
00116 /** \} */
00117 #endif // GNSS_SDR_FREQ_XLATING_FIR_FILTER_H

```

## 11.119 notch\_filter.h File Reference

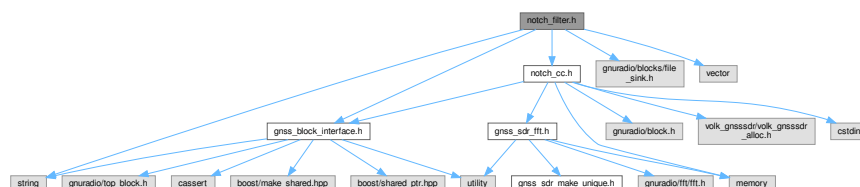
Adapter of a multistate [Notch](#) filter.

```

#include "gnss_block_interface.h"
#include "notch_cc.h"
#include <gnuradio/blocks/file_sink.h>
#include <string>
#include <vector>

```

Include dependency graph for notch\_filter.h:



### Classes

- class [NotchFilter](#)

### 11.119.1 Detailed Description

Adapter of a multistate [Notch](#) filter.

#### Author

Antonio Ramos, 2017. antonio.ramosdet(at)gmail.com

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [notch\\_filter.h](#).



## 11.120 notch\_filter.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file notch_filter.h
00003  * \brief Adapter of a multistate Notch filter
00004  * \author Antonio Ramos, 2017. antonio.ramosdet(at)gmail.com
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_NOTCH_FILTER_H
00019 #define GNSS_SDR_NOTCH_FILTER_H
00020
00021 #include "gnss_block_interface.h"
00022 #include "notch_cc.h"
00023 #include <gnuradio/blocks/file_sink.h>
00024 #include <string>
00025 #include <vector>
00026
00027 /** \addtogroup Input_Filter
00028  * \{ */
00029 /** \addtogroup Input_filter_adapters
00030  * \{ */
00031
00032
00033 class ConfigurationInterface;
00034
00035 class NotchFilter : public GNSSBlockInterface
00036 {
00037 public:
00038     NotchFilter(const ConfigurationInterface* configuration,
00039                 const std::string& role, unsigned int in_streams,
00040                 unsigned int out_streams);
00041
00042     ~NotchFilter() = default;
00043
00044     std::string role()
00045     {
00046         return role_;
00047     }
00048
00049     /*! Returns "Notch_Filter"
00050     std::string implementation()
00051     {
00052         return "Notch_Filter";
00053     }
00054
00055     size_t item_size()
00056     {
00057         return item_size_;
00058     }
00059
00060     void connect(gr::top_block_sptr top_block);
00061     void disconnect(gr::top_block_sptr top_block);
00062     gr::basic_block_sptr get_left_block();
00063     gr::basic_block_sptr get_right_block();
00064
00065 private:
00066     notch_sptr notch_filter_;
00067     gr::blocks::file_sink::sptr file_sink_;
00068     std::string dump_filename_;
00069     std::string role_;
00070     std::string item_type_;
00071     size_t item_size_;
00072     unsigned int in_streams_;
00073     unsigned int out_streams_;
00074     bool dump_;
00075 };
00076
00077
00078 /** \} */
00079 /** \} */
00080 #endif // GNSS_SDR_NOTCH_FILTER_H

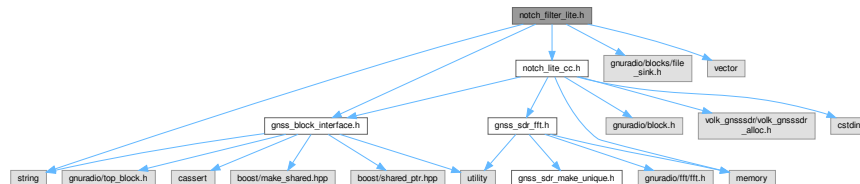
```

## 11.121 notch\_filter\_lite.h File Reference

Adapts a light version of a multistate notch filter.

```
#include "gnss_block_interface.h"
#include "notch_lite_cc.h"
#include <gnuradio/blocks/file_sink.h>
#include <string>
#include <vector>
```

Include dependency graph for notch\_filter\_lite.h:



### Classes

- class [NotchFilterLite](#)

### 11.121.1 Detailed Description

Adapts a light version of a multistate notch filter.

#### Author

Antonio Ramos, 2017. antonio.ramosdet(at)gmail.com

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [notch\\_filter\\_lite.h](#).

---

## 11.122 notch\_filter\_lite.h

[Go to the documentation of this file.](#)

```
00001  /*!
00002  * \file notch_filter_lite.h
00003  * \brief Adapts a light version of a multistate notch filter
00004  * \author Antonio Ramos, 2017. antonio.ramosdet(at)gmail.com
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_NOTCH_FILTER_LITE_H
00019 #define GNSS_SDR_NOTCH_FILTER_LITE_H
00020
00021 #include "gnss_block_interface.h"
00022 #include "notch_lite_cc.h"
00023 #include <gnuradio/blocks/file_sink.h>
00024 #include <string>
00025 #include <vector>
00026
00027 /** \addtogroup Input_Filter
00028  * \{ */
00029 /** \addtogroup Input_filter_adapters
00030  * \{ */
00031
```

```

00032
00033 class ConfigurationInterface;
00034
00035 class NotchFilterLite : public GNSSBlockInterface
00036 {
00037 public:
00038     NotchFilterLite(const ConfigurationInterface* configuration,
00039                     const std::string& role, unsigned int in_streams,
00040                     unsigned int out_streams);
00041
00042     ~NotchFilterLite() = default;
00043
00044     std::string role()
00045     {
00046         return role_;
00047     }
00048
00049     //! Returns "Notch_Filter_Lite"
00050     std::string implementation()
00051     {
00052         return "Notch_Filter_Lite";
00053     }
00054
00055     size_t item_size()
00056     {
00057         return item_size_;
00058     }
00059
00060     void connect(gr::top_block_sptr top_block);
00061     void disconnect(gr::top_block_sptr top_block);
00062     gr::basic_block_sptr get_left_block();
00063     gr::basic_block_sptr get_right_block();
00064
00065 private:
00066     notch_lite_sptr notch_filter_lite_;
00067     gr::blocks::file_sink::sptr file_sink_;
00068     std::string dump_filename_;
00069     std::string role_;
00070     std::string item_type_;
00071     size_t item_size_;
00072     unsigned int in_streams_;
00073     unsigned int out_streams_;
00074     bool dump_;
00075 };
00076
00077
00078 /** \} */
00079 /** \} */
00080 #endif // GNSS_SDR_NOTCH_FILTER_LITE_H

```

## 11.123 pulse\_blanking\_filter.h File Reference

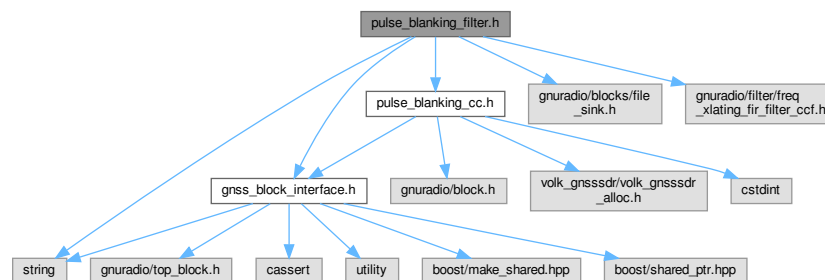
Instantiates the GNSS-SDR pulse blanking filter.

```

#include "gnss_block_interface.h"
#include "pulse_blanking_cc.h"
#include <gnuradio/blocks/file_sink.h>
#include <gnuradio/filter/freq_xlating_fir_filter_ccf.h>
#include <string>

```

Include dependency graph for pulse\_blanking\_filter.h:



## Classes

- class [PulseBlankingFilter](#)

### 11.123.1 Detailed Description

Instantiates the GNSS-SDR pulse blanking filter.

#### Author

Javier Arribas 2017 Antonio Ramos 2017

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [pulse\\_blanking\\_filter.h](#).

---

## 11.124 pulse\_blanking\_filter.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file pulse_blanking_filter.h
00003  * \brief Instantiates the GNSS-SDR pulse blanking filter
00004  * \author Javier Arribas 2017
00005  *         Antonio Ramos 2017
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_PULSE_BLANKING_FILTER_H
00019 #define GNSS_SDR_PULSE_BLANKING_FILTER_H
00020
00021 #include "gnss_block_interface.h"
00022 #include "pulse_blanking_cc.h"
00023 #include <gnuradio/blocks/file_sink.h>
00024 #ifdef GR_GREATER_38
00025 #include <gnuradio/filter/freq_xlating_fir_filter.h>
00026 #else
00027 #include <gnuradio/filter/freq_xlating_fir_filter_ccf.h>
00028 #endif
00029 #include <string>
00030
00031 /** \addtogroup Input_Filter
00032  * \{ */
00033 /** \addtogroup Input_filter_adapters
00034  * \{ */
00035
00036
00037 class ConfigurationInterface;
00038
00039 class PulseBlankingFilter : public GNSSBlockInterface
00040 {
00041 public:
00042     PulseBlankingFilter(const ConfigurationInterface* configuration,
00043         std::string role, unsigned int in_streams,
00044         unsigned int out_streams);
00045
00046     ~PulseBlankingFilter() = default;
00047
00048     inline std::string role() override
00049     {
00050         return role_;
00051     }
00052
00053     /*! Returns "Pulse_Blanking_Filter"
00054     inline std::string implementation() override
00055     {
00056         return "Pulse_Blanking_Filter";
00057     }
00058
00059     inline size_t item_size() override
00060     {
00061         return input_size_;

```

```

00062     }
00063
00064     void connect(gr::top_block_sptr top_block) override;
00065     void disconnect(gr::top_block_sptr top_block) override;
00066     gr::basic_block_sptr get_left_block() override;
00067     gr::basic_block_sptr get_right_block() override;
00068
00069 private:
00070     pulse_blanking_cc_sptr pulse_blanking_cc_;
00071     gr::filter::freq_xlating_fir_filter_ccf::sptr freq_xlating_;
00072     gr::blocks::file_sink::sptr file_sink_;
00073     std::string dump_filename_;
00074     std::string item_type_;
00075     std::string role_;
00076     size_t input_size_;
00077     unsigned int in_streams_;
00078     unsigned int out_streams_;
00079     bool dump_;
00080     bool xlat_;
00081 };
00082
00083
00084 /** \} */
00085 /** \} */
00086 #endif // GNSS_SDR_PULSE_BLANKING_FILTER_H

```

## 11.125 beamformer.h File Reference

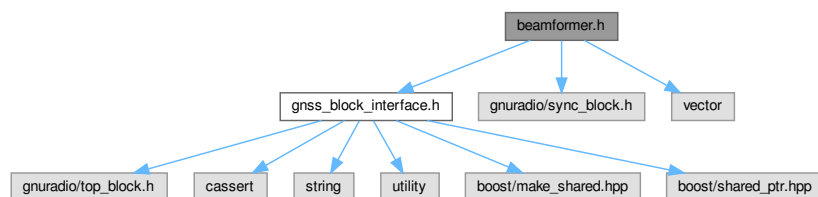
Simple spatial filter using RAW array input and beamforming coefficients.

```
#include "gnss_block_interface.h"
```

```
#include <gnuradio/sync_block.h>
```

```
#include <vector>
```

Include dependency graph for beamformer.h:



### Classes

- class [beamformer](#)

*This class implements a real-time software-defined spatial filter using the CTTC GNSS experimental antenna array input and a set of dynamically reloadable weights.*

### Typedefs

- using [beamformer\\_sptr](#) = [gnss\\_shared\\_ptr](#)<[beamformer](#)>

### Functions

- [beamformer\\_sptr](#) [make\\_beamformer\\_sptr](#) ()

### Variables

- const int [GNSS\\_SDR\\_BEAMFORMER\\_CHANNELS](#) = 8

### 11.125.1 Detailed Description

Simple spatial filter using RAW array input and beamforming coefficients.

**Author**

Javier Arribas jarribas (at) cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [beamformer.h](#).

---

**11.126 beamformer.h**

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file beamformer.h
00003  *
00004  * \brief Simple spatial filter using RAW array input and beamforming coefficients
00005  * \author Javier Arribas jarribas (at) cttc.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_BEAMFORMER_H
00019 #define GNSS_SDR_BEAMFORMER_H
00020
00021 #include "gnss_block_interface.h"
00022 #include <gnuradio/sync_block.h>
00023 #include <vector>
00024
00025 /** \addtogroup Input_Filter
00026  * \{ */
00027 /** \addtogroup Input_filter_gnuradio_blocks
00028  * \{ */
00029
00030
00031 class beamformer;
00032
00033 using beamformer_sptr = gnss_shared_ptr<beamformer>;
00034
00035 beamformer_sptr make_beamformer_sptr();
00036
00037 const int GNSS_SDR_BEAMFORMER_CHANNELS = 8;
00038
00039 /*!
00040  * \brief This class implements a real-time software-defined spatial filter using the CTTC GNSS
00041  * experimental antenna array input and a set of dynamically reloadable weights
00042  */
00042 class beamformer : public gr::sync_block
00043 {
00044 public:
00045     ~beamformer() = default;
00046     int work(int noutput_items, gr_vector_const_void_star &input_items,
00047             gr_vector_void_star &output_items);
00048 private:
00049     friend beamformer_sptr make_beamformer_sptr();
00050     beamformer();
00051     std::vector<gr_complex> weight_vector = std::vector<gr_complex>(GNSS_SDR_BEAMFORMER_CHANNELS,
00052                             gr_complex(1.0, 0.0));
00053 };
00054
00055
00056 /** \} */
00057 /** \} */
00058 #endif // GNSS_SDR_BEAMFORMER_H

```

**11.127 notch\_cc.h File Reference**

Implements a notch filter algorithm.

```

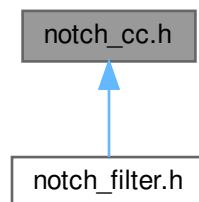
#include "gnss_block_interface.h"
#include "gnss_sdr_fft.h"
#include <gnuradio/block.h>

```

```
#include <volk_gnssdr/volk_gnssdr_alloc.h>
#include <stdint>
#include <memory>
Include dependency graph for notch_cc.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [Notch](#)

*This class implements a real-time software-defined multi state notch filter.*

## Typedefs

- using [notch\\_sptr](#) = gnss\_shared\_ptr<[Notch](#)>

## Functions

- notch\_sptr **make\_notch\_filter** (float pfa, float p\_c\_factor, int32\_t length, int32\_t n\_segments\_est, int32\_t n\_segments\_reset)

### 11.127.1 Detailed Description

Implements a notch filter algorithm.

#### Author

Antonio Ramos (antonio.ramosdet(at)gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [notch\\_cc.h](#).

---

## 11.128 notch\_cc.h

[Go to the documentation of this file.](#)

00001 /\*!

```

00002  * \file notch_cc.h
00003  * \brief Implements a notch filter algorithm
00004  * \author Antonio Ramos (antonio.ramosdet(at)gmail.com)
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  *
00016  */
00017
00018 #ifndef GNSS_SDR_NOTCH_CC_H
00019 #define GNSS_SDR_NOTCH_CC_H
00020
00021 #include "gnss_block_interface.h"
00022 #include "gnss_sdr_fft.h"
00023 #include <gnuradio/block.h>
00024 #include <volk_gnssssdr/volk_gnssssdr_alloc.h> // for volk_gnssssdr::vector
00025 #include <stdint>
00026 #include <memory>
00027
00028 /** \addtogroup Input_Filter
00029  * \{ */
00030 /** \addtogroup Input_filter_gnuradio_blocks
00031  * \{ */
00032
00033
00034 class Notch;
00035
00036 using notch_sptr = gnss_shared_ptr<Notch>;
00037
00038 notch_sptr make_notch_filter(
00039     float pfa,
00040     float p_c_factor,
00041     int32_t length,
00042     int32_t n_segments_est,
00043     int32_t n_segments_reset);
00044
00045 /*!
00046  * \brief This class implements a real-time software-defined multi state notch filter
00047  */
00048 class Notch : public gr::block
00049 {
00050 public:
00051     ~Notch() = default;
00052
00053     int general_work(int noutput_items, gr_vector_int &ninput_items,
00054         gr_vector_const_void_star &input_items,
00055         gr_vector_void_star &output_items);
00056
00057 private:
00058     friend notch_sptr make_notch_filter(float pfa, float p_c_factor, int32_t length, int32_t
n_segments_est, int32_t n_segments_reset);
00059     Notch(float pfa, float p_c_factor, int32_t length, int32_t n_segments_est, int32_t
n_segments_reset);
00060
00061     std::unique_ptr<gnss_fft_complex_fwd> d_fft_;
00062     volk_gnssssdr::vector<gr_complex> c_samples_;
00063     volk_gnssssdr::vector<float> angle_;
00064     volk_gnssssdr::vector<float> power_spect_;
00065     gr_complex last_out_;
00066     gr_complex z_0_;
00067     gr_complex p_c_factor_;
00068     float pfa_;
00069     float noise_pow_est_;
00070     float thres_;
00071     int32_t length_;
00072     int32_t n_deg_fred_;
00073     uint32_t n_segments_;
00074     uint32_t n_segments_est_;
00075     uint32_t n_segments_reset_;
00076     bool filter_state_;
00077 };
00078
00079
00080 /** \} */
00081 /** \} */
00082 #endif // GNSS_SDR_NOTCH_CC_H

```

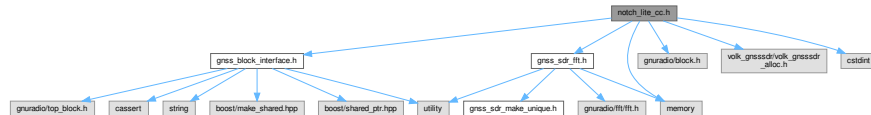


## 11.129 notch\_lite\_cc.h File Reference

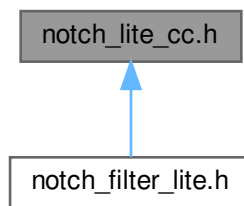
Implements a notch filter light algorithm.

```
#include "gnss_block_interface.h"
#include "gnss_sdr_fft.h"
#include <gnuradio/block.h>
#include <volk_gnssdr/volk_gnssdr_alloc.h>
#include <stdint>
#include <memory>
```

Include dependency graph for notch\_lite\_cc.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [NotchLite](#)

*This class implements a real-time software-defined multi state notch filter light version.*

### Typedefs

- using [notch\\_lite\\_sptr](#) = `gnss_shared_ptr<NotchLite>`

### Functions

- `notch_lite_sptr make_notch_filter_lite (float p_c_factor, float pfa, int32_t length, int32_t n_segments_est, int32_t n_segments_reset, int32_t n_segments_coeff)`

### 11.129.1 Detailed Description

Implements a notch filter light algorithm.

#### Author

Antonio Ramos ([antonio.ramosdet@gmail.com](mailto:antonio.ramosdet@gmail.com))

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [notch\\_lite\\_cc.h](#).

---

## 11.130 notch\_lite\_cc.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file notch_lite_cc.h
00003  * \brief Implements a notch filter light algorithm
00004  * \author Antonio Ramos (antonio.ramosdet(at)gmail.com)
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016  */
00017
00018  #ifndef GNSS_SDR_NOTCH_LITE_CC_H
00019  #define GNSS_SDR_NOTCH_LITE_CC_H
00020
00021  #include "gnss_block_interface.h"
00022  #include "gnss_sdr_fft.h"
00023  #include <gnuradio/block.h>
00024  #include <volk_gnssssdr/volk_gnssssdr_alloc.h> // for volk_gnssssdr::vector
00025  #include <cstdint>
00026  #include <memory>
00027
00028  /** \addtogroup Input_Filter
00029  * \{ */
00030  /** \addtogroup Input_filter_gnuradio_blocks
00031  * \{ */
00032
00033
00034  class NotchLite;
00035
00036  using notch_lite_sptr = gnss_shared_ptr<NotchLite>;
00037
00038  notch_lite_sptr make_notch_filter_lite(
00039      float p_c_factor,
00040      float pfa,
00041      int32_t length,
00042      int32_t n_segments_est,
00043      int32_t n_segments_reset,
00044      int32_t n_segments_coeff);
00045
00046  /*!
00047  * \brief This class implements a real-time software-defined multi state notch filter light version
00048  */
00049  class NotchLite : public gr::block
00050  {
00051  public:
00052      ~NotchLite() = default;
00053
00054      int general_work(int noutput_items, gr_vector_int &ninput_items,
00055                      gr_vector_const_void_star &input_items,
00056                      gr_vector_void_star &output_items);
00057
00058  private:
00059      friend notch_lite_sptr make_notch_filter_lite(float p_c_factor, float pfa, int32_t length, int32_t
n_segments_est, int32_t n_segments_reset, int32_t n_segments_coeff);
00060      NotchLite(float p_c_factor, float pfa, int32_t length, int32_t n_segments_est, int32_t
n_segments_reset, int32_t n_segments_coeff);
00061
00062      std::unique_ptr<gnss_fft_complex_fwd> d_fft_;
00063      volk_gnssssdr::vector<float> power_spect_;
00064      gr_complex last_out_;
00065      gr_complex z_0_;
00066      gr_complex p_c_factor_;
00067      gr_complex c_samples1_;
00068      gr_complex c_samples2_;
00069      float pfa_;
00070      float thres_;
00071      float noise_pow_est_;
00072      float angle1_;
00073      float angle2_;
00074      int32_t length_;
00075      int32_t n_segments_;
00076      int32_t n_segments_est_;
00077      int32_t n_segments_reset_;
00078      int32_t n_segments_coeff_reset_;
00079      int32_t n_segments_coeff_;
00080      int32_t n_deg_fred_;
00081      bool filter_state_;

```

```

00082 };
00083
00084
00085 /** \} */
00086 /** \} */
00087 #endif // GNSS_SDR_NOTCH_LITE_CC_H

```

## 11.131 pulse\_blanking\_cc.h File Reference

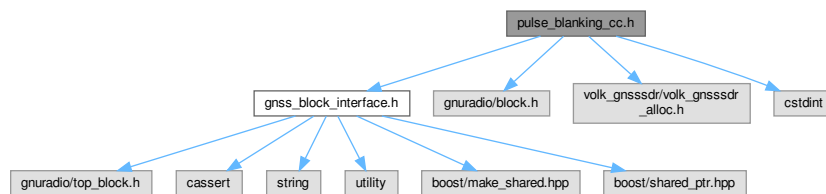
Implements a pulse blanking algorithm.

```

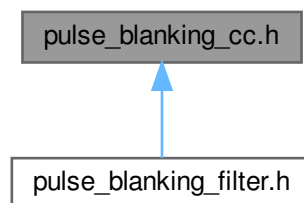
#include "gnss_block_interface.h"
#include <gnuradio/block.h>
#include <volk_gnssdr/volk_gnssdr_alloc.h>
#include <stdint>

```

Include dependency graph for pulse\_blanking\_cc.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [pulse\\_blanking\\_cc](#)

### Typedefs

- using [pulse\\_blanking\\_cc\\_sptr](#) = [gnss\\_shared\\_ptr](#)<[pulse\\_blanking\\_cc](#)>

### Functions

- [pulse\\_blanking\\_cc\\_sptr](#) **make\_pulse\_blanking\_cc** (float pfa, int32\_t length, int32\_t n\_segments\_est, int32\_t n\_segments\_reset)

### 11.131.1 Detailed Description

Implements a pulse blanking algorithm.

**Author**

Javier Arribas (jarribas(at)cttc.es) Antonio Ramos (antonio.ramosdet(at)gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [pulse\\_blanking\\_cc.h](#).

---

**11.132 pulse\_blanking\_cc.h**

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file pulse_blanking_cc.h
00003  * \brief Implements a pulse blanking algorithm
00004  * \author Javier Arribas (jarribas(at)cttc.es)
00005  *         Antonio Ramos (antonio.ramosdet(at)gmail.com)
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  *
00017  */
00018
00019 #ifndef GNSS_SDR_PULSE_BLANKING_CC_H
00020 #define GNSS_SDR_PULSE_BLANKING_CC_H
00021
00022 #include "gnss_block_interface.h"
00023 #include <gnuradio/block.h>
00024 #include <volk_gnssssdr/volk_gnssssdr_alloc.h> // for volk_gnssssdr::vector
00025 #include <cstdint>
00026
00027 /** \addtogroup Input_Filter
00028  * \{ */
00029 /** \addtogroup Input_filter_gnuradio_blocks input_filter_gr_blocks
00030  * GNU Radio blocks implementing input filters,
00031  * \{ */
00032
00033
00034 class pulse_blanking_cc;
00035
00036 using pulse_blanking_cc_sptr = gnss_shared_ptr<pulse_blanking_cc>;
00037
00038 pulse_blanking_cc_sptr make_pulse_blanking_cc(
00039     float pfa,
00040     int32_t length,
00041     int32_t n_segments_est,
00042     int32_t n_segments_reset);
00043
00044 class pulse_blanking_cc : public gr::block
00045 {
00046 public:
00047     ~pulse_blanking_cc() = default;
00048
00049     int general_work(int noutput_items __attribute__((unused)), gr_vector_int &ninput_items
00050         __attribute__((unused)),
00051         gr_vector_const_void_star &input_items, gr_vector_void_star &output_items);
00052 private:
00053     friend pulse_blanking_cc_sptr make_pulse_blanking_cc(float pfa, int32_t length, int32_t
00054         n_segments_est, int32_t n_segments_reset);
00055     pulse_blanking_cc(float pfa, int32_t length, int32_t n_segments_est, int32_t n_segments_reset);
00056     volk_gnssssdr::vector<gr_complex> zeros_;
00057     float noise_power_estimation_;
00058     float thres_;
00059     float pfa_;
00060     int32_t length_;
00061     int32_t n_segments_;
00062     int32_t n_segments_est_;
00063     int32_t n_segments_reset_;
00064     int32_t n_deg_fred_;
00065     bool last_filtered_;
00066 };
00067
00068 /** \} */
00069 /** \} */
00070 #endif // GNSS_SDR_PULSE_BLANKING_CC_H

```

## 11.133 beidou\_b1i\_signal\_replica.h File Reference

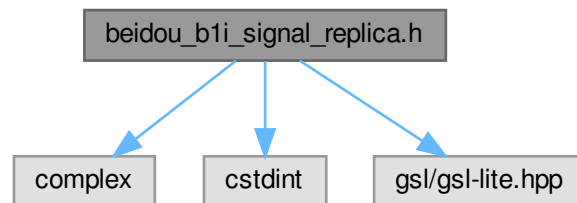
This file implements various functions for BeiDou B1I signal replica generation.

```
#include <complex>
```

```
#include <cstdint>
```

```
#include <gsl/gsl-lite.hpp>
```

Include dependency graph for beidou\_b1i\_signal\_replica.h:



### Functions

- void [beidou\\_b1i\\_code\\_gen\\_int](#) (own::span< int32\_t > dest, int32\_t prn, uint32\_t chip\_shift)  
*Generates int32\_t GPS L1 C/A code for the desired SV ID and code shift.*
- void [beidou\\_b1i\\_code\\_gen\\_float](#) (own::span< float > dest, int32\_t prn, uint32\_t chip\_shift)  
*Generates float GPS L1 C/A code for the desired SV ID and code shift.*
- void [beidou\\_b1i\\_code\\_gen\\_complex](#) (own::span< std::complex< float > > dest, int32\_t prn, uint32\_t chip\_shift)  
*Generates complex GPS L1 C/A code for the desired SV ID and code shift.*
- void [beidou\\_b1i\\_code\\_gen\\_complex\\_sampled](#) (own::span< std::complex< float > > dest, uint32\_t prn, int32\_t sampling\_freq, uint32\_t chip\_shift)  
*Generates complex GPS L1 C/A code for the desired SV ID and code shift, and sampled to specific sampling frequency.*

### 11.133.1 Detailed Description

This file implements various functions for BeiDou B1I signal replica generation.

#### Author

Sergi Segura, 2018. [sergi.segura.munoz\(at\)gmail.com](mailto:sergi.segura.munoz(at)gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [beidou\\_b1i\\_signal\\_replica.h](#).

---

## 11.134 beidou\_b1i\_signal\_replica.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file beidou_b1i_signal_replica.h
00003  * \brief This file implements various functions for BeiDou B1I signal replica
00004  * generation
00005  * \author Sergi Segura, 2018. sergi.segura.munoz\(at\)gmail.com
00006  *
00007  *
00008  * -----
00009  *

```

```

00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019 #ifndef GNSS_SDR_BEIDOU_B1I_SIGNAL_REPLICA_H
00020 #define GNSS_SDR_BEIDOU_B1I_SIGNAL_REPLICA_H
00021
00022 #include <complex>
00023 #include <stdint>
00024 #if HAS_STD_SPAN
00025 #include <span>
00026 namespace own = std;
00027 #else
00028 #include <gsl/gsl-lite.hpp>
00029 namespace own = gsl;
00030 #endif
00031
00032 /** \addtogroup Algorithms_Library Algorithms Common Library
00033  * Common utilities for the GNSS receiver.
00034  * @{ */
00035 /** \addtogroup Algorithm_libs algorithms_libs
00036  * Common utilities for GNSS algorithms.
00037  * @{ */
00038
00039
00040 //! Generates int32_t GPS L1 C/A code for the desired SV ID and code shift
00041 void beidou_b1i_code_gen_int(own::span<int32_t> dest, int32_t prn, uint32_t chip_shift);
00042
00043 //! Generates float GPS L1 C/A code for the desired SV ID and code shift
00044 void beidou_b1i_code_gen_float(own::span<float> dest, int32_t prn, uint32_t chip_shift);
00045
00046 //! Generates complex GPS L1 C/A code for the desired SV ID and code shift
00047 void beidou_b1i_code_gen_complex(own::span<std::complex<float>> dest, int32_t prn, uint32_t
chip_shift);
00048
00049 //! Generates complex GPS L1 C/A code for the desired SV ID and code shift, and sampled to specific
sampling frequency
00050 void beidou_b1i_code_gen_complex_sampled(own::span<std::complex<float>> dest, uint32_t prn, int32_t
sampling_freq, uint32_t chip_shift);
00051
00052
00053 /** @} */
00054 /** @} */
00055 #endif // GNSS_SDR_BEIDOU_B1I_SIGNAL_REPLICA_H

```

## 11.135 beidou\_b3i\_signal\_replica.h File Reference

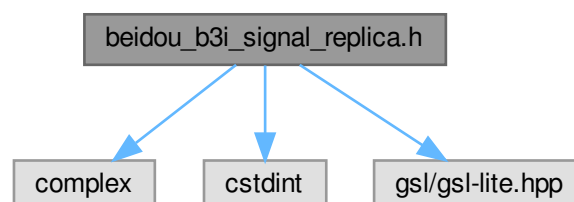
This file implements various functions for BeiDou B3I signal replica generation.

```
#include <complex>
```

```
#include <stdint>
```

```
#include <gsl/gsl-lite.hpp>
```

Include dependency graph for beidou\_b3i\_signal\_replica.h:



## Functions

- void `beidou_b3i_code_gen_int` (own::span< int > dest, int32\_t prn, uint32\_t chip\_shift)  
*Generates int BeiDou B3I code for the desired SV ID and code shift.*
- void `beidou_b3i_code_gen_float` (own::span< float > dest, int32\_t prn, uint32\_t chip\_shift)  
*Generates float BeiDou B3I code for the desired SV ID and code shift.*
- void `beidou_b3i_code_gen_complex` (own::span< std::complex< float > > dest, int32\_t prn, uint32\_t chip\_shift)  
*Generates complex BeiDou B3I code for the desired SV ID and code shift.*
- void `beidou_b3i_code_gen_complex_sampled` (own::span< std::complex< float > > dest, uint32\_t prn, int sampling\_freq, uint32\_t chip\_shift)  
*Generates complex BeiDou B3I code for the desired SV ID and code shift, and sampled to specific sampling frequency.*

### 11.135.1 Detailed Description

This file implements various functions for BeiDou B3I signal replica generation.

#### Author

Damian Miralles, 2019. [dmiralles2009@gmail.com](mailto:dmiralles2009@gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file `beidou_b3i_signal_replica.h`.

---

## 11.136 beidou\_b3i\_signal\_replica.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file beidou_b3i_signal_replica.h
00003  * \brief This file implements various functions for BeiDou B3I signal replica
00004  * generation
00005  * \author Damian Miralles, 2019. dmiralles2009@gmail.com
00006  *
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019 #ifndef GNSS_SDR_BEIDOU_B3I_SIGNAL_REPLICA_H
00020 #define GNSS_SDR_BEIDOU_B3I_SIGNAL_REPLICA_H
00021
00022 #include <complex>
00023 #include <cstdint>
00024 #if HAS_STD_SPAN
00025 #include <span>
00026 namespace own = std;
00027 #else
00028 #include <gsl/gsl-lite.hpp>
00029 namespace own = gsl;
00030 #endif
00031
00032 /** \addtogroup Algorithms_Library
00033  * \{ */
00034 /** \addtogroup Algorithm_libs algorithms_libs
00035  * \{ */
00036
00037
00038 /*! Generates int BeiDou B3I code for the desired SV ID and code shift
00039 void beidou_b3i_code_gen_int(own::span<int> dest, int32_t prn, uint32_t chip_shift);
00040
00041 /*! Generates float BeiDou B3I code for the desired SV ID and code shift
00042 void beidou_b3i_code_gen_float(own::span<float> dest, int32_t prn, uint32_t chip_shift);
00043
00044 /*! Generates complex BeiDou B3I code for the desired SV ID and code shift
```

```

00045 void beidou_b3i_code_gen_complex(own::span<std::complex<float>> dest, int32_t prn, uint32_t
      chip_shift);
00046
00047 //! Generates complex BeiDou B3I code for the desired SV ID and code shift, and sampled to specific
      sampling frequency
00048 void beidou_b3i_code_gen_complex_sampled(own::span<std::complex<float>> dest, uint32_t prn, int
      sampling_freq, uint32_t chip_shift);
00049
00050
00051 /** \} */
00052 /** \} */
00053 #endif // GNSS_SDR_BEIDOU_B3I_SIGNAL_REPLICA_H

```

## 11.137 byte\_x2\_to\_complex\_byte.h File Reference

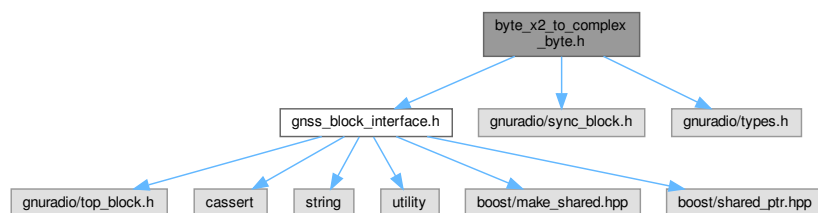
Adapts two signed char streams into a `std::complex<signed char>` stream.

```
#include "gnss_block_interface.h"
```

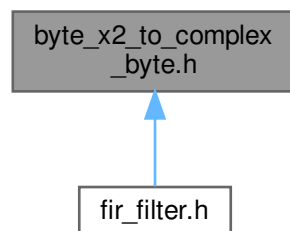
```
#include <gnuradio/sync_block.h>
```

```
#include <gnuradio/types.h>
```

Include dependency graph for `byte_x2_to_complex_byte.h`:



This graph shows which files directly or indirectly include this file:



### Classes

- class [byte\\_x2\\_to\\_complex\\_byte](#)

*This class adapts two signed char streams into a `std::complex<signed char>` stream.*

### Typedefs

- using [byte\\_x2\\_to\\_complex\\_byte\\_sptr](#) = `gnss_shared_ptr<byte\_x2\_to\_complex\_byte>`



## Functions

- `byte_x2_to_complex_byte_sptr make_byte_x2_to_complex_byte ()`

### 11.137.1 Detailed Description

Adapts two signed char streams into a `std::complex<signed char>` stream.

#### Author

Carles Fernandez Prades, [cfernandez\(at\)cttc.es](mailto:cfernandez(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [byte\\_x2\\_to\\_complex\\_byte.h](#).

---

## 11.138 byte\_x2\_to\_complex\_byte.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file byte_x2_to_complex_byte.h
00003  * \brief Adapts two signed char streams into a std::complex<signed char> stream
00004  * \author Carles Fernandez Prades, cfernandez(at)cttc.es
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_BYTE_X2_TO_COMPLEX_BYTE_H
00019 #define GNSS_SDR_BYTE_X2_TO_COMPLEX_BYTE_H
00020
00021 #include "gnss_block_interface.h"
00022 #include <gnuradio/sync_block.h>
00023 #include <gnuradio/types.h> // for gr_vector_const_void_star
00024
00025 /** \addtogroup Algorithms_Library
00026  * \{ */
00027 /** \addtogroup Algorithm_libs algorithms_libs
00028  * \{ */
00029
00030
00031 class byte_x2_to_complex_byte;
00032
00033 using byte_x2_to_complex_byte_sptr = gnss_shared_ptr<byte_x2_to_complex_byte>;
00034
00035 byte_x2_to_complex_byte_sptr make_byte_x2_to_complex_byte();
00036
00037 /*!
00038  * \brief This class adapts two signed char streams
00039  * into a std::complex<signed char> stream
00040  */
00041 class byte_x2_to_complex_byte : public gr::sync_block
00042 {
00043 public:
00044     int work(int noutput_items,
00045             gr_vector_const_void_star &input_items,
00046             gr_vector_void_star &output_items);
00047
00048 private:
00049     friend byte_x2_to_complex_byte_sptr make_byte_x2_to_complex_byte();
00050     byte_x2_to_complex_byte();
00051 };
00052
00053
00054 /** \} */
00055 /** \} */
00056 #endif // GNSS_SDR_BYTE_X2_TO_COMPLEX_BYTE_Hs

```



```

00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_COMPLEX_BYTE_TO_FLOAT_X2_H
00019 #define GNSS_SDR_COMPLEX_BYTE_TO_FLOAT_X2_H
00020
00021 #include "gnss_block_interface.h"
00022 #include <gnuradio/sync_block.h>
00023 #include <gnuradio/types.h> // for gr_vector_const_void_star
00024
00025 /** \addtogroup Algorithms_Library
00026  * \{ */
00027 /** \addtogroup Algorithm_libs algorithms_libs
00028  * \{ */
00029
00030
00031 class complex_byte_to_float_x2;
00032
00033 using complex_byte_to_float_x2_sptr = gnss_shared_ptr<complex_byte_to_float_x2>;
00034
00035 complex_byte_to_float_x2_sptr make_complex_byte_to_float_x2();
00036
00037 /*!
00038  * \brief This class adapts a std::complex<signed char> stream
00039  * into two 16-bits (short) streams
00040  */
00041 class complex_byte_to_float_x2 : public gr::sync_block
00042 {
00043 public:
00044     int work(int noutput_items,
00045             gr_vector_const_void_star &input_items,
00046             gr_vector_void_star &output_items);
00047 private:
00048     friend complex_byte_to_float_x2_sptr make_complex_byte_to_float_x2();
00049     complex_byte_to_float_x2();
00050 };
00051
00052
00053
00054 /** \} */
00055 /** \} */
00056 #endif // GNSS_SDR_COMPLEX_BYTE_TO_FLOAT_X2_H

```

## 11.141 complex\_float\_to\_complex\_byte.h File Reference

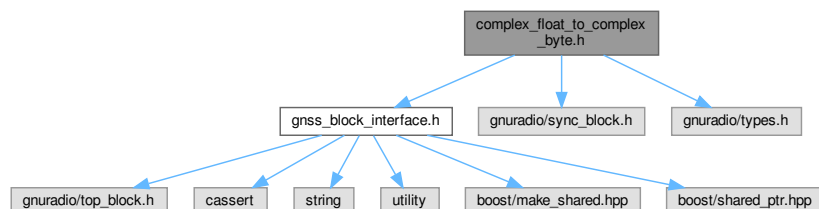
Adapts a gr\_complex stream into a std::complex<signed char> stream.

```
#include "gnss_block_interface.h"
```

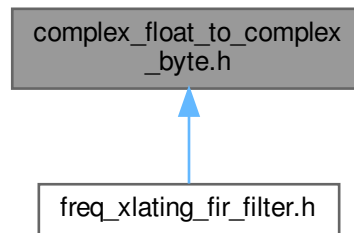
```
#include <gnuradio/sync_block.h>
```

```
#include <gnuradio/types.h>
```

Include dependency graph for complex\_float\_to\_complex\_byte.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [complex\\_float\\_to\\_complex\\_byte](#)

*This class adapts a gr\_complex stream into a std::complex<signed char> stream.*

## Typedefs

- using [complex\\_float\\_to\\_complex\\_byte\\_sptr](#) = gnss\_shared\_ptr<[complex\\_float\\_to\\_complex\\_byte](#)>

## Functions

- [complex\\_float\\_to\\_complex\\_byte\\_sptr](#) **make\_complex\_float\_to\_complex\_byte** ()

### 11.141.1 Detailed Description

Adapts a gr\_complex stream into a std::complex<signed char> stream.

#### Author

Carles Fernandez Prades, cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [complex\\_float\\_to\\_complex\\_byte.h](#).

---

## 11.142 complex\_float\_to\_complex\_byte.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file complex_float_to_complex_byte.h
00003  * \brief Adapts a gr_complex stream into a std::complex<signed char> stream
00004  * \author Carles Fernandez Prades, cfernandez(at)cttc.es
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_COMPLEX_FLOAT_TO_COMPLEX_BYTE_H
00019 #define GNSS_SDR_COMPLEX_FLOAT_TO_COMPLEX_BYTE_H
00020
  
```

```

00021 #include "gnss_block_interface.h"
00022 #include <gnuradio/sync_block.h>
00023 #include <gnuradio/types.h> // for gr_vector_const_void_star
00024
00025 /** \addtogroup Algorithms_Library
00026 * \{ */
00027 /** \addtogroup Algorithm_libs algorithms_libs
00028 * \{ */
00029
00030
00031 class complex_float_to_complex_byte;
00032
00033 using complex_float_to_complex_byte_sptr = gnss_shared_ptr<complex_float_to_complex_byte>;
00034
00035 complex_float_to_complex_byte_sptr make_complex_float_to_complex_byte();
00036
00037 /*!
00038 * \brief This class adapts a gr_complex stream into a std::complex<signed char> stream
00039 */
00040 class complex_float_to_complex_byte : public gr::sync_block
00041 {
00042 public:
00043     int work(int noutput_items,
00044             gr_vector_const_void_star &input_items,
00045             gr_vector_void_star &output_items);
00046 private:
00047     friend complex_float_to_complex_byte_sptr make_complex_float_to_complex_byte();
00048     complex_float_to_complex_byte();
00049 };
00050
00051
00052
00053 /** \} */
00054 /** \} */
00055 #endif // GNSS_SDR_COMPLEX_FLOAT_TO_COMPLEX_BYTE_H

```

## 11.143 conjugate\_cc.h File Reference

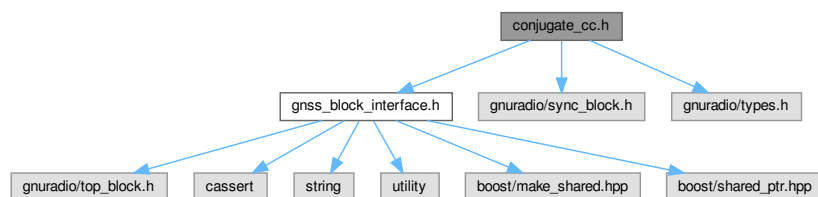
Conjugate a stream of gr\_complex.

```
#include "gnss_block_interface.h"
```

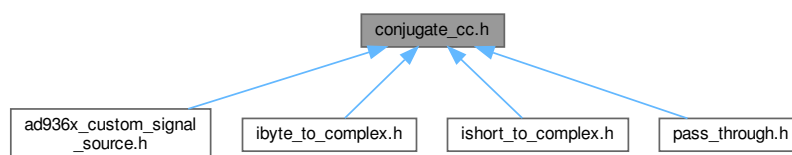
```
#include <gnuradio/sync_block.h>
```

```
#include <gnuradio/types.h>
```

Include dependency graph for conjugate\_cc.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [conjugate\\_cc](#)

*This class adapts a `std::complex<short>` stream into two 32-bits (float) streams.*

## Typedefs

- using [conjugate\\_cc\\_sptr](#) = `gnss_shared_ptr<conjugate\_cc>`

## Functions

- `conjugate_cc_sptr make_conjugate_cc ()`

### 11.143.1 Detailed Description

Conjugate a stream of `gr_complex`.

#### Author

Carles Fernandez Prades, [cfernandez\(at\)cttc.es](mailto:cfernandez(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [conjugate\\_cc.h](#).

---

## 11.144 conjugate\_cc.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file conjugate_cc.h
00003  * \brief Conjugate a stream of gr_complex
00004  * \author Carles Fernandez Prades, cfernandez(at)cttc.es
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_CONJUGATE_CC_H
00019 #define GNSS_SDR_CONJUGATE_CC_H
00020
00021 #include "gnss_block_interface.h"
00022 #include <gnuradio/sync_block.h>
00023 #include <gnuradio/types.h> // for gr_vector_const_void_star
00024
00025 /** \addtogroup Algorithms_Library
00026  * \{ */
00027 /** \addtogroup Algorithm_libs algorithms_libs
00028  * \{ */
00029
00030
00031 class conjugate_cc;
00032
00033 using conjugate_cc_sptr = gnss_shared_ptr<conjugate_cc>;
00034
00035 conjugate_cc_sptr make_conjugate_cc();
00036
00037 /*!
00038 * \brief This class adapts a std::complex<short> stream
00039 * into two 32-bits (float) streams
00040 */
00041 class conjugate_cc : public gr::sync_block
00042 {
00043 public:
00044     int work(int noutput_items,
00045             gr_vector_const_void_star &input_items,
00046             gr_vector_void_star &output_items);
00047

```

```

00048 private:
00049     friend conjugate_cc_sptr make_conjugate_cc();
00050     conjugate_cc();
00051 };
00052
00053
00054 /** \} */
00055 /** \} */
00056 #endif // GNSS_SDR_CONJUGATE_CC_H

```

## 11.145 conjugate\_ic.h File Reference

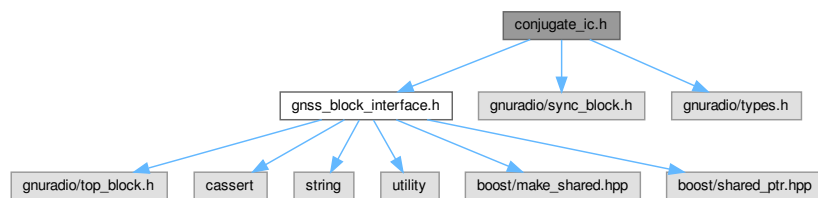
Conjugate a stream of `lv_8sc_t ( std::complex<char> )`

```
#include "gnss_block_interface.h"
```

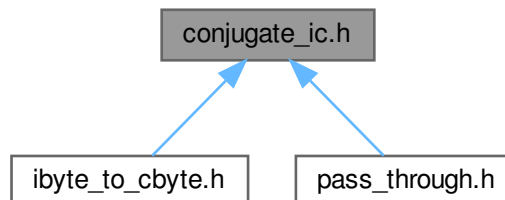
```
#include <gnuradio/sync_block.h>
```

```
#include <gnuradio/types.h>
```

Include dependency graph for conjugate\_ic.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [conjugate\\_ic](#)

*This class adapts a `std::complex<short>` stream into two 32-bits (float) streams.*

### Typedefs

- using [conjugate\\_ic\\_sptr](#) = `gnss_shared_ptr<conjugate_ic>`

### Functions

- `conjugate_ic_sptr make_conjugate_ic ()`

### 11.145.1 Detailed Description

Conjugate a stream of `lv_8sc_t ( std::complex<char> )`

Author

Carles Fernandez Prades, [cfernandez\(at\)cttc.es](mailto:cfernandez(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [conjugate\\_ic.h](#).

---

## 11.146 conjugate\_ic.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file conjugate_ic.h
00003  * \brief Conjugate a stream of lv_8sc_t ( std::complex<char> )
00004  * \author Carles Fernandez Prades, cfernandez(at)cttc.es
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_CONJUGATE_IC_H
00019 #define GNSS_SDR_CONJUGATE_IC_H
00020
00021 #include "gnss_block_interface.h"
00022 #include <gnuradio/sync_block.h>
00023 #include <gnuradio/types.h> // for gr_vector_const_void_star
00024
00025 /** \addtogroup Algorithms_Library
00026  * \{ */
00027 /** \addtogroup Algorithm_libs algorithms_libs
00028  * \{ */
00029
00030
00031 class conjugate_ic;
00032
00033 using conjugate_ic_sptr = gnss_shared_ptr<conjugate_ic>;
00034
00035 conjugate_ic_sptr make_conjugate_ic();
00036
00037 /*!
00038  * \brief This class adapts a std::complex<short> stream
00039  * into two 32-bits (float) streams
00040  */
00041 class conjugate_ic : public gr::sync_block
00042 {
00043 public:
00044     int work(int noutput_items,
00045             gr_vector_const_void_star &input_items,
00046             gr_vector_void_star &output_items);
00047
00048 private:
00049     friend conjugate_ic_sptr make_conjugate_ic();
00050     conjugate_ic();
00051 };
00052
00053
00054 /** \} */
00055 /** \} */
00056 #endif // GNSS_SDR_CONJUGATE_IC_H
```

## 11.147 conjugate\_sc.h File Reference

Conjugate a stream of `lv_16sc_t ( std::complex<short> )`

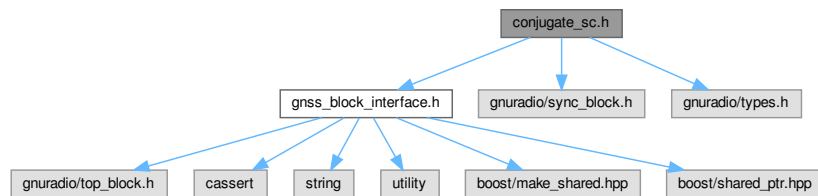
`#include "gnss_block_interface.h"`

`#include <gnuradio/sync_block.h>`

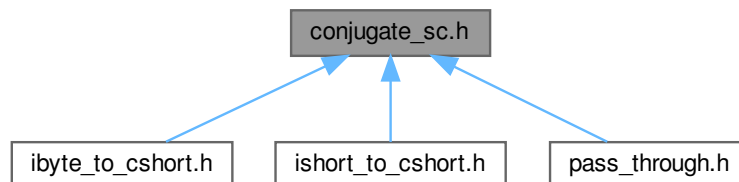


```
#include <gnuradio/types.h>
```

Include dependency graph for conjugate\_sc.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [conjugate\\_sc](#)

*This class adapts a `std::complex<short>` stream into two 32-bits (float) streams.*

## Typedefs

- using [conjugate\\_sc\\_sptr](#) = `gnss_shared_ptr<conjugate\_sc>`

## Functions

- `conjugate_sc_sptr make_conjugate_sc ()`

### 11.147.1 Detailed Description

Conjugate a stream of `lv_16sc_t` ( `std::complex<short>` )

#### Author

Carles Fernandez Prades, [cfernandez\(at\)cttc.es](mailto:cfernandez(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [conjugate\\_sc.h](#).

---

## 11.148 conjugate\_sc.h

[Go to the documentation of this file.](#)

```
00001 /*!
```

```

00002  * \file conjugate_sc.h
00003  * \brief Conjugate a stream of lv_16sc_t ( std::complex<short> )
00004  * \author Carles Fernandez Prades, cfernandez(at)cttc.es
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_CONJUGATE_SC_H
00019 #define GNSS_SDR_CONJUGATE_SC_H
00020
00021 #include "gnss_block_interface.h"
00022 #include <gnuradio/sync_block.h>
00023 #include <gnuradio/types.h> // for gr_vector_const_void_star
00024
00025 /** \addtogroup Algorithms_Library
00026  * \{ */
00027 /** \addtogroup Algorithm_libs algorithms_libs
00028  * \{ */
00029
00030
00031 class conjugate_sc;
00032
00033 using conjugate_sc_sptr = gnss_shared_ptr<conjugate_sc>;
00034
00035 conjugate_sc_sptr make_conjugate_sc();
00036
00037 /*!
00038  * \brief This class adapts a std::complex<short> stream
00039  * into two 32-bits (float) streams
00040  */
00041 class conjugate_sc : public gr::sync_block
00042 {
00043 public:
00044     int work(int noutput_items,
00045             gr_vector_const_void_star &input_items,
00046             gr_vector_void_star &output_items);
00047 private:
00048     friend conjugate_sc_sptr make_conjugate_sc();
00049     conjugate_sc();
00050 };
00051
00052
00053
00054 /** \} */
00055 /** \} */
00056 #endif // GNSS_SDR_CONJUGATE_SC_H

```

## 11.149 cshort\_to\_float\_x2.h File Reference

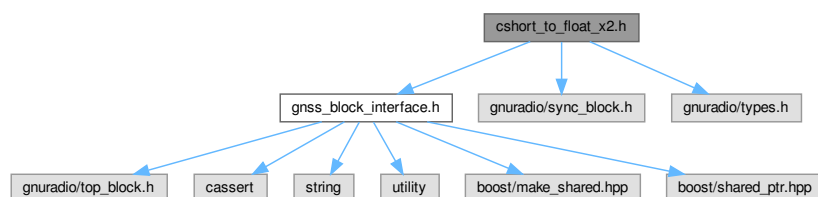
Adapts a `std::complex<short>` stream into two float streams.

```
#include "gnss_block_interface.h"
```

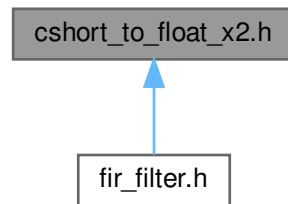
```
#include <gnuradio/sync_block.h>
```

```
#include <gnuradio/types.h>
```

Include dependency graph for `csort_to_float_x2.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [cshort\\_to\\_float\\_x2](#)

*This class adapts a `std::complex<short>` stream into two 32-bits (float) streams.*

## Typedefs

- using [cshort\\_to\\_float\\_x2\\_sptr](#) = `gnss_shared_ptr<cshort_to_float_x2>`

## Functions

- `cshort_to_float_x2_sptr make_cshort_to_float_x2 ()`

### 11.149.1 Detailed Description

Adapts a `std::complex<short>` stream into two float streams.

#### Author

Carles Fernandez Prades, [cfernandez\(at\)cttc.es](mailto:cfernandez(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [cshort\\_to\\_float\\_x2.h](#).

---

## 11.150 cshort\_to\_float\_x2.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file cshort_to_float_x2.h
00003  * \brief Adapts a std::complex<short> stream into two float streams
00004  * \author Carles Fernandez Prades, cfernandez(at)cttc.es
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_CSHORT_TO_FLOAT_X2_H
00019 #define GNSS_SDR_CSHORT_TO_FLOAT_X2_H
00020
00021 #include "gnss_block_interface.h"
  
```

```

00022 #include <gnuradio/sync_block.h>
00023 #include <gnuradio/types.h> // for gr_vector_const_void_star
00024
00025 /** \addtogroup Algorithms_Library
00026 * \{ */
00027 /** \addtogroup Algorithm_libs algorithms_libs
00028 * \{ */
00029
00030
00031 class cshort_to_float_x2;
00032
00033 using cshort_to_float_x2_sptr = gnss_shared_ptr<csshort_to_float_x2>;
00034
00035 cshort_to_float_x2_sptr make_cshort_to_float_x2();
00036
00037 /*!
00038 * \brief This class adapts a std::complex<short> stream
00039 * into two 32-bits (float) streams
00040 */
00041 class cshort_to_float_x2 : public gr::sync_block
00042 {
00043 public:
00044     int work(int noutput_items,
00045             gr_vector_const_void_star &input_items,
00046             gr_vector_void_star &output_items);
00047 private:
00048     friend cshort_to_float_x2_sptr make_cshort_to_float_x2();
00049     cshort_to_float_x2();
00050 };
00051
00052
00053
00054 /** \} */
00055 /** \} */
00056 #endif // GNSS_SDR_CSHORT_TO_FLOAT_X2_H

```

## 11.151 galileo\_e1\_signal\_replica.h File Reference

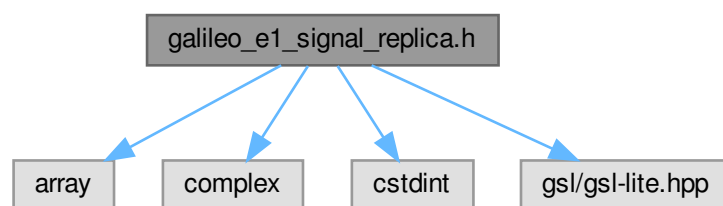
This library implements various functions for Galileo E1 signal replica generation.

```

#include <array>
#include <complex>
#include <cstdint>
#include <gsl/gsl-lite.hpp>

```

Include dependency graph for galileo\_e1\_signal\_replica.h:



### Functions

- void [galileo\\_e1\\_code\\_gen\\_sinboc11\\_float](#) (own::span< float > dest, const std::array< char, 3 > &signal\_id, uint32\_t prn)

*This function generates Galileo E1 code (can select E1B or E1C sinboc).*

- void [galileo\\_e1\\_code\\_gen\\_float\\_sampled](#) (own::span< float > dest, const std::array< char, 3 > &signal\_id, bool cboc, uint32\_t prn, int32\_t sampling\_freq, uint32\_t chip\_shift, bool secondary\_flag)

*This function generates Galileo E1 code (can select E1B or E1C, cboc or sinboc and the sample frequency sampling←\_freq).*

- void [galileo\\_e1\\_code\\_gen\\_float\\_sampled](#) (own::span< float > dest, const std::array< char, 3 > &signal\_id, bool cboc, uint32\_t prn, int32\_t sampling\_freq, uint32\_t chip\_shift)  
*This function generates Galileo E1 code (can select E1B or E1C, cboc or sinboc and the sample frequency sampling←\_freq).*
- void [galileo\\_e1\\_code\\_gen\\_complex\\_sampled](#) (own::span< std::complex< float > > dest, const std::array< char, 3 > &signal\_id, bool cboc, uint32\_t prn, int32\_t sampling\_freq, uint32\_t chip\_shift, bool secondary\_flag)  
*This function generates Galileo E1 code (can select E1B or E1C, cboc or sinboc and the sample frequency sampling←\_freq).*
- void [galileo\\_e1\\_code\\_gen\\_complex\\_sampled](#) (own::span< std::complex< float > > dest, const std::array< char, 3 > &signal\_id, bool cboc, uint32\_t prn, int32\_t sampling\_freq, uint32\_t chip\_shift)  
*galileo\_e1\_code\_gen\_complex\_sampled without secondary\_flag for backward compatibility.*

### 11.151.1 Detailed Description

This library implements various functions for Galileo E1 signal replica generation.

#### Author

Luis Esteve, 2012. luis(at)epsilon-formacion.com

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [galileo\\_e1\\_signal\\_replica.h](#).

---

## 11.152 galileo\_e1\_signal\_replica.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file galileo_e1_signal_replica.h
00003  * \brief This library implements various functions for Galileo E1 signal
00004  * replica generation
00005  * \author Luis Esteve, 2012. luis(at)epsilon-formacion.com
00006  *
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019 #ifndef GNSS_SDR_GALILEO_E1_SIGNAL_REPLICA_H
00020 #define GNSS_SDR_GALILEO_E1_SIGNAL_REPLICA_H
00021
00022 #include <array>
00023 #include <complex>
00024 #include <cstdint>
00025 #if HAS_STD_SPAN
00026 #include <span>
00027 namespace own = std;
00028 #else
00029 #include <gsl/gsl-lite.hpp>
00030 namespace own = gsl;
00031 #endif
00032
00033 /** \addtogroup Algorithms_Library
00034  * \{ */
00035 /** \addtogroup Algorithm_libs algorithms_libs
00036  * \{ */
00037
00038
00039 /*!
00040  * \brief This function generates Galileo E1 code (can select E1B or E1C sinboc).
00041  *
00042  */
00043 void galileo_e1_code_gen_sinboc11_float(own::span<float> dest, const std::array<char, 3>& signal_id,
uint32_t prn);
00044
00045 /*!
00046  * \brief This function generates Galileo E1 code (can select E1B or E1C, cboc or sinboc
```

```

00047 * and the sample frequency sampling_freq).
00048 *
00049 */
00050 void galileo_e1_code_gen_float_sampled(own::span<float> dest, const std::array<char, 3>& signal_id,
00051     bool cboc, uint32_t prn, int32_t sampling_freq, uint32_t chip_shift,
00052     bool secondary_flag);
00053
00054 /*!
00055 * \brief This function generates Galileo E1 code (can select E1B or E1C, cboc or sinboc
00056 * and the sample frequency sampling_freq).
00057 *
00058 */
00059 void galileo_e1_code_gen_float_sampled(own::span<float> dest, const std::array<char, 3>& signal_id,
00060     bool cboc, uint32_t prn, int32_t sampling_freq, uint32_t chip_shift);
00061
00062 /*!
00063 * \brief This function generates Galileo E1 code (can select E1B or E1C, cboc or sinboc
00064 * and the sample frequency sampling_freq).
00065 *
00066 */
00067 void galileo_e1_code_gen_complex_sampled(own::span<std::complex<float>> dest, const std::array<char,
00068     3>& signal_id,
00069     bool cboc, uint32_t prn, int32_t sampling_freq, uint32_t chip_shift,
00070     bool secondary_flag);
00071
00072 /*!
00073 * \brief galileo_e1_code_gen_complex_sampled without secondary_flag for backward compatibility.
00074 */
00074 void galileo_e1_code_gen_complex_sampled(own::span<std::complex<float>> dest, const std::array<char,
00075     3>& signal_id,
00076     bool cboc, uint32_t prn, int32_t sampling_freq, uint32_t chip_shift);
00077
00078 /** \} */
00079 /** \} */
00080 #endif // GNSS_SDR_GALILEO_E1_SIGNAL_REPLICA_H

```

## 11.153 galileo\_e5\_signal\_replica.h File Reference

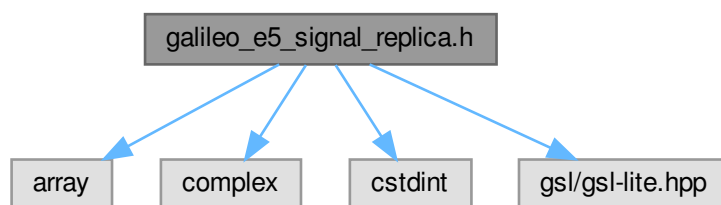
This library implements various functions for Galileo E5 signal replica generation.

```

#include <array>
#include <complex>
#include <cstdint>
#include <gsl/gsl-lite.hpp>

```

Include dependency graph for galileo\_e5\_signal\_replica.h:



### Functions

- void [galileo\\_e5\\_a\\_code\\_gen\\_complex\\_primary](#) (own::span< std::complex< float > > dest, int32\_t prn, const std::array< char, 3 > &signal\_id)  
*Generates Galileo E5a code at 1 sample/chip.*
- void [galileo\\_e5\\_a\\_code\\_gen\\_complex\\_sampled](#) (own::span< std::complex< float > > dest, uint32\_t prn, const std::array< char, 3 > &signal\_id, int32\_t sampling\_freq, uint32\_t chip\_shift)  
*Generates Galileo E5a complex code, shifted to the desired chip and sampled at a frequency sampling\_freq.*

- void [galileo\\_e5\\_b\\_code\\_gen\\_complex\\_primary](#) (own::span< std::complex< float > > dest, int32\_t prn, const std::array< char, 3 > &signal\_id)  
Generates Galileo E5b code at 1 sample/chip.
- void [galileo\\_e5\\_b\\_code\\_gen\\_complex\\_sampled](#) (own::span< std::complex< float > > dest, uint32\_t prn, const std::array< char, 3 > &signal\_id, int32\_t sampling\_freq, uint32\_t chip\_shift)  
Generates Galileo E5b complex code, shifted to the desired chip and sampled at a frequency sampling\_freq.

### 11.153.1 Detailed Description

This library implements various functions for Galileo E5 signal replica generation.

#### Author

Marc Sales, 2014. [marcsales92\(at\)gmail.com](mailto:marcsales92(at)gmail.com)

Piyush Gupta, 2020. [piyush04111999@gmail.com](mailto:piyush04111999@gmail.com)

#### Note

Code added as part of GSoC 2020 Program.

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [galileo\\_e5\\_signal\\_replica.h](#).

---

## 11.154 galileo\_e5\_signal\_replica.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file galileo_e5_signal_replica.h
00003  * \brief This library implements various functions for Galileo E5 signal
00004  * replica generation
00005  * \author Marc Sales, 2014. marcsales92(at)gmail.com
00006  * \author Piyush Gupta, 2020. piyush04111999@gmail.com
00007  * \note Code added as part of GSoC 2020 Program.
00008  *
00009  *
00010  * -----
00011  *
00012  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00013  * This file is part of GNSS-SDR.
00014  *
00015  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00016  * SPDX-License-Identifier: GPL-3.0-or-later
00017  *
00018  * -----
00019  */
00020
00021 #ifndef GNSS_SDR_GALILEO_E5_SIGNAL_REPLICA_H
00022 #define GNSS_SDR_GALILEO_E5_SIGNAL_REPLICA_H
00023
00024 #include <array>
00025 #include <complex>
00026 #include <stdint>
00027 #if HAS_STD_SPAN
00028 #include <span>
00029 namespace own = std;
00030 #else
00031 #include <gsl/gsl-lite.hpp>
00032 namespace own = gsl;
00033 #endif
00034
00035 /** \addtogroup Algorithms_Library
00036  * \{ */
00037 /** \addtogroup Algorithm_libs algorithms_libs
00038  * \{ */
00039
00040
00041 /*!
00042  * \brief Generates Galileo E5a code at 1 sample/chip
00043  */
00044 void galileo\_e5\_a\_code\_gen\_complex\_primary(own::span<std::complex<float> > dest,
00045      int32_t prn,
00046      const std::array<char, 3>& signal_id);
00047
```

```

00048
00049 /*!
00050 * \brief Generates Galileo E5a complex code, shifted to the desired chip and
00051 * sampled at a frequency sampling_freq
00052 */
00053 void galileo_e5_a_code_gen_complex_sampled(own::span<std::complex<float> dest,
00054      uint32_t prn,
00055      const std::array<char, 3>& signal_id,
00056      int32_t sampling_freq,
00057      uint32_t chip_shift);
00058
00059
00060 /*!
00061 * \brief Generates Galileo E5b code at 1 sample/chip
00062 */
00063 void galileo_e5_b_code_gen_complex_primary(own::span<std::complex<float> dest,
00064      int32_t prn,
00065      const std::array<char, 3>& signal_id);
00066
00067
00068 /*!
00069 * \brief Generates Galileo E5b complex code, shifted to the desired chip and
00070 * sampled at a frequency sampling_freq
00071 */
00072 void galileo_e5_b_code_gen_complex_sampled(own::span<std::complex<float> dest,
00073      uint32_t prn,
00074      const std::array<char, 3>& signal_id,
00075      int32_t sampling_freq,
00076      uint32_t chip_shift);
00077
00078
00079 /** \} */
00080 /** \} */
00081 #endif // GNSS_SDR_GALILEO_E5_SIGNAL_REPLICA_H

```

## 11.155 galileo\_e6\_signal\_replica.h File Reference

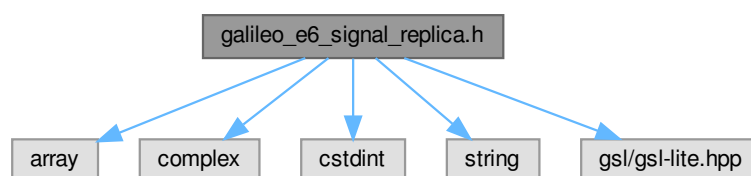
This library implements various functions for Galileo E6 signal replica generation.

```

#include <array>
#include <complex>
#include <cstdint>
#include <string>
#include <gsl/gsl-lite.hpp>

```

Include dependency graph for galileo\_e6\_signal\_replica.h:



### Functions

- void [galileo\\_e6\\_b\\_code\\_gen\\_complex\\_primary](#) (own::span< std::complex< float > > dest, int32\_t prn)  
*Generates Galileo E6B code at 1 sample/chip.*
- void [galileo\\_e6\\_b\\_code\\_gen\\_float\\_primary](#) (own::span< float > dest, int32\_t prn)  
*Generates Galileo E6B code at 1 sample/chip.*
- void [galileo\\_e6\\_b\\_code\\_gen\\_complex\\_sampled](#) (own::span< std::complex< float > > dest, uint32\_t prn, int32\_t sampling\_freq, uint32\_t chip\_shift)  
*Generates Galileo E6B complex code, shifted to the desired chip and sampled at a frequency sampling\_freq.*
- void [galileo\\_e6\\_c\\_code\\_gen\\_complex\\_primary](#) (own::span< std::complex< float > > dest, int32\_t prn)



- Generates Galileo E6C codes at 1 sample/chip.*
- void `galileo_e6_c_code_gen_float_primary` (own::span< float > dest, int32\_t prn)
- Generates Galileo E6C codes at 1 sample/chip.*
- void `galileo_e6_c_code_gen_complex_sampled` (own::span< std::complex< float > > dest, uint32\_t prn, int32\_t sampling\_freq, uint32\_t chip\_shift)
- Generates Galileo E6C complex codes, shifted to the desired chip and sampled at a frequency sampling\_freq.*
- void `galileo_e6_c_secondary_code_gen_complex` (own::span< std::complex< float > > dest, int32\_t prn)
- Generates Galileo E6C secondary codes at 1 sample/chip.*
- void `galileo_e6_c_secondary_code_gen_float` (own::span< float > dest, int32\_t prn)
- Generates Galileo E6C secondary codes at 1 sample/chip.*
- std::string `galileo_e6_c_secondary_code` (int32\_t prn)
- Generates a string with Galileo E6C secondary codes at 1 sample/chip.*

### 11.155.1 Detailed Description

This library implements various functions for Galileo E6 signal replica generation.

#### Author

Carles Fernandez-Prades, 2020. cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file `galileo_e6_signal_replica.h`.

---

## 11.156 galileo\_e6\_signal\_replica.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file galileo_e6_signal_replica.h
00003  * \brief This library implements various functions for Galileo E6 signal
00004  * replica generation
00005  * \author Carles Fernandez-Prades, 2020. cfernandez(at)cttc.es
00006  *
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019 #ifndef GNSS_SDR_GALILEO_E6_SIGNAL_REPLICA_H
00020 #define GNSS_SDR_GALILEO_E6_SIGNAL_REPLICA_H
00021
00022 #include <array>
00023 #include <complex>
00024 #include <cstdint>
00025 #include <string>
00026 #if HAS_STD_SPAN
00027 #include <span>
00028 namespace own = std;
00029 #else
00030 #include <gsl/gsl-lite.hpp>
00031 namespace own = gsl;
00032 #endif
00033
00034 /** \addtogroup Algorithms_Library
00035  * \{ */
00036 /** \addtogroup Algorithm_libs algorithms_libs
00037  * \{ */
00038
00039
00040 /*!
00041  * \brief Generates Galileo E6B code at 1 sample/chip
00042  */
00043 void galileo_e6_b_code_gen_complex_primary(own::span<std::complex<float> > dest,
00044      int32_t prn);

```

```

00045
00046
00047 /*!
00048  * \brief Generates Galileo E6B code at 1 sample/chip
00049  */
00050 void galileo_e6_b_code_gen_float_primary(own::span<float> dest, int32_t prn);
00051
00052
00053 /*!
00054  * \brief Generates Galileo E6B complex code, shifted to the desired chip and
00055  * sampled at a frequency sampling_freq
00056  */
00057 void galileo_e6_b_code_gen_complex_sampled(own::span<std::complex<float> dest,
00058      uint32_t prn,
00059      int32_t sampling_freq,
00060      uint32_t chip_shift);
00061
00062
00063 /*!
00064  * \brief Generates Galileo E6C codes at 1 sample/chip
00065  */
00066 void galileo_e6_c_code_gen_complex_primary(own::span<std::complex<float> dest,
00067      int32_t prn);
00068
00069
00070 /*!
00071  * \brief Generates Galileo E6C codes at 1 sample/chip
00072  */
00073 void galileo_e6_c_code_gen_float_primary(own::span<float> dest, int32_t prn);
00074
00075
00076 /*!
00077  * \brief Generates Galileo E6C complex codes, shifted to the desired chip and
00078  * sampled at a frequency sampling_freq
00079  */
00080 void galileo_e6_c_code_gen_complex_sampled(own::span<std::complex<float> dest,
00081      uint32_t prn,
00082      int32_t sampling_freq,
00083      uint32_t chip_shift);
00084
00085
00086 /*!
00087  * \brief Generates Galileo E6C secondary codes at 1 sample/chip
00088  */
00089 void galileo_e6_c_secondary_code_gen_complex(own::span<std::complex<float> dest,
00090      int32_t prn);
00091
00092
00093 /*!
00094  * \brief Generates Galileo E6C secondary codes at 1 sample/chip
00095  */
00096 void galileo_e6_c_secondary_code_gen_float(own::span<float> dest,
00097      int32_t prn);
00098
00099
00100 /*!
00101  * \brief Generates a string with Galileo E6C secondary codes at 1 sample/chip
00102  */
00103 std::string galileo_e6_c_secondary_code(int32_t prn);
00104
00105
00106 /** \} */
00107 /** \} */
00108 #endif // GNSS_SDR_GALILEO_E6_SIGNAL_REPLICA_H

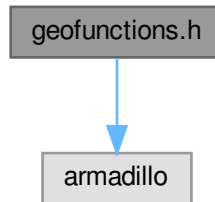
```

## 11.157 geofunctions.h File Reference

A set of coordinate transformations functions and helpers, some of them migrated from MATLAB, for geographic information systems.

```
#include <armadillo>
```

Include dependency graph for geofunctions.h:



## Functions

- arma::mat [Skew\\_symmetric](#) (const arma::vec &a)  
*Calculates skew-symmetric matrix.*
- double **WGS84\_g0** (double Lat\_rad)
- double **WGS84\_geocentric\_radius** (double Lat\_geodetic\_rad)
- int [topocent](#) (double \*Az, double \*El, double \*D, const arma::vec &x, const arma::vec &dx)  
*Transformation of vector dx into topocentric coordinate system with origin at x Inputs: x - vector origin coordinates (in ECEF system [X; Y; Z;]) dx - vector ([dX; dY; dZ;]).*
- int [togeod](#) (double \*dphi, double \*dlambda, double \*h, double a, double finv, double X, double Y, double Z)  
*Subroutine to calculate geodetic coordinates latitude, longitude, height given Cartesian coordinates X,Y,Z, and reference ellipsoid values semi-major axis (a) and the inverse of flattening (finv).*
- arma::vec [Gravity\\_ECEF](#) (const arma::vec &r\_eb\_e)  
*Calculates acceleration due to gravity resolved about ECEF-frame.*
- arma::vec [cart2geo](#) (const arma::vec &XYZ, int ellipsoid\_selection)  
*Conversion of Cartesian coordinates (X,Y,Z) to geographical coordinates (latitude, longitude, h) on a selected reference ellipsoid.*
- arma::vec **LLH\_to\_deg** (const arma::vec &LLH)
- double **degtorad** (double angleInDegrees)
- double **radtodeg** (double angleInRadians)
- double **mstoknotsh** (double MetersPerSeconds)
- double **mstokph** (double MetersPerSeconds)
- arma::vec **CTM\_to\_Euler** (const arma::mat &C)
- arma::mat **Euler\_to\_CTM** (const arma::vec &eul)
- void **ECEF\_to\_Geo** (const arma::vec &r\_eb\_e, const arma::vec &v\_eb\_e, const arma::mat &C\_b\_e, arma::vec &LLH, arma::vec &v\_eb\_n, arma::mat &C\_b\_n)
- void [Geo\\_to\\_ECEF](#) (const arma::vec &LLH, const arma::vec &v\_eb\_n, const arma::mat &C\_b\_n, arma::vec &r\_eb\_e, arma::vec &v\_eb\_e, arma::mat &C\_b\_e)  
*From Geographic to ECEF coordinates.*
- void [pv\\_Geo\\_to\\_ECEF](#) (double L\_b, double lambda\_b, double h\_b, const arma::vec &v\_eb\_n, arma::vec &r\_eb\_e, arma::vec &v\_eb\_e)  
*Converts curvilinear to Cartesian position and velocity resolving axes from NED to ECEF This function created 11/4/2012 by Paul Groves.*
- double [great\\_circle\\_distance](#) (double lat1, double lon1, double lat2, double lon2)  
*The Haversine formula determines the great-circle distance between two points on a sphere given their longitudes and latitudes.*
- void [cart2utm](#) (const arma::vec &r\_eb\_e, int zone, arma::vec &r\_enu)  
*Transformation of ECEF (X,Y,Z) to (E,N,U) in UTM, zone 'zone'.*

- int [findUtmZone](#) (double latitude\_deg, double longitude\_deg)  
*Function finds the UTM zone number for given longitude and latitude.*
- double [clsin](#) (const arma::colvec &ar, int degree, double argument)  
*Clenshaw summation of sinus of argument.*
- void [clkisin](#) (const arma::colvec &ar, int degree, double arg\_real, double arg\_imag, double \*re, double \*im)  
*Clenshaw summation of sinus with complex argument.*

### 11.157.1 Detailed Description

A set of coordinate transformations functions and helpers, some of them migrated from MATLAB, for geographic information systems.

#### Author

Javier Arribas, 2018. [jarribas\(at\)cttc.es](mailto:jarribas(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [geofunctions.h](#).

---

## 11.158 geofunctions.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file geofunctions.h
00003  * \brief A set of coordinate transformations functions and helpers,
00004  * some of them migrated from MATLAB, for geographic information systems.
00005  * \author Javier Arribas, 2018. jarribas\(at\)cttc.es
00006  *
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019 #ifndef GNSS_SDR_GEOFUNCTIONS_H
00020 #define GNSS_SDR_GEOFUNCTIONS_H
00021
00022 #if ARMA_NO_BOUND_CHECKING
00023 #define ARMA_NO_DEBUG 1
00024 #endif
00025
00026 #include <armadillo>
00027
00028 /** \addtogroup Algorithms_Library
00029  * \{ */
00030 /** \addtogroup Algorithm_libs algorithms_libs
00031  * \{ */
00032
00033 00034 arma::mat Skew_symmetric(const arma::vec &a); //!< Calculates skew-symmetric matrix
00035
00036 double WGS84_g0(double Lat_rad);
00037
00038 double WGS84_geocentric_radius(double Lat_geodetic_rad);
00039
00040 /*!
00041  * \brief Transformation of vector dx into topocentric coordinate
00042  * system with origin at x
00043  * Inputs:
00044  * x - vector origin coordinates (in ECEF system [X; Y; Z])
00045  * dx - vector ([dX; dY; dZ;]).
00046  *
00047  * Outputs:
00048  * D - vector length. Units like the input
00049  * Az - azimuth from north positive clockwise, degrees
00050  * El - elevation angle, degrees
00051  *
00052  * Based on a Matlab function by Kai Borre
00053  */
```

```

00054 int topocent(double *Az, double *El, double *D, const arma::vec &x, const arma::vec &dx);
00055
00056 /*!
00057 * \brief Subroutine to calculate geodetic coordinates latitude, longitude,
00058 * height given Cartesian coordinates X,Y,Z, and reference ellipsoid
00059 * values semi-major axis (a) and the inverse of flattening (finv).
00060 *
00061 * The output units of angular quantities will be in decimal degrees
00062 * (15.5 degrees not 15 deg 30 min). The output units of h will be the
00063 * same as the units of X,Y,Z,a.
00064 *
00065 * Inputs:
00066 * a - semi-major axis of the reference ellipsoid
00067 * finv - inverse of flattening of the reference ellipsoid
00068 * X,Y,Z - Cartesian coordinates
00069 *
00070 * Outputs:
00071 * dphi - latitude
00072 * dlambda - longitude
00073 * h - height above reference ellipsoid
00074 *
00075 * Based in a Matlab function by Kai Borre
00076 */
00077 int togeod(double *dphi, double *dlambda, double *h, double a, double finv, double X, double Y, double
Z);
00078
00079 arma::vec Gravity_ECEF(const arma::vec &r_eb_e); //!< Calculates acceleration due to gravity resolved
about ECEF-frame
00080
00081 /*!
00082 * \brief Conversion of Cartesian coordinates (X,Y,Z) to geographical
00083 * coordinates (latitude, longitude, h) on a selected reference ellipsoid.
00084 *
00085 * Choices of Reference Ellipsoid for Geographical Coordinates
00086 * 0. International Ellipsoid 1924
00087 * 1. International Ellipsoid 1967
00088 * 2. World Geodetic System 1972
00089 * 3. Geodetic Reference System 1980
00090 * 4. World Geodetic System 1984
00091 */
00092 arma::vec cart2geod(const arma::vec &XYZ, int ellipsoid_selection);
00093
00094 arma::vec LLH_to_deg(const arma::vec &LLH);
00095
00096 double degtorad(double angleInDegrees);
00097
00098 double radtodeg(double angleInRadians);
00099
00100 double mstoknotsh(double MetersPerSeconds);
00101
00102 double mstokph(double MetersPerSeconds);
00103
00104 arma::vec CTM_to_Euler(const arma::mat &C);
00105
00106 arma::mat Euler_to_CTM(const arma::vec &eul);
00107
00108 void ECEF_to_Geo(const arma::vec &r_eb_e, const arma::vec &v_eb_e, const arma::mat &C_b_e, arma::vec
&LLH, arma::vec &v_eb_n, arma::mat &C_b_n);
00109
00110
00111 /*!
00112 * \brief From Geographic to ECEF coordinates
00113 *
00114 * Inputs:
00115 * LLH latitude (rad), longitude (rad), height (m)
00116 * v_eb_n velocity of body frame w.r.t. ECEF frame, resolved along
00117 * north, east, and down (m/s)
00118 * C_b_n body-to-NED coordinate transformation matrix
00119 *
00120 * Outputs:
00121 * r_eb_e Cartesian position of body frame w.r.t. ECEF frame, resolved
00122 * along ECEF-frame axes (m)
00123 * v_eb_e velocity of body frame w.r.t. ECEF frame, resolved along
00124 * ECEF-frame axes (m/s)
00125 * C_b_e body-to-ECEF-frame coordinate transformation matrix
00126 *
00127 */
00128 void Geo_to_ECEF(const arma::vec &LLH, const arma::vec &v_eb_n, const arma::mat &C_b_n, arma::vec
&r_eb_e, arma::vec &v_eb_e, arma::mat &C_b_e);
00129
00130
00131 /*!
00132 * \brief Converts curvilinear to Cartesian position and velocity
00133 * resolving axes from NED to ECEF
00134 * This function created 11/4/2012 by Paul Groves
00135 *
00136 * Inputs:

```

```

00137 *   L_b           latitude (rad)
00138 *   lambda_b      longitude (rad)
00139 *   h_b           height (m)
00140 *   v_eb_n        velocity of body frame w.r.t. ECEF frame, resolved along
00141 *                 north, east, and down (m/s)
00142 *
00143 * Outputs:
00144 *   r_eb_e         Cartesian position of body frame w.r.t. ECEF frame, resolved
00145 *                 along ECEF-frame axes (m)
00146 *   v_eb_e         velocity of body frame w.r.t. ECEF frame, resolved along
00147 *                 ECEF-frame axes (m/s)
00148 */
00149 void pv_Geo_to_ECEF(double L_b, double lambda_b, double h_b, const arma::vec &v_eb_n, arma::vec
&r_eb_e, arma::vec &v_eb_e);
00150
00151
00152 /*!
00153 * \brief The Haversine formula determines the great-circle distance between two points on a sphere
given their longitudes and latitudes.
00154 */
00155 double great_circle_distance(double lat1, double lon1, double lat2, double lon2);
00156
00157 /*!
00158 * \brief Transformation of ECEF (X,Y,Z) to (E,N,U) in UTM, zone 'zone'.
00160 */
00161 void cart2utm(const arma::vec &r_eb_e, int zone, arma::vec &r_enu);
00162
00163 /*!
00164 * \brief Function finds the UTM zone number for given longitude and latitude.
00166 */
00167 int findUtmZone(double latitude_deg, double longitude_deg);
00168
00169 /*!
00170 * \brief Clenshaw summation of sinus of argument.
00172 */
00173 double clsin(const arma::colvec &ar, int degree, double argument);
00174
00175 /*!
00176 * \brief Clenshaw summation of sinus with complex argument.
00178 */
00179 void clksin(const arma::colvec &ar, int degree, double arg_real, double arg_imag, double *re, double
*im);
00180
00181
00182 /** \} */
00183 /** \} */
00184 #endif // GNSS_SDR_GEOFUNCTIONS_H

```

## 11.159 glonass\_l1\_signal\_replica.h

```

00001 /*!
00002 * \file glonass_l1_signal_processing.h
00003 * \brief This file implements various functions for GLONASS L1 CA signal
00004 * replica generation
00005 * \author Gabriel Araujo, 2017. gabriel.araujo(at)ieee.org
00006 *
00007 *
00008 * -----
00009 *
00010 * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011 * This file is part of GNSS-SDR.
00012 *
00013 * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014 * SPDX-License-Identifier: GPL-3.0-or-later
00015 *
00016 * -----
00017 */
00018
00019 #ifndef GNSS_SDR_GLOONASS_L1_SIGNAL_REPLICA_H
00020 #define GNSS_SDR_GLOONASS_L1_SIGNAL_REPLICA_H
00021
00022 #include <complex>
00023 #include <stdint>
00024 #if HAS_STD_SPAN
00025 #include <span>
00026 namespace own = std;
00027 #else
00028 #include <gsl/gsl-lite.hpp>
00029 namespace own = gsl;
00030 #endif

```

```

00031
00032 /** \addtogroup Algorithms_Library
00033 * \{ */
00034 /** \addtogroup Algorithm_libs algorithms_libs
00035 * \{ */
00036
00037
00038 //! Generates complex GLONASS L1 C/A code for the desired SV ID and code shift
00039 void glonass_l1_ca_code_gen_complex(own::span<std::complex<float> dest, uint32_t chip_shift);
00040
00041 //! Generates complex GLONASS L1 C/A code for the desired SV ID and code shift, and sampled to
    specific sampling frequency
00042 void glonass_l1_ca_code_gen_complex_sampled(own::span<std::complex<float> dest, int32_t sampling_freq,
    uint32_t chip_shift);
00043
00044
00045 /** \} */
00046 /** \} */
00047 #endif // GNSS_SDR_GLOMSS_L1_SIGNAL_REPLICA_H

```

## 11.160 glonass\_l2\_signal\_replica.h File Reference

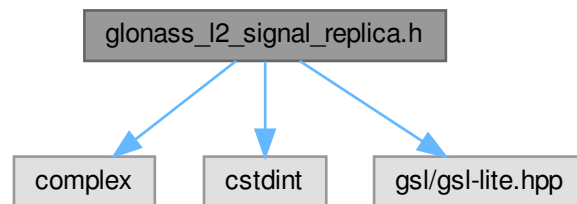
This file implements various functions for GLONASS L2 CA signal replica generation.

```
#include <complex>
```

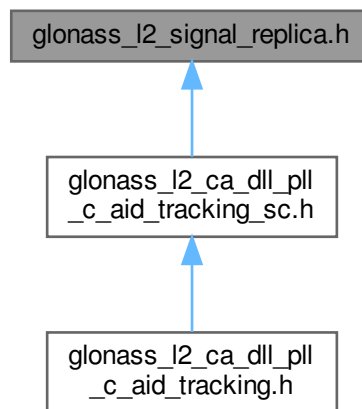
```
#include <cstdint>
```

```
#include <gsl/gsl-lite.hpp>
```

Include dependency graph for glonass\_l2\_signal\_replica.h:



This graph shows which files directly or indirectly include this file:



## Functions

- void [glonass\\_l2\\_ca\\_code\\_gen\\_complex](#) (own::span< std::complex< float > > dest, uint32\_t chip\_shift)  
*Generates complex GLONASS L2 C/A code for the desired SV ID and code shift.*
- void [glonass\\_l2\\_ca\\_code\\_gen\\_complex\\_sampled](#) (own::span< std::complex< float > > dest, int32\_t sampling\_freq, uint32\_t chip\_shift)  
*Generates complex GLONASS L2 C/A code for the desired SV ID and code shift, and sampled to specific sampling frequency.*

### 11.160.1 Detailed Description

This file implements various functions for GLONASS L2 CA signal replica generation.

#### Author

Damian Miralles, 2018, dmiralles2009(at)gmail.com

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [glonass\\_l2\\_signal\\_replica.h](#).

---

## 11.161 glonass\_l2\_signal\_replica.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file glonass_l2_signal_replica.h
00003  * \brief This file implements various functions for GLONASS L2 CA signal
00004  * replica generation
00005  * \author Damian Miralles, 2018, dmiralles2009(at)gmail.com
00006  *
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
  
```



```

00018
00019 #ifndef GNSS_SDR_GLONASS_L2_SIGNAL_REPLICA_H
00020 #define GNSS_SDR_GLONASS_L2_SIGNAL_REPLICA_H
00021
00022 #include <complex>
00023 #include <cstdint>
00024 #if HAS_STD_SPAN
00025 #include <span>
00026 namespace own = std;
00027 #else
00028 #include <gsl/gsl-lite.hpp>
00029 namespace own = gsl;
00030 #endif
00031
00032 /** \addtogroup Algorithms_Library
00033  * \{ */
00034 /** \addtogroup Algorithm_libs algorithms_libs
00035  * \{ */
00036
00037
00038 /*! Generates complex GLONASS L2 C/A code for the desired SV ID and code shift
00039 void glonass_l2_ca_code_gen_complex(own::span<std::complex<float>> dest, uint32_t chip_shift);
00040
00041 /*! Generates complex GLONASS L2 C/A code for the desired SV ID and code shift, and sampled to
    specific sampling frequency
00042 void glonass_l2_ca_code_gen_complex_sampled(own::span<std::complex<float>> dest, int32_t sampling_freq,
    uint32_t chip_shift);
00043
00044
00045 /** \} */
00046 /** \} */
00047 #endif // GNSS_SDR_GLONASS_L2_SIGNAL_REPLICA_H

```

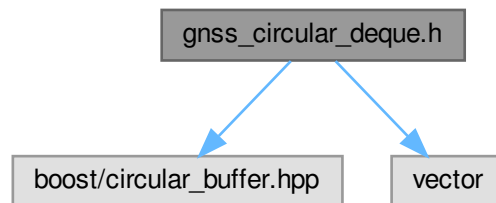
## 11.162 gnss\_circular\_deque.h File Reference

This class implements a circular deque for [Gnss\\_Synchro](#).

```
#include <boost/circular_buffer.hpp>
```

```
#include <vector>
```

Include dependency graph for gnss\_circular\_deque.h:



### Classes

- class [Gnss\\_circular\\_deque< T >](#)

### 11.162.1 Detailed Description

This class implements a circular deque for [Gnss\\_Synchro](#).

#### Author

Antonio Ramos, 2018. antonio.ramosdet(at)gmail.com

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gnss\\_circular\\_deque.h](#).

---

## 11.163 gnss\_circular\_deque.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file gnss_circular_deque.h
00003  * \brief This class implements a circular deque for Gnss_Synchro
00004  * \author Antonio Ramos, 2018. antonio.ramosdet(at)gmail.com
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_CIRCULAR_DEQUE_H
00019 #define GNSS_SDR_CIRCULAR_DEQUE_H
00020
00021 #include <boost/circular_buffer.hpp>
00022 #include <vector>
00023
00024 /** \addtogroup Algorithms_Library
00025  * \{ */
00026 /** \addtogroup Algorithm_libs algorithms_libs
00027  * \{ */
00028
00029
00030 template <class T>
00031 class Gnss_circular_deque
00032 {
00033 public:
00034     Gnss_circular_deque(); //!< Default constructor
00035     Gnss_circular_deque(unsigned int max_size, unsigned int nchann); //!< nchann = number of
        channels; max_size = channel capacity
00036     unsigned int size(unsigned int ch) const; //!< Returns the number of
        available elements in a channel
00037     T& at(unsigned int ch, unsigned int pos); //!< Returns a reference to an
        element with bound checking
00038     const T& get(unsigned int ch, unsigned int pos) const; //!< Returns a const reference
        to an element without bound checking
00039     T& front(unsigned int ch); //!< Returns a reference to the
        first element in the deque
00040     T& back(unsigned int ch); //!< Returns a reference to the
        last element in the deque
00041     void push_back(unsigned int ch, const T& new_data); //!< Inserts an element at the
        end of the deque
00042     void pop_front(unsigned int ch); //!< Removes the first element
        of the deque
00043     void clear(unsigned int ch); //!< Removes all the elements of
        the deque (Sets size to 0). Capacity is not modified
00044     void reset(unsigned int max_size, unsigned int nchann); //!< Removes all the elements in
        all the channels. Re-sets the number of channels and their capacity
00045     void reset(); //!< Removes all the channels
        (Sets nchann to 0)
00046
00047 private:
00048     std::vector<boost::circular_buffer<T> > d_data;
00049 };
00050
00051
00052 template <class T>
00053 Gnss_circular_deque<T>::Gnss_circular_deque()
00054 {
00055     reset();
00056 }
00057
00058
00059 template <class T>
00060 Gnss_circular_deque<T>::Gnss_circular_deque(unsigned int max_size, unsigned int nchann)
00061 {
00062     reset(max_size, nchann);
00063 }
00064
00065
00066 template <class T>
00067 unsigned int Gnss_circular_deque<T>::size(unsigned int ch) const
00068 {
00069     return d_data[ch].size();
00070 }
00071
00072

```

```

00073 template <class T>
00074 T& Gnss_circular_deque<T>::back(unsigned int ch)
00075 {
00076     return d_data[ch].back();
00077 }
00078
00079
00080 template <class T>
00081 T& Gnss_circular_deque<T>::front(unsigned int ch)
00082 {
00083     return d_data[ch].front();
00084 }
00085
00086
00087 template <class T>
00088 T& Gnss_circular_deque<T>::at(unsigned int ch, unsigned int pos)
00089 {
00090     return d_data.at(ch).at(pos);
00091 }
00092
00093
00094 template <class T>
00095 const T& Gnss_circular_deque<T>::get(unsigned int ch, unsigned int pos) const
00096 {
00097     return d_data[ch][pos];
00098 }
00099
00100
00101 template <class T>
00102 void Gnss_circular_deque<T>::clear(unsigned int ch)
00103 {
00104     d_data[ch].clear();
00105 }
00106
00107
00108 template <class T>
00109 void Gnss_circular_deque<T>::reset(unsigned int max_size, unsigned int nchann)
00110 {
00111     d_data.clear();
00112     if (max_size > 0 and nchann > 0)
00113     {
00114         for (unsigned int i = 0; i < nchann; i++)
00115         {
00116             d_data.push_back(boost::circular_buffer<T>(max_size));
00117         }
00118     }
00119 }
00120
00121
00122 template <class T>
00123 void Gnss_circular_deque<T>::reset()
00124 {
00125     d_data.clear();
00126 }
00127
00128
00129 template <class T>
00130 void Gnss_circular_deque<T>::pop_front(unsigned int ch)
00131 {
00132     d_data[ch].pop_front();
00133 }
00134
00135
00136 template <class T>
00137 void Gnss_circular_deque<T>::push_back(unsigned int ch, const T& new_data)
00138 {
00139     d_data[ch].push_back(new_data);
00140 }
00141
00142
00143 /** \} */
00144 /** \} */
00145 #endif // GNSS_SDR_CIRCULAR_DEQUE_H

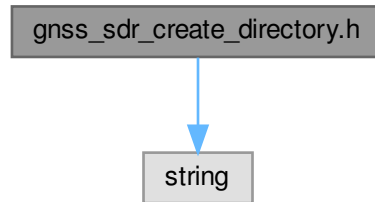
```

## 11.164 gnss\_sdr\_create\_directory.h File Reference

Create a directory.

```
#include <string>
```

Include dependency graph for gnss\_sdr\_create\_directory.h:



## Functions

- `bool gnss_sdr_create_directory (const std::string &foldername)`

### 11.164.1 Detailed Description

Create a directory.

#### Author

Carles Fernandez-Prades, 2018. cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gnss\\_sdr\\_create\\_directory.h](#).

---

## 11.165 gnss\_sdr\_create\_directory.h

[Go to the documentation of this file.](#)

```

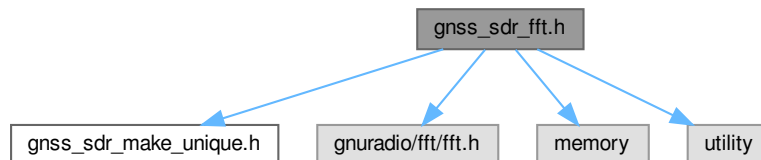
00001 /*!
00002  * \file gnss_sdr_create_directory.h
00003  * \brief Create a directory
00004  * \author Carles Fernandez-Prades, 2018. cfernandez(at)cttc.es
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_GNSS_SDR_CREATE_DIRECTORY_H
00019 #define GNSS_SDR_GNSS_SDR_CREATE_DIRECTORY_H
00020
00021 #include <string>
00022
00023 /** \addtogroup Algorithms_Library
00024  * \{ */
00025 /** \addtogroup Algorithm_libs algorithms_libs
00026  * \{ */
00027
00028
00029 bool gnss_sdr_create_directory(const std::string& foldername);
00030
00031
00032 /** \} */
00033 /** \} */
00034 #endif // GNSS_SDR_GNSS_SDR_CREATE_DIRECTORY_H
  
```

## 11.166 gnss\_sdr\_fft.h File Reference

Helper file for FFT interface.

```
#include "gnss_sdr_make_unique.h"
#include <gnuradio/fft/fft.h>
#include <memory>
#include <utility>
```

Include dependency graph for gnss\_sdr\_fft.h:



This graph shows which files directly or indirectly include this file:



### Typedefs

- using [gnss\\_fft\\_complex\\_fwd](#) = gr::fft::fft\_complex
- using [gnss\\_fft\\_complex\\_rev](#) = gr::fft::fft\_complex
- template<typename T>  
using [gnss\\_fft\\_fwd\\_unique\\_ptr](#) = std::unique\_ptr<T>
- template<typename T>  
using [gnss\\_fft\\_rev\\_unique\\_ptr](#) = std::unique\_ptr<T>

### Functions

- template<typename... Args>  
gnss\_fft\_fwd\_unique\_ptr< gr::fft::fft\_complex > [gnss\\_fft\\_fwd\\_make\\_unique](#) (Args &&... args)
- template<typename... Args>  
gnss\_fft\_rev\_unique\_ptr< gr::fft::fft\_complex > [gnss\\_fft\\_rev\\_make\\_unique](#) (Args &&... args)

### 11.166.1 Detailed Description

Helper file for FFT interface.

Author

Carles Fernandez Prades, 2021. cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [gnss\\_sdr\\_fft.h](#).

---

### 11.166.2 Typedef Documentation

#### 11.166.2.1 gnss\_fft\_complex\_fwd

using [gnss\\_fft\\_complex\\_fwd](#) = gr::fft::fft\_complex

Definition at line 46 of file [gnss\\_sdr\\_fft.h](#).

### 11.166.2.2 gnss\_fft\_complex\_rev

using gnss\_fft\_complex\_rev = gr::fft::fft\_complex

Definition at line 47 of file [gnss\\_sdr\\_fft.h](#).

### 11.166.2.3 gnss\_fft\_fwd\_unique\_ptr

template<typename T>

using gnss\_fft\_fwd\_unique\_ptr = std::unique\_ptr<T>

Definition at line 49 of file [gnss\\_sdr\\_fft.h](#).

### 11.166.2.4 gnss\_fft\_rev\_unique\_ptr

template<typename T>

using gnss\_fft\_rev\_unique\_ptr = std::unique\_ptr<T>

Definition at line 56 of file [gnss\\_sdr\\_fft.h](#).

## 11.166.3 Function Documentation

### 11.166.3.1 gnss\_fft\_fwd\_make\_unique()

template<typename... Args>

gnss\_fft\_fwd\_unique\_ptr< gr::fft::fft\_complex > gnss\_fft\_fwd\_make\_unique (  
     Args &&... args)

Definition at line 51 of file [gnss\\_sdr\\_fft.h](#).

### 11.166.3.2 gnss\_fft\_rev\_make\_unique()

template<typename... Args>

gnss\_fft\_rev\_unique\_ptr< gr::fft::fft\_complex > gnss\_fft\_rev\_make\_unique (  
     Args &&... args)

Definition at line 58 of file [gnss\\_sdr\\_fft.h](#).

## 11.167 gnss\_sdr\_fft.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002   * \file gnss_sdr_fft.h
00003   * \brief Helper file for FFT interface
00004   * \author Carles Fernandez Prades, 2021. cfernandez(at)cttc.es
00005   *
00006   * -----
00007   *
00008   * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009   * This file is part of GNSS-SDR.
00010   *
00011   * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)
00012   * SPDX-License-Identifier: GPL-3.0-or-later
00013   *
00014   * -----
00015   */
00016
00017
00018 #ifndef GNSS_SDR_GNSS_SDR_FFT_H
00019 #define GNSS_SDR_GNSS_SDR_FFT_H
00020
00021 #include "gnss_sdr_make_unique.h"
00022 #include <gnuradio/fft/fft.h>
00023 #include <memory>
00024 #include <utility>
00025
00026 #if GNURADIO_FFT_USES_TEMPLATES
00027 using gnss_fft_complex_fwd = gr::fft::fft_complex_fwd;
00028 using gnss_fft_complex_rev = gr::fft::fft_complex_rev;
00029 template <typename T>
00030 using gnss_fft_fwd_unique_ptr = std::unique_ptr<T>;
00031 template <typename... Args>
00032 gnss_fft_fwd_unique_ptr<gr::fft::fft_complex_fwd> gnss_fft_fwd_make_unique(Args&&... args)
00033 {
00034     return std::make_unique<gr::fft::fft_complex_fwd>(std::forward<Args>(args)...);

```

```

00035 }
00036 template <typename T>
00037 using gnss_fft_rev_unique_ptr = std::unique_ptr<T>;
00038 template <typename... Args>
00039 gnss_fft_rev_unique_ptr<gr::fft::fft_complex_rev> gnss_fft_rev_make_unique(Args&&... args)
00040 {
00041     return std::make_unique<gr::fft::fft_complex_rev>(std::forward<Args>(args)...);
00042 }
00043
00044 #else
00045
00046 using gnss_fft_complex_fwd = gr::fft::fft_complex;
00047 using gnss_fft_complex_rev = gr::fft::fft_complex;
00048 template <typename T>
00049 using gnss_fft_fwd_unique_ptr = std::unique_ptr<T>;
00050 template <typename... Args>
00051 gnss_fft_fwd_unique_ptr<gr::fft::fft_complex> gnss_fft_fwd_make_unique(Args&&... args)
00052 {
00053     return std::make_unique<gr::fft::fft_complex>(std::forward<Args>(args)..., true);
00054 }
00055 template <typename T>
00056 using gnss_fft_rev_unique_ptr = std::unique_ptr<T>;
00057 template <typename... Args>
00058 gnss_fft_rev_unique_ptr<gr::fft::fft_complex> gnss_fft_rev_make_unique(Args&&... args)
00059 {
00060     return std::make_unique<gr::fft::fft_complex>(std::forward<Args>(args)..., false);
00061 }
00062
00063 #endif
00064
00065 #endif // GNSS_SDR_GNSS_SDR_FFT_H

```

## 11.168 gnss\_sdr\_filesystem.h File Reference

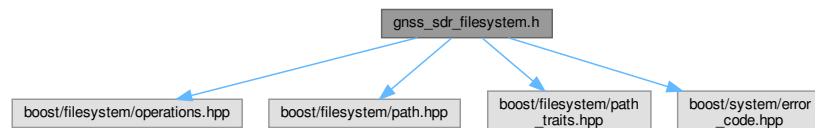
Helper file for filesystem library interface.

```

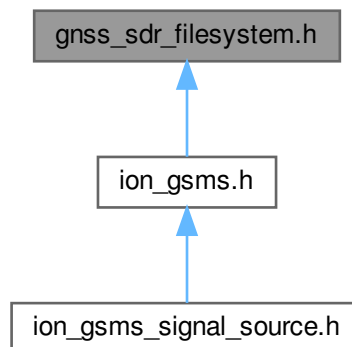
#include <boost/filesystem/operations.hpp>
#include <boost/filesystem/path.hpp>
#include <boost/filesystem/path_traits.hpp>
#include <boost/system/error_code.hpp>

```

Include dependency graph for gnss\_sdr\_filesystem.h:



This graph shows which files directly or indirectly include this file:



### 11.168.1 Detailed Description

Helper file for fylesystem library interface.

#### Author

Carles Fernandez Prades, 2021. cfernandez(at)cttc.es

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [gnss\\_sdr\\_filesystem.h](#).

## 11.169 gnss\_sdr\_filesystem.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002   * \file gnss_sdr_filesystem.h
00003   * \brief Helper file for fylesystem library interface
00004   * \author Carles Fernandez Prades, 2021. cfernandez(at)cttc.es
00005   *
00006   * -----
00007   *
00008   * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009   * This file is part of GNSS-SDR.
00010   *
00011   * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)
00012   * SPDX-License-Identifier: GPL-3.0-or-later
00013   *
00014   * -----
00015   */
00016
00017
00018 #ifndef GNSS_SDR_GNSS_SDR_FILESYSTEM_H
00019 #define GNSS_SDR_GNSS_SDR_FILESYSTEM_H
00020
00021 // clang-format off
00022 #if HAS_STD_FILESYSTEM
00023 #include <system_error>
00024 namespace errorlib = std;
00025 #if HAS_STD_FILESYSTEM_EXPERIMENTAL
00026 #include <experimental/filesystem>
00027 namespace fs = std::experimental::filesystem;
00028 #else
00029 #include <filesystem>
00030 namespace fs = std::filesystem;
00031 #endif
00032 #else
00033 #include <boost/filesystem/operations.hpp> // for create_directories, exists
00034 #include <boost/filesystem/path.hpp>      // for path, operator«
  
```



```

00035 #include <boost/filesystem/path_traits.hpp> // for filesystem
00036 #include <boost/system/error_code.hpp>      // for error_code
00037 namespace fs = boost::filesystem;
00038 namespace errorlib = boost::system;
00039 #endif
00040 // clang-format on
00041
00042
00043 #endif // GNSS_SDR_GNSS_SDR_FILESYSTEM_H

```

## 11.170 gnss\_sdr\_flags.h File Reference

Helper file for gnss-sdr commandline flags.

```

#include <stdint>
#include <string>
#include <absl/flags/declare.h>
#include <absl/flags/flag.h>
#include <cstdlib>
#include <iostream>
#include <sys/stat.h>
#include <vector>

```

Include dependency graph for gnss\_sdr\_flags.h:



### Functions

- [ABSL\\_DECLARE\\_FLAG](#) (std::string, c)  
*Path to the configuration file.*
- [ABSL\\_DECLARE\\_FLAG](#) (std::string, config\_file)  
*Path to the configuration file.*
- [ABSL\\_DECLARE\\_FLAG](#) (std::string, log\_dir)  
*Path to the folder in which logging will be stored.*
- [ABSL\\_DECLARE\\_FLAG](#) (std::string, s)  
*Path to the file containing the signal samples.*
- [ABSL\\_DECLARE\\_FLAG](#) (std::string, signal\_source)  
*Path to the file containing the signal samples.*
- [ABSL\\_DECLARE\\_FLAG](#) (std::string, timestamp\_source)  
*Path to the file containing the signal samples.*
- [ABSL\\_DECLARE\\_FLAG](#) (bool, rf\_shutdown)  
*Shutdown RF when program exits.*
- [ABSL\\_DECLARE\\_FLAG](#) (int32\_t, doppler\_max)  
*If defined, maximum Doppler value in the search grid, in Hz (overrides the configuration file).*
- [ABSL\\_DECLARE\\_FLAG](#) (int32\_t, doppler\_step)  
*If defined, sets the frequency step in the search grid, in Hz, in Hz (overrides the configuration file).*
- [ABSL\\_DECLARE\\_FLAG](#) (int32\_t, cn0\_samples)  
*Number of correlator outputs used for CN0 estimation.*
- [ABSL\\_DECLARE\\_FLAG](#) (int32\_t, cn0\_min)  
*Minimum valid CN0 (in dB-Hz).*
- [ABSL\\_DECLARE\\_FLAG](#) (int32\_t, max\_lock\_fail)

- *Maximum number of code lock failures before dropping a satellite.*  
• [ABSL\\_DECLARE\\_FLAG](#) (int32\_t, max\_carrier\_lock\_fail)
- *Maximum number of carrier lock failures before dropping a satellite.*  
• [ABSL\\_DECLARE\\_FLAG](#) (double, carrier\_lock\_th)
- *Carrier lock threshold (in rad).*  
• [ABSL\\_DECLARE\\_FLAG](#) (double, dll\_bw\_hz)
- *Bandwidth of the DLL low pass filter, in Hz (overrides the configuration file).*  
• [ABSL\\_DECLARE\\_FLAG](#) (double, pll\_bw\_hz)
- *Bandwidth of the PLL low pass filter, in Hz (overrides the configuration file).*  
• [ABSL\\_DECLARE\\_FLAG](#) (int32\_t, carrier\_smoothing\_factor)
- *Sets carrier smoothing factor M (overrides the configuration file).*  
• [ABSL\\_DECLARE\\_FLAG](#) (std::string, RINEX\_version)
- *If defined, specifies the RINEX version (2.11 or 3.02). Overrides the configuration file.*  
• [ABSL\\_DECLARE\\_FLAG](#) (std::string, RINEX\_name)
- *If defined, specifies the RINEX files base name.*  
• [ABSL\\_DECLARE\\_FLAG](#) (bool, keyboard)
- *If set to false, disables the keyboard listener. Only for debug purposes (e.g. ASAN mode termination)*  
• bool [ValidateFlags](#) ()

## Variables

- const int32\_t [DEFAULT\\_CARRIER\\_SMOOTHING\\_FACTOR](#) = 200

## 11.170.1 Detailed Description

Helper file for gnss-sdr commandline flags.

### Author

Carles Fernandez-Prades, 2018-2024. cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [gnss\\_sdr\\_flags.h](#).

---

## 11.171 gnss\_sdr\_flags.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file gnss_sdr_flags.h
00003  * \brief Helper file for gnss-sdr commandline flags
00004  * \author Carles Fernandez-Prades, 2018-2024. cfernandez(at)cttc.es
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_GNSS_SDR_FLAGS_H
00019 #define GNSS_SDR_GNSS_SDR_FLAGS_H
00020
00021 #include <stdint>
00022 #include <string>
00023
00024 #if USE_GLOG_AND_GFLAGS
00025 #include <gflags/gflags.h>
00026 #else
00027 #include <absl/flags/declare.h>
00028 #include <absl/flags/flag.h>
```

```

00029 #include <cstdlib>
00030 #include <iostream>
00031 #include <sys/stat.h>
00032 #include <vector>
00033 #endif
00034
00035
00036 /** \addtogroup Algorithms_Library
00037  * \{ */
00038 /** \addtogroup Gflags gnss_sdr_flags
00039  * Library for command-line handling.
00040  * \{ */
00041
00042 #if USE_GLOG_AND_GFLAGS
00043 DECLARE_string(c);          //!< Path to the configuration file.
00044 DECLARE_string(config_file); //!< Path to the configuration file.
00045
00046 DECLARE_string(log_dir);    //!< Path to the folder in which logging will be stored.
00047
00048 // Declare flags for signal sources
00049 DECLARE_string(s);          //!< Path to the file containing the signal samples.
00050 DECLARE_string(signal_source); //!< Path to the file containing the signal samples.
00051 DECLARE_string(timestamp_source); //!< Path to the file containing the signal samples.
00052 DECLARE_bool(rf_shutdown);   //!< Shutdown RF when program exits.
00053
00054 // Declare flags for acquisition blocks
00055 DECLARE_int32(doppler_max);   //!< If defined, maximum Doppler value in the search grid, in Hz
                                (overrides the configuration file).
00056 DECLARE_int32(doppler_step);  //!< If defined, sets the frequency step in the search grid, in Hz, in
                                Hz (overrides the configuration file).
00057
00058 // Declare flags for tracking blocks
00059 DECLARE_int32(cn0_samples);   //!< Number of correlator outputs used for CN0 estimation.
00060 DECLARE_int32(cn0_min);       //!< Minimum valid CN0 (in dB-Hz).
00061 DECLARE_int32(max_lock_fail); //!< Maximum number of code lock failures before dropping a
                                satellite.
00062 DECLARE_int32(max_carrier_lock_fail); //!< Maximum number of carrier lock failures before dropping a
                                satellite.
00063 DECLARE_double(carrier_lock_th); //!< Carrier lock threshold (in rad).
00064 DECLARE_double(dll_bw_hz);      //!< Bandwidth of the DLL low pass filter, in Hz (overrides the
                                configuration file).
00065 DECLARE_double pll_bw_hz;       //!< Bandwidth of the PLL low pass filter, in Hz (overrides the
                                configuration file).
00066
00067 // Declare flags for observables block
00068 DECLARE_int32(carrier_smoothing_factor); //!< Sets carrier smoothing factor M (overrides the
                                configuration file).
00069 const int32_t DEFAULT_CARRIER_SMOOTHING_FACTOR = 200;
00070
00071 // Declare flags for PVT
00072 DECLARE_string(RINEX_version); //!< If defined, specifies the RINEX version (2.11 or 3.02). Overrides
                                the configuration file.
00073 DECLARE_string(RINEX_name);    //!< If defined, specifies the RINEX files base name
00074 DECLARE_bool(keyboard);        //!< If set to false, disables the keyboard listener. Only for debug
                                purposes (e.g. ASAN mode termination)
00075
00076 #else
00077 ABSL_DECLARE_FLAG(std::string, c);          //!< Path to the configuration file.
00078 ABSL_DECLARE_FLAG(std::string, config_file); //!< Path to the configuration file.
00079
00080 ABSL_DECLARE_FLAG(std::string, log_dir);    //!< Path to the folder in which logging will be stored.
00081
00082 // Declare flags for signal sources
00083 ABSL_DECLARE_FLAG(std::string, s);          //!< Path to the file containing the signal
                                samples.
00084 ABSL_DECLARE_FLAG(std::string, signal_source); //!< Path to the file containing the signal
                                samples.
00085 ABSL_DECLARE_FLAG(std::string, timestamp_source); //!< Path to the file containing the signal
                                samples.
00086 ABSL_DECLARE_FLAG(bool, rf_shutdown);       //!< Shutdown RF when program exits.
00087
00088 // Declare flags for acquisition blocks
00089 ABSL_DECLARE_FLAG(int32_t, doppler_max);   //!< If defined, maximum Doppler value in the search grid,
                                in Hz (overrides the configuration file).
00090 ABSL_DECLARE_FLAG(int32_t, doppler_step);  //!< If defined, sets the frequency step in the search
                                grid, in Hz, in Hz (overrides the configuration file).
00091
00092 // Declare flags for tracking blocks
00093 ABSL_DECLARE_FLAG(int32_t, cn0_samples);   //!< Number of correlator outputs used for CN0
                                estimation.
00094 ABSL_DECLARE_FLAG(int32_t, cn0_min);       //!< Minimum valid CN0 (in dB-Hz).
00095 ABSL_DECLARE_FLAG(int32_t, max_lock_fail); //!< Maximum number of code lock failures before
                                dropping a satellite.
00096 ABSL_DECLARE_FLAG(int32_t, max_carrier_lock_fail); //!< Maximum number of carrier lock failures
                                before dropping a satellite.
00097 ABSL_DECLARE_FLAG(double, carrier_lock_th); //!< Carrier lock threshold (in rad).
00098 ABSL_DECLARE_FLAG(double, dll_bw_hz);      //!< Bandwidth of the DLL low pass filter, in Hz

```

```

    (overrides the configuration file).
00099 ABSL_DECLARE_FLAG(double, pll_bw_hz);           //!< Bandwidth of the PLL low pass filter, in Hz
    (overrides the configuration file).
00100
00101 // Declare flags for observables block
00102 ABSL_DECLARE_FLAG(int32_t, carrier_smoothing_factor); //!< Sets carrier smoothing factor M (overrides
the configuration file).
00103 const int32_t DEFAULT_CARRIER_SMOOTHING_FACTOR = 200;
00104
00105 // Declare flags for PVT
00106 ABSL_DECLARE_FLAG(std::string, RINEX_version);      //!< If defined, specifies the RINEX version (2.11 or
3.02). Overrides the configuration file.
00107 ABSL_DECLARE_FLAG(std::string, RINEX_name);         //!< If defined, specifies the RINEX files base name
00108 ABSL_DECLARE_FLAG(bool, keyboard);                  //!< If set to false, disables the keyboard listener.
    Only for debug purposes (e.g. ASAN mode termination)
00109
00110 static inline void GetTempDirectories(std::vector<std::string>& list)
00111 {
00112     list.clear();
00113     // Directories, in order of preference. If we find a dir that
00114     // exists, we stop adding other less-preferred dirs
00115     const char* candidates[] = {
00116         // Non-null only during unittest/regtest
00117         std::getenv("TEST_TMPDIR"),
00118
00119         // Explicitly-supplied temp dirs
00120         std::getenv("TMPDIR"),
00121         std::getenv("TMP"),
00122
00123         // If all else fails
00124         "/tmp",
00125     };
00126     for (auto d : candidates)
00127     {
00128         if (!d) continue; // Empty env var
00129
00130         // Make sure we don't surprise anyone who's expecting a '/'
00131         std::string dstr = d;
00132         if (dstr[dstr.size() - 1] != '/')
00133         {
00134             dstr += "/";
00135         }
00136         list.push_back(dstr);
00137
00138         struct stat statbuf;
00139         if (!stat(d, &statbuf) && S_ISDIR(statbuf.st_mode))
00140         {
00141             // We found a dir that exists - we're done.
00142             return;
00143         }
00144     }
00145 }
00146
00147
00148 static inline void GetExistingTempDirectories(std::vector<std::string>& list)
00149 {
00150     GetTempDirectories(list);
00151     auto i_dir = list.begin();
00152     while (i_dir != list.end())
00153     {
00154         if (access(i_dir->c_str(), 0))
00155         {
00156             i_dir = list.erase(i_dir);
00157         }
00158         else
00159         {
00160             ++i_dir;
00161         }
00162     };
00163 }
00164
00165
00166 static inline std::string GetTempDir()
00167 {
00168     std::vector<std::string> temp_directories_list;
00169     GetExistingTempDirectories(temp_directories_list);
00170
00171     if (temp_directories_list.empty())
00172     {
00173         std::cerr << "No temporary directory found\n";
00174         exit(EXIT_FAILURE);
00175     }
00176
00177     // Use first directory from list of existing temporary directories.
00178     return temp_directories_list.front();
00179 }
00180

```

```

00181 bool ValidateFlags();
00182
00183 #endif
00184
00185 /** \} */
00186 /** \} */
00187 #endif // GNSS_SDR_GNSS_SDR_FLAGS_H

```

## 11.172 gnss\_sdr\_make\_unique.h File Reference

This file implements std::make\_unique for C++11.

This graph shows which files directly or indirectly include this file:



### 11.172.1 Detailed Description

This file implements std::make\_unique for C++11.

#### Author

Carles Fernandez-Prades, 2020. cfernandez(at)cttc.es

Based on <https://stackoverflow.com/a/17902439>

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [gnss\\_sdr\\_make\\_unique.h](#).

## 11.173 gnss\_sdr\_make\_unique.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file gnss_sdr_make_unique.h
00003  * \brief This file implements std::make_unique for C++11
00004  *
00005  * \author Carles Fernandez-Prades, 2020. cfernandez(at)cttc.es
00006  *
00007  * Based on https://stackoverflow.com/a/17902439
00008  *
00009  *
00010  * -----
00011  *
00012  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00013  * This file is part of GNSS-SDR.
00014  *
00015  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00016  * SPDX-License-Identifier: GPL-3.0-or-later
00017  *
00018  * -----
00019  */
00020
00021 #ifndef GNSS_SDR_GNSS_SDR_MAKE_UNIQUE_H
00022 #define GNSS_SDR_GNSS_SDR_MAKE_UNIQUE_H
00023
00024 #if __cplusplus == 201103L
00025
00026 #include <cstdint>
00027 #include <memory>
00028 #include <type_traits>
00029 #include <utility>
00030
00031 /** \addtogroup Algorithms_Library
00032  * \{ */
00033 /** \addtogroup Algorithm_libs algorithms_libs
00034  * \{ */
00035
00036 namespace std
00037 {
00038     template <class T>
00039     struct _Unique_if

```

```

00041 {
00042     typedef unique_ptr<T> _Single_object;
00043 };
00044
00045 template <class T>
00046 struct _Unique_if<T[]>
00047 {
00048     typedef unique_ptr<T[]> _Unknown_bound;
00049 };
00050
00051 template <class T, size_t N>
00052 struct _Unique_if<T[N]>
00053 {
00054     typedef void _Known_bound;
00055 };
00056
00057 template <class T, class... Args>
00058 typename _Unique_if<T>::_Single_object
00059 make_unique(Args&&... args)
00060 {
00061     return unique_ptr<T>(new T(std::forward<Args>(args)...));
00062 }
00063
00064 template <class T>
00065 typename _Unique_if<T>::_Unknown_bound
00066 make_unique(size_t n)
00067 {
00068     typedef typename remove_extent<T>::type U;
00069     return unique_ptr<T>(new U[n]());
00070 }
00071
00072 template <class T, class... Args>
00073 typename _Unique_if<T>::_Known_bound
00074 make_unique(Args&&...) = delete;
00075 } // namespace std
00076
00077 #endif // __cplusplus == 201103L
00078
00079 /** \} */
00080 /** \} */
00081 /** \} */
00082 #endif // GNSS_SDR_GNSS_SDR_MAKE_UNIQUE_H

```

## 11.174 gnss\_sdr\_string\_literals.h File Reference

This file implements the `""s` operator for `std::string` in C++11, and puts it into the `std::string_literals` namespace. This is already implemented in C++14, so this is only compiled when using C++11. The `.cc` file is required for avoiding the duplication of symbols.

### 11.174.1 Detailed Description

This file implements the `""s` operator for `std::string` in C++11, and puts it into the `std::string_literals` namespace. This is already implemented in C++14, so this is only compiled when using C++11. The `.cc` file is required for avoiding the duplication of symbols.

#### Author

Carles Fernandez-Prades, 2021. cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [gnss\\_sdr\\_string\\_literals.h](#).

---

## 11.175 gnss\_sdr\_string\_literals.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file gnss_sdr_string_literals.h
00003  * \brief This file implements the ""s operator for std::string in C++11, and
00004  * puts it into the std::string_literals namespace. This is already implemented
00005  * in C++14, so this is only compiled when using C++11. The .cc file is required
00006  * for avoiding the duplication of symbols.
00007  *
00008  * \author Carles Fernandez-Prades, 2021. cfernandez(at)cttc.es
00009  *

```

```

00010  *
00011  * -----
00012  *
00013  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00014  * This file is part of GNSS-SDR.
00015  *
00016  * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)
00017  * SPDX-License-Identifier: GPL-3.0-or-later
00018  *
00019  * -----
00020  */
00021
00022 #ifndef GNSS_SDR_STRING_LITERALS_H
00023 #define GNSS_SDR_STRING_LITERALS_H
00024
00025 /** \addtogroup Algorithms_Library
00026  * \{ */
00027 /** \addtogroup Algorithm_libs algorithms_libs
00028  * \{ */
00029
00030 #if __cplusplus == 201103L
00031
00032 #include <cstdlib>
00033 #include <string>
00034
00035 namespace std
00036 {
00037     namespace string_literals
00038     {
00039         std::string operator"" s(const char* str, std::size_t len);
00040     } // namespace string_literals
00041 } // namespace std
00042
00043 #endif // __cplusplus == 201103L
00044
00045 /** \} */
00046 /** \} */
00047
00048 #endif // GNSS_SDR_STRING_LITERALS_H

```

## 11.176 gnss\_signal\_replica.h File Reference

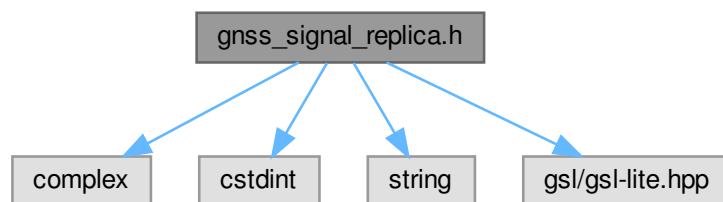
This library gathers a few functions used for GNSS signal replica generation regardless of system used.

```

#include <complex>
#include <stdint>
#include <string>
#include <gsl/gsl-lite.hpp>

```

Include dependency graph for gnss\_signal\_replica.h:



### Functions

- void [complex\\_exp\\_gen](#) (own::span< std::complex< float > > dest, double freq, double sampling\_freq)  
*This function generates a complex exponential in dest.*
- void [complex\\_exp\\_gen\\_conj](#) (own::span< std::complex< float > > dest, double freq, double sampling\_freq)  
*This function generates a conjugate complex exponential in dest.*

- void [hex\\_to\\_binary\\_converter](#) (own::span< int32\_t > dest, char from)  
*This function makes a conversion from hex (the input is a char) to binary (the output are 4 ints with +1 or -1 values).*
- std::string [hex\\_to\\_binary\\_string](#) (char from)  
*This function makes a conversion from hex (the input is a char) to binary (the output is a string of 4 char with 0 or 1 values).*
- void [resampler](#) (const own::span< float > from, own::span< float > dest, float fs\_in, float fs\_out)  
*This function resamples a sequence of float values.*
- void [resampler](#) (own::span< const std::complex< float > > from, own::span< std::complex< float > > dest, float fs\_in, float fs\_out)  
*This function resamples a sequence of complex values.*

### 11.176.1 Detailed Description

This library gathers a few functions used for GNSS signal replica generation regardless of system used.

#### Author

Luis Esteve, 2012. [luis\(at\)epsilon-formacion.com](mailto:luis(at)epsilon-formacion.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [gnss\\_signal\\_replica.h](#).

---

## 11.177 gnss\_signal\_replica.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file gnss_signal_replica.h
00003  * \brief This library gathers a few functions used for GNSS signal replica
00004  * \generation regardless of system used
00005  * \author Luis Esteve, 2012. luis\(at\)epsilon-formacion.com
00006  *
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019 #ifndef GNSS_SDR_GNSS_SIGNAL_REPLICA_H
00020 #define GNSS_SDR_GNSS_SIGNAL_REPLICA_H
00021
00022 #include <complex>
00023 #include <cstdint>
00024 #include <string>
00025 #if HAS_STD_SPAN
00026 #include <span>
00027 namespace own = std;
00028 #else
00029 #include <gsl/gsl-lite.hpp>
00030 namespace own = gsl;
00031 #endif
00032
00033 /** \addtogroup Algorithms_Library
00034  * \{ */
00035 /** \addtogroup Algorithm_libs algorithms_libs
00036  * \{ */
00037
00038
00039 /*!
00040  * \brief This function generates a complex exponential in dest.
00041  *
00042  */
00043 void complex\_exp\_gen(own::span<std::complex<float>> dest, double freq, double sampling_freq);
00044
00045 /*!
00046  * \brief This function generates a conjugate complex exponential in dest.
00047  *
00048  */

```



```

00049 void complex_exp_gen_conj(own::span<std::complex<float>> dest, double freq, double sampling_freq);
00050
00051 /*!
00052 * \brief This function makes a conversion from hex (the input is a char)
00053 * to binary (the output are 4 ints with +1 or -1 values).
00054 *
00055 */
00056 void hex_to_binary_converter(own::span<int32_t> dest, char from);
00057
00058 /*!
00059 * \brief This function makes a conversion from hex (the input is a char)
00060 * to binary (the output is a string of 4 char with 0 or 1 values).
00061 *
00062 */
00063 std::string hex_to_binary_string(char from);
00064
00065 /*!
00066 * \brief This function resamples a sequence of float values.
00067 *
00068 */
00069 void resampler(const own::span<float> from, own::span<float> dest,
00070               float fs_in, float fs_out);
00071
00072 /*!
00073 * \brief This function resamples a sequence of complex values.
00074 *
00075 */
00076 void resampler(own::span<const std::complex<float>> from, own::span<std::complex<float>> dest,
00077               float fs_in, float fs_out);
00078
00079
00080 /** \} */
00081 /** \} */
00082 #endif // GNSS_SDR_GNSS_SIGNAL_REPLICA_H

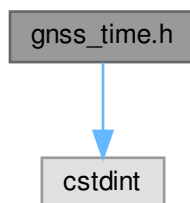
```

## 11.178 gnss\_time.h File Reference

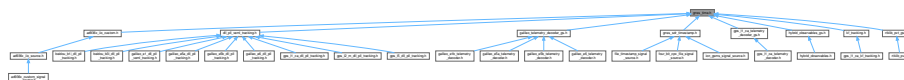
class that stores both the receiver time, relative to the receiver start and the GNSS time (absolute)

#include <cstdint>

Include dependency graph for gnss\_time.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [GnssTime](#)

### 11.178.1 Detailed Description

class that stores both the receiver time, relative to the receiver start and the GNSS time (absolute)

Author

Javier Arribas 2022. jarribas(at)cttc.es

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [gnss\\_time.h](#).

## 11.179 gnss\_time.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file gnss_time.h
00003  * \brief class that stores both the receiver time, relative to the receiver start and the GNSS time
00004  *        (absolute)
00005  * \author Javier Arribas 2022. jarribas(at)cttc.es
00006  *
00007  * -----
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_GNSS_TIME_H
00019 #define GNSS_SDR_GNSS_TIME_H
00020
00021 #include <stdint>
00022
00023 class GnssTime
00024 {
00025 public:
00026     double rx_time;
00027     int week;           /*!< GPS week number (since January 1980) */
00028     int tow_ms;         /* time of week [ms]*/
00029     double tow_ms_fraction; /* tow ms fractional part [ms]*/
00030 };
00031
00032 #endif

```

## 11.180 gps\_l2c\_signal\_replica.h File Reference

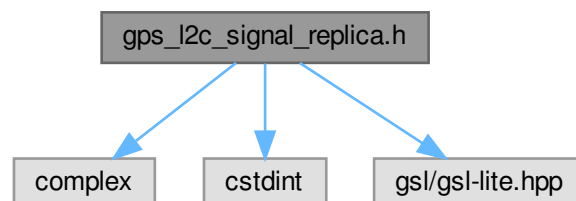
This file implements signal generators for GPS L2C signals.

```
#include <complex>
```

```
#include <stdint>
```

```
#include <gsl/gsl-lite.hpp>
```

Include dependency graph for `gps_l2c_signal_replica.h`:



## Functions

- void [gps\\_l2c\\_m\\_code\\_gen\\_complex](#) (own::span< std::complex< float > > dest, uint32\_t prn)  
*Generates complex GPS L2C M code for the desired SV ID.*
- void [gps\\_l2c\\_m\\_code\\_gen\\_float](#) (own::span< float > dest, uint32\_t prn)  
*Generates float GPS L2C M code for the desired SV ID.*
- void [gps\\_l2c\\_m\\_code\\_gen\\_complex\\_sampled](#) (own::span< std::complex< float > > dest, uint32\_t prn, int32\_t sampling\_freq)  
*Generates complex GPS L2C M code for the desired SV ID, and sampled to specific sampling frequency.*

### 11.180.1 Detailed Description

This file implements signal generators for GPS L2C signals.

#### Author

Javier Arribas, 2015. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [gps\\_l2c\\_signal\\_replica.h](#).

---

## 11.181 gps\_l2c\_signal\_replica.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002   * \file gps_l2c_signal_replica.h
00003   * \brief This file implements signal generators for GPS L2C signals
00004   * \author Javier Arribas, 2015. jarribas(at)cttc.es
00005   *
00006   *
00007   * -----
00008   *
00009   * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010   * This file is part of GNSS-SDR.
00011   *
00012   * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013   * SPDX-License-Identifier: GPL-3.0-or-later
00014   *
00015   * -----
00016   */
00017
00018 #ifndef GNSS_SDR_GPS_L2C_SIGNAL_REPLICA_H
00019 #define GNSS_SDR_GPS_L2C_SIGNAL_REPLICA_H
00020
00021 #include <complex>
00022 #include <stdint>
00023 #if HAS_STD_SPAN
00024 #include <span>
00025 namespace own = std;
00026 #else
00027 #include <gsl/gsl-lite.hpp>
00028 namespace own = gsl;
00029 #endif
00030
00031 /** \addtogroup Algorithms_Library
00032  * \{ */
00033 /** \addtogroup Algorithm_libs algorithms_libs
00034  * \{ */
00035
00036
00037 //! Generates complex GPS L2C M code for the desired SV ID
00038 void gps\_l2c\_m\_code\_gen\_complex(own::span<std::complex<float>> dest, uint32_t prn);
00039
00040 //! Generates float GPS L2C M code for the desired SV ID
00041 void gps\_l2c\_m\_code\_gen\_float(own::span<float> dest, uint32_t prn);
00042
00043 //! Generates complex GPS L2C M code for the desired SV ID, and sampled to specific sampling frequency
00044 void gps\_l2c\_m\_code\_gen\_complex\_sampled(own::span<std::complex<float>> dest, uint32_t prn, int32_t
    sampling_freq);
00045
00046
00047 /** \} */
00048 /** \} */
00049 #endif // GNSS_SDR_GPS_L2C_SIGNAL_REPLICA_H

```

## 11.182 gps\_l5\_signal\_replica.h File Reference

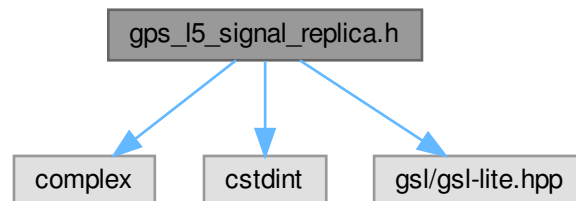
This file implements signal generators for GPS L5 signals.

```
#include <complex>
```

```
#include <cstdint>
```

```
#include <gsl/gsl-lite.hpp>
```

Include dependency graph for `gps_l5_signal_replica.h`:



### Functions

- void [gps\\_l5i\\_code\\_gen\\_complex](#) (own::span< std::complex< float > > dest, uint32\_t prn)  
*Generates complex GPS L5I code for the desired SV ID.*
- void [gps\\_l5i\\_code\\_gen\\_float](#) (own::span< float > dest, uint32\_t prn)  
*Generates real GPS L5I code for the desired SV ID.*
- void [gps\\_l5q\\_code\\_gen\\_complex](#) (own::span< std::complex< float > > dest, uint32\_t prn)  
*Generates complex GPS L5Q code for the desired SV ID.*
- void [gps\\_l5q\\_code\\_gen\\_float](#) (own::span< float > dest, uint32\_t prn)  
*Generates real GPS L5Q code for the desired SV ID.*
- void [gps\\_l5i\\_code\\_gen\\_complex\\_sampled](#) (own::span< std::complex< float > > dest, uint32\_t prn, int32\_t sampling\_freq)  
*Generates complex GPS L5I code for the desired SV ID, and sampled to specific sampling frequency.*
- void [gps\\_l5q\\_code\\_gen\\_complex\\_sampled](#) (own::span< std::complex< float > > dest, uint32\_t prn, int32\_t sampling\_freq)  
*Generates complex GPS L5Q code for the desired SV ID, and sampled to specific sampling frequency.*

### 11.182.1 Detailed Description

This file implements signal generators for GPS L5 signals.

#### Author

Javier Arribas, 2017. [jarribas\(at\)cttc.es](mailto:jarribas(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [gps\\_l5\\_signal\\_replica.h](#).

---

## 11.183 gps\_l5\_signal\_replica.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file gps_l5_signal_replica.h
00003  * \brief This file implements signal generators for GPS L5 signals
00004  * \author Javier Arribas, 2017. jarribas\(at\)cttc.es

```

```

00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_GPS_L5_SIGNAL_REPLICA_H
00019 #define GNSS_SDR_GPS_L5_SIGNAL_REPLICA_H
00020
00021 #include <complex>
00022 #include <stdint>
00023 #if HAS_STD_SPAN
00024 #include <span>
00025 namespace own = std;
00026 #else
00027 #include <gsl/gsl-lite.hpp>
00028 namespace own = gsl;
00029 #endif
00030
00031 /** \addtogroup Algorithms_Library
00032  * \{ */
00033 /** \addtogroup Algorithm_libs algorithms_libs
00034  * \{ */
00035
00036
00037 //! Generates complex GPS L5I code for the desired SV ID
00038 void gps_l5i_code_gen_complex(own::span<std::complex<float>> dest, uint32_t prn);
00039
00040 //! Generates real GPS L5I code for the desired SV ID
00041 void gps_l5i_code_gen_float(own::span<float> dest, uint32_t prn);
00042
00043 //! Generates complex GPS L5Q code for the desired SV ID
00044 void gps_l5q_code_gen_complex(own::span<std::complex<float>> dest, uint32_t prn);
00045
00046 //! Generates real GPS L5Q code for the desired SV ID
00047 void gps_l5q_code_gen_float(own::span<float> dest, uint32_t prn);
00048
00049 //! Generates complex GPS L5I code for the desired SV ID, and sampled to specific sampling frequency
00050 void gps_l5i_code_gen_complex_sampled(own::span<std::complex<float>> dest, uint32_t prn, int32_t
    sampling_freq);
00051
00052 //! Generates complex GPS L5Q code for the desired SV ID, and sampled to specific sampling frequency
00053 void gps_l5q_code_gen_complex_sampled(own::span<std::complex<float>> dest, uint32_t prn, int32_t
    sampling_freq);
00054
00055
00056 /** \} */
00057 /** \} */
00058 #endif // GNSS_SDR_GPS_L5_SIGNAL_REPLICA_H

```

## 11.184 gps\_sdr\_signal\_replica.h File Reference

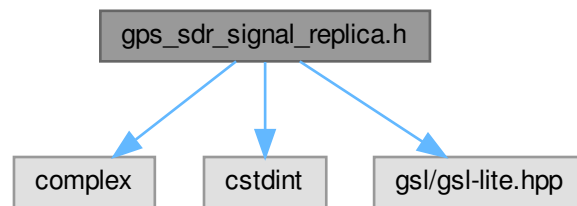
This file implements functions for GPS L1 C/A signal replica generation.

```

#include <complex>
#include <stdint>
#include <gsl/gsl-lite.hpp>

```

Include dependency graph for `gps_sdr_signal_replica.h`:



## Functions

- void `gps_l1_ca_code_gen_int` (own::span< int32\_t > dest, int32\_t prn, uint32\_t chip\_shift)  
*Generates int GPS L1 C/A code for the desired SV ID and code shift.*
- void `gps_l1_ca_code_gen_float` (own::span< float > dest, int32\_t prn, uint32\_t chip\_shift)  
*Generates float GPS L1 C/A code for the desired SV ID and code shift.*
- void `gps_l1_ca_code_gen_complex` (own::span< std::complex< float > > dest, int32\_t prn, uint32\_t chip\_shift)  
*Generates complex GPS L1 C/A code for the desired SV ID and code shift.*
- void `gps_l1_ca_code_gen_complex_sampled` (own::span< std::complex< float > > dest, uint32\_t prn, int32\_t sampling\_freq, uint32\_t chip\_shift)  
*Generates complex GPS L1 C/A code for the desired SV ID and code shift, and sampled to specific sampling frequency.*

### 11.184.1 Detailed Description

This file implements functions for GPS L1 C/A signal replica generation.

#### Author

Javier Arribas, 2011. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [gps\\_sdr\\_signal\\_replica.h](#).

---

### 11.185 `gps_sdr_signal_replica.h`

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file gps_sdr_signal_replica.h
00003  * \brief This file implements functions for GPS L1 C/A signal replica
00004  * generation
00005  * \author Javier Arribas, 2011. jarribas(at)cttc.es
00006  *
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018

```

```

00019 #ifndef GNSS_SDR_GPS_SDR_SIGNAL_REPLICA_H
00020 #define GNSS_SDR_GPS_SDR_SIGNAL_REPLICA_H
00021
00022 #include <complex>
00023 #include <stdint>
00024 #if HAS_STD_SPAN
00025 #include <span>
00026 namespace own = std;
00027 #else
00028 #include <gsl/gsl-lite.hpp>
00029 namespace own = gsl;
00030 #endif
00031
00032 /** \addtogroup Algorithms_Library
00033  * \{ */
00034 /** \addtogroup Algorithm_libs algorithms_libs
00035  * \{ */
00036
00037
00038 /// Generates int GPS L1 C/A code for the desired SV ID and code shift
00039 void gps_l1_ca_code_gen_int(own::span<int32_t> dest, int32_t prn, uint32_t chip_shift);
00040
00041 /// Generates float GPS L1 C/A code for the desired SV ID and code shift
00042 void gps_l1_ca_code_gen_float(own::span<float> dest, int32_t prn, uint32_t chip_shift);
00043
00044 /// Generates complex GPS L1 C/A code for the desired SV ID and code shift
00045 void gps_l1_ca_code_gen_complex(own::span<std::complex<float>> dest, int32_t prn, uint32_t chip_shift);
00046
00047 /// Generates complex GPS L1 C/A code for the desired SV ID and code shift, and sampled to specific
sampling frequency
00048 void gps_l1_ca_code_gen_complex_sampled(own::span<std::complex<float>> dest, uint32_t prn, int32_t
sampling_freq, uint32_t chip_shift);
00049
00050
00051 /** \} */
00052 /** \} */
00053 #endif // GNSS_SDR_GPS_SDR_SIGNAL_REPLICA_H

```

## 11.186 item\_type\_helpers.h File Reference

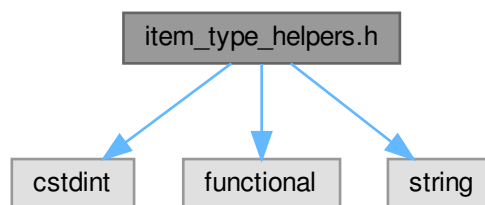
Utility functions for converting between item types.

```

#include <stdint>
#include <functional>
#include <string>

```

Include dependency graph for item\_type\_helpers.h:



### Typedefs

- using [item\\_type\\_converter\\_t](#) = std::function<void(void \*, const void \*, uint32\_t)>

### Functions

- bool [item\\_type\\_valid](#) (const std::string &item\_type)  
*Check if a string is a valid item type.*
- size\_t [item\\_type\\_size](#) (const std::string &item\_type)

*Return the size of the given item type, or zero if unknown.*

- bool `item_type_is_complex` (const std::string &item\_type)

*Determine if an item\_type is complex.*

- item\_type\_converter\_t `make_vector_converter` (const std::string &input\_type, const std::string &output\_type)

*Create a function to convert an array of input\_type to an array of output\_type.*

### 11.186.1 Detailed Description

Utility functions for converting between item types.

#### Authors

- Cillian O'Driscoll, 2019. cillian.odriscoll(at)gmail.com

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [item\\_type\\_helpers.h](#).

---

## 11.187 item\_type\_helpers.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file item_type_helpers.h
00003  * \brief Utility functions for converting between item types
00004  * \authors <ul>
00005  *         <li> Cillian O'Driscoll, 2019. cillian.odriscoll(at)gmail.com
00006  *         </li>
00007  *
00008  *
00009  * -----
00010  *
00011  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00012  * This file is part of GNSS-SDR.
00013  *
00014  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00015  * SPDX-License-Identifier: GPL-3.0-or-later
00016  *
00017  * -----
00018  */
00019
00020 #ifndef GNSS_SDR_ITEM_TYPE_HELPERS_H
00021 #define GNSS_SDR_ITEM_TYPE_HELPERS_H
00022
00023
00024 #include <stdint>
00025 #include <functional>
00026 #include <string>
00027
00028 /** \addtogroup Algorithms_Library
00029  * \{ */
00030 /** \addtogroup Algorithm_libs algorithms_libs
00031  * \{ */
00032
00033
00034 using item_type_converter_t = std::function<void(void *, const void *, uint32_t)>;
00035
00036 /*!
00037  * \brief Check if a string is a valid item type
00038  *
00039  * \description Valid item types include:
00040  *         "byte", "short", "float", "ibyte", "ishort", "cbyte", "cshort", "gr_complex"
00041  *
00042  */
00043 bool item_type_valid(const std::string &item_type);
00044
00045 /*!
00046  * \brief Return the size of the given item type, or zero if unknown
00047  */
00048 size_t item_type_size(const std::string &item_type);
00049
00050 /*!
00051  * \brief Determine if an item_type is complex
00052  */
00053 bool item_type_is_complex(const std::string &item_type);
00054
00055 /*!
00056  * \brief Create a function to convert an array of input_type to an array of output_type
```



```

00057 *
00058 * \description Provides a generic interface to generate conversion functions for mapping
00059 * arrays of items.
00060 *
00061 * \param input_type - String representation of the input item type
00062 * \param output_type - String representation of the output item type
00063 *
00064 * The item types accepted are:
00065 *
00066 * 1. "byte" for 8 bit integers
00067 * 2. "cbyte" for complex (interleaved) 8 bit integers
00068 * 4. "ibyte" for complex (interleaved) 8 bit integers
00069 * 4. "short" for 16 bit integers
00070 * 5. "cshort" for complex (interleaved) 16 bit integers
00071 * 6. "ishort" for complex (interleaved) 16 bit integers
00072 * 7. "float" for 32 bit floating point values
00073 * 8. "gr_complex" for complex (interleaved) 32 bit floating point values
00074 *
00075 * \returns A function object with the following prototype:
00076 * void convert_fun( void *dest, void *src, int num_items );
00077 *
00078 */
00079 item_type_converter_t make_vector_converter(const std::string &input_type,
00080      const std::string &output_type);
00081
00082
00083 /** \} */
00084 /** \} */
00085 #endif // GNSS_SDR_ITEM_TYPE_HELPERS_H

```

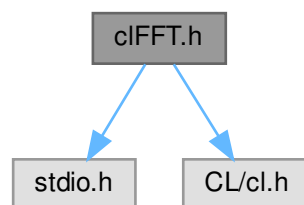
## 11.188 cIFFT.h File Reference

FFT in OpenCL.

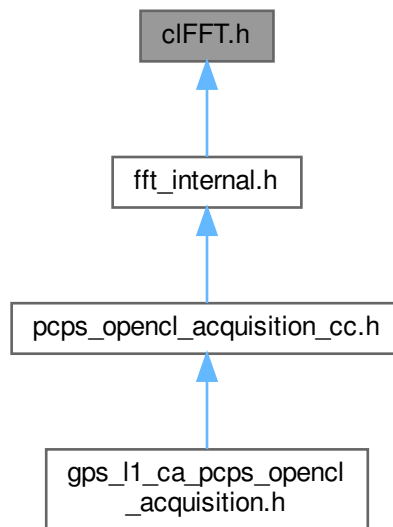
```
#include <stdio.h>
```

```
#include <CL/cl.h>
```

Include dependency graph for cIFFT.h:



This graph shows which files directly or indirectly include this file:



## Classes

- struct [clFFT\\_Dim3](#)
- struct [clFFT\\_SplitComplex](#)
- struct [clFFT\\_Complex](#)

## Typedefs

- typedef void \* [clFFT\\_Plan](#)

## Enumerations

- enum [clFFT\\_Direction](#) { [clFFT\\_Forward](#) = -1 , [clFFT\\_Inverse](#) = 1 }
- enum [clFFT\\_Dimension](#) { [clFFT\\_1D](#) = 0 , [clFFT\\_2D](#) = 1 , [clFFT\\_3D](#) = 3 }
- enum [clFFT\\_DataFormat](#) { [clFFT\\_SplitComplexFormat](#) = 0 , [clFFT\\_InterleavedComplexFormat](#) = 1 }

## Functions

- [clFFT\\_Plan](#) **clFFT\_CreatePlan** (cl\_context context, [clFFT\\_Dim3](#) n, [clFFT\\_Dimension](#) dim, [clFFT\\_DataFormat](#) dataFormat, cl\_int \*error\_code)
- void **clFFT\_DestroyPlan** ([clFFT\\_Plan](#) plan)
- cl\_int **clFFT\_ExecuteInterleaved** (cl\_command\_queue queue, [clFFT\\_Plan](#) plan, cl\_int batchSize, [clFFT\\_Direction](#) dir, cl\_mem data\_in, cl\_mem data\_out, cl\_int num\_events, cl\_event \*event\_list, cl\_event \*event)
- cl\_int **clFFT\_ExecutePlannar** (cl\_command\_queue queue, [clFFT\\_Plan](#) plan, cl\_int batchSize, [clFFT\\_Direction](#) dir, cl\_mem data\_in\_real, cl\_mem data\_in\_imag, cl\_mem data\_out\_real, cl\_mem data\_out\_imag, cl\_int num\_events, cl\_event \*event\_list, cl\_event \*event)
- cl\_int **clFFT\_1DTwistInterleaved** ([clFFT\\_Plan](#) Plan, cl\_command\_queue queue, cl\_mem array, size\_t numRows, size\_t numCols, size\_t startRow, size\_t rowsToProcess, [clFFT\\_Direction](#) dir)
- cl\_int **clFFT\_1DTwistPlannar** ([clFFT\\_Plan](#) Plan, cl\_command\_queue queue, cl\_mem array\_real, cl\_mem array\_imag, size\_t numRows, size\_t numCols, size\_t startRow, size\_t rowsToProcess, [clFFT\\_Direction](#) dir)
- void **clFFT\_DumpPlan** ([clFFT\\_Plan](#) plan, FILE \*file)

### 11.188.1 Detailed Description

FFT in OpenCL.

Version: <1.0>

Copyright ( C ) 2008 Apple Inc. All Rights Reserved. SPDX-License-Identifier: LicenseRef-Apple-Permissive

Definition in file [clFFT.h](#).

### 11.188.2 Typedef Documentation

#### 11.188.2.1 clFFT\_Plan

```
typedef void* clFFT_Plan
```

Definition at line 71 of file [clFFT.h](#).

### 11.188.3 Enumeration Type Documentation

#### 11.188.3.1 clFFT\_DataFormat

```
enum clFFT_DataFormat
```

Definition at line 46 of file [clFFT.h](#).

#### 11.188.3.2 clFFT\_Dimension

```
enum clFFT_Dimension
```

Definition at line 38 of file [clFFT.h](#).

#### 11.188.3.3 clFFT\_Direction

```
enum clFFT_Direction
```

Definition at line 31 of file [clFFT.h](#).

## 11.189 clFFT.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file clFFT.h
00003  * \brief FFT in OpenCL
00004  *
00005  *
00006  * Version:    <1.0>
00007  *
00008  * Copyright ( C ) 2008 Apple Inc. All Rights Reserved.
00009  * SPDX-License-Identifier: LicenseRef-Apple-Permissive
00010  *
00011  */
00012
00013 #ifndef __CLFFT_H
00014 #define __CLFFT_H
00015
00016 #ifdef __cplusplus
00017 extern "C"
00018 {
00019 #endif
00020
00021 #include <stdio.h>
00022
00023 #ifdef __APPLE__
00024 #define CL_SILENCE_DEPRECATION
00025 #include <OpenCL/opencl.h>
00026 #else
00027 #include <CL/cl.h>
00028 #endif
00029
00030 // XForm type
00031 typedef enum
00032 {
00033     clFFT_Forward = -1,
00034     clFFT_Inverse = 1
00035 } clFFT_Direction;
00036
00037 // XForm dimension
```

```

00038     typedef enum
00039     {
00040         clFFT_1D = 0,
00041         clFFT_2D = 1,
00042         clFFT_3D = 3
00043     } clFFT_Dimension;
00044
00045     // XForm Data type
00046     typedef enum
00047     {
00048         clFFT_SplitComplexFormat = 0,
00049         clFFT_InterleavedComplexFormat = 1
00050     } clFFT_DataFormat;
00051
00052     typedef struct
00053     {
00054         unsigned int x;
00055         unsigned int y;
00056         unsigned int z;
00057     } clFFT_Dim3;
00058
00059     typedef struct
00060     {
00061         float *real;
00062         float *imag;
00063     } clFFT_SplitComplex;
00064
00065     typedef struct
00066     {
00067         float real;
00068         float imag;
00069     } clFFT_Complex;
00070
00071     typedef void *clFFT_Plan;
00072
00073     clFFT_Plan clFFT_CreatePlan(cl_context context, clFFT_Dim3 n, clFFT_Dimension dim,
00074                                clFFT_DataFormat dataFormat, cl_int *error_code);
00075
00076     void clFFT_DestroyPlan(clFFT_Plan plan);
00077
00078     cl_int clFFT_ExecuteInterleaved(cl_command_queue queue, clFFT_Plan plan, cl_int batchSize,
00079                                     clFFT_Direction dir,
00080                                     cl_mem data_in, cl_mem data_out,
00081                                     cl_int num_events, cl_event *event_list, cl_event *event);
00082
00083     cl_int clFFT_ExecutePlannar(cl_command_queue queue, clFFT_Plan plan, cl_int batchSize,
00084                                 clFFT_Direction dir,
00085                                 cl_mem data_in_real, cl_mem data_in_imag, cl_mem data_out_real, cl_mem data_out_imag,
00086                                 cl_int num_events, cl_event *event_list, cl_event *event);
00087
00088     cl_int clFFT_1DTwistInterleaved(clFFT_Plan Plan, cl_command_queue queue, cl_mem array,
00089                                     size_t numRows, size_t numCols, size_t startRow, size_t rowsToProcess, clFFT_Direction dir);
00090
00091     cl_int clFFT_1DTwistPlannar(clFFT_Plan Plan, cl_command_queue queue, cl_mem array_real, cl_mem
00092                                 array_imag,
00093                                 size_t numRows, size_t numCols, size_t startRow, size_t rowsToProcess, clFFT_Direction dir);
00094
00095     void clFFT_DumpPlan(clFFT_Plan plan, FILE *file);
00096
00097 #ifdef __cplusplus
00098 }
00099 #endif
00100 #endif

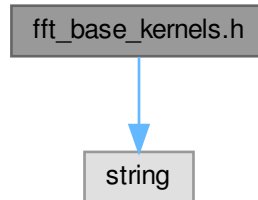
```

## 11.190 fft\_base\_kernels.h File Reference

FFT base kernels for OpenCL.

```
#include <string>
```

Include dependency graph for fft\_base\_kernels.h:



### 11.190.1 Detailed Description

FFT base kernels for OpenCL.

Version: <1.0>

Copyright ( C ) 2008 Apple Inc. All Rights Reserved. SPDX-License-Identifier: LicenseRef-Apple-Permissive

Definition in file [fft\\_base\\_kernels.h](#).

## 11.191 fft\_base\_kernels.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002   * \file fft_base_kernels.h
00003   * \brief FFT base kernels for OpenCL
00004   *
00005   *
00006   * Version:      <1.0>
00007   *
00008   * Copyright ( C ) 2008 Apple Inc. All Rights Reserved.
00009   * SPDX-License-Identifier: LicenseRef-Apple-Permissive
00010   *
00011   */
00012
00013
00014 #ifndef __CL_FFT_BASE_KERNELS_
00015 #define __CL_FFT_BASE_KERNELS_
00016
00017 #include <string>
00018
00019 using namespace std;
00020
00021 static string baseKernels = string(
00022     "#ifndef M_PI\n"
00023     "#define M_PI 0x1.921fb54442d18p+1\n"
00024     "#endif\n"
00025     "#define complexMul(a,b) ((float2)(mad(-(a).y, (b).y, (a).x * (b).x), mad((a).y, (b).x, (a).x * (b).y)))\n"
00026     "#define conj(a) ((float2)((a).x, -(a).y))\n"
00027     "#define conjTransp(a) ((float2)(-(a).y, (a).x))\n"
00028     "\n"
00029     "#define fftKernel2(a,dir) \\\n"
00030     "{ \\\n"
00031     "    float2 c = (a)[0];    \\\n"
00032     "    (a)[0] = c + (a)[1];  \\\n"
00033     "    (a)[1] = c - (a)[1];  \\\n"
00034     "}\n"
00035     "\n"
00036     "#define fftKernel2S(d1,d2,dir) \\\n"
00037     "{ \\\n"
00038     "    float2 c = (d1);    \\\n"
00039     "    (d1) = c + (d2);    \\\n"
00040     "    (d2) = c - (d2);    \\\n"
00041     "}\n"
00042     "\n"
00043     "#define fftKernel4(a,dir) \\\n"

```

```

00044     "{ \\n"
00045     "    fftKernel2S((a)[0], (a)[2], dir); \\n"
00046     "    fftKernel2S((a)[1], (a)[3], dir); \\n"
00047     "    fftKernel2S((a)[0], (a)[1], dir); \\n"
00048     "    (a)[3] = (float2)(dir)*(conjTransp((a)[3])); \\n"
00049     "    fftKernel2S((a)[2], (a)[3], dir); \\n"
00050     "    float2 c = (a)[1]; \\n"
00051     "    (a)[1] = (a)[2]; \\n"
00052     "    (a)[2] = c; \\n"
00053     "}\\n"
00054     "\\n"
00055     "#define fftKernel4s(a0,a1,a2,a3,dir) \\n"
00056     "{ \\n"
00057     "    fftKernel2S((a0), (a2), dir); \\n"
00058     "    fftKernel2S((a1), (a3), dir); \\n"
00059     "    fftKernel2S((a0), (a1), dir); \\n"
00060     "    (a3) = (float2)(dir)*(conjTransp((a3))); \\n"
00061     "    fftKernel2S((a2), (a3), dir); \\n"
00062     "    float2 c = (a1); \\n"
00063     "    (a1) = (a2); \\n"
00064     "    (a2) = c; \\n"
00065     "}\\n"
00066     "\\n"
00067     "#define bitreverse8(a) \\n"
00068     "{ \\n"
00069     "    float2 c; \\n"
00070     "    c = (a)[1]; \\n"
00071     "    (a)[1] = (a)[4]; \\n"
00072     "    (a)[4] = c; \\n"
00073     "    c = (a)[3]; \\n"
00074     "    (a)[3] = (a)[6]; \\n"
00075     "    (a)[6] = c; \\n"
00076     "}\\n"
00077     "\\n"
00078     "#define fftKernel8(a,dir) \\n"
00079     "{ \\n"
00080     "    const float2 w1 = (float2)(0x1.6a09e6p-1f, dir*0x1.6a09e6p-1f); \\n"
00081     "    const float2 w3 = (float2)(-0x1.6a09e6p-1f, dir*0x1.6a09e6p-1f); \\n"
00082     "    float2 c; \\n"
00083     "    fftKernel2S((a)[0], (a)[4], dir); \\n"
00084     "    fftKernel2S((a)[1], (a)[5], dir); \\n"
00085     "    fftKernel2S((a)[2], (a)[6], dir); \\n"
00086     "    fftKernel2S((a)[3], (a)[7], dir); \\n"
00087     "    (a)[5] = complexMul(w1, (a)[5]); \\n"
00088     "    (a)[6] = (float2)(dir)*(conjTransp((a)[6])); \\n"
00089     "    (a)[7] = complexMul(w3, (a)[7]); \\n"
00090     "    fftKernel2S((a)[0], (a)[2], dir); \\n"
00091     "    fftKernel2S((a)[1], (a)[3], dir); \\n"
00092     "    fftKernel2S((a)[4], (a)[6], dir); \\n"
00093     "    fftKernel2S((a)[5], (a)[7], dir); \\n"
00094     "    (a)[3] = (float2)(dir)*(conjTransp((a)[3])); \\n"
00095     "    (a)[7] = (float2)(dir)*(conjTransp((a)[7])); \\n"
00096     "    fftKernel2S((a)[0], (a)[1], dir); \\n"
00097     "    fftKernel2S((a)[2], (a)[3], dir); \\n"
00098     "    fftKernel2S((a)[4], (a)[5], dir); \\n"
00099     "    fftKernel2S((a)[6], (a)[7], dir); \\n"
00100     "    bitreverse8((a)); \\n"
00101     "}\\n"
00102     "\\n"
00103     "#define bitreverse4x4(a) \\n"
00104     "{ \\n"
00105     "    float2 c; \\n"
00106     "    c = (a)[1]; (a)[1] = (a)[4]; (a)[4] = c; \\n"
00107     "    c = (a)[2]; (a)[2] = (a)[8]; (a)[8] = c; \\n"
00108     "    c = (a)[3]; (a)[3] = (a)[12]; (a)[12] = c; \\n"
00109     "    c = (a)[6]; (a)[6] = (a)[9]; (a)[9] = c; \\n"
00110     "    c = (a)[7]; (a)[7] = (a)[13]; (a)[13] = c; \\n"
00111     "    c = (a)[11]; (a)[11] = (a)[14]; (a)[14] = c; \\n"
00112     "}\\n"
00113     "\\n"
00114     "#define fftKernel16(a,dir) \\n"
00115     "{ \\n"
00116     "    const float w0 = 0x1.d906bcp-1f; \\n"
00117     "    const float w1 = 0x1.87de2ap-2f; \\n"
00118     "    const float w2 = 0x1.6a09e6p-1f; \\n"
00119     "    fftKernel4s((a)[0], (a)[4], (a)[8], (a)[12], dir); \\n"
00120     "    fftKernel4s((a)[1], (a)[5], (a)[9], (a)[13], dir); \\n"
00121     "    fftKernel4s((a)[2], (a)[6], (a)[10], (a)[14], dir); \\n"
00122     "    fftKernel4s((a)[3], (a)[7], (a)[11], (a)[15], dir); \\n"
00123     "    (a)[5] = complexMul((a)[5], (float2)(w0, dir*w1)); \\n"
00124     "    (a)[6] = complexMul((a)[6], (float2)(w2, dir*w2)); \\n"
00125     "    (a)[7] = complexMul((a)[7], (float2)(w1, dir*w0)); \\n"
00126     "    (a)[9] = complexMul((a)[9], (float2)(w2, dir*w2)); \\n"
00127     "    (a)[10] = (float2)(dir)*(conjTransp((a)[10])); \\n"
00128     "    (a)[11] = complexMul((a)[11], (float2)(-w2, dir*w2)); \\n"
00129     "    (a)[13] = complexMul((a)[13], (float2)(w1, dir*w0)); \\n"
00130     "    (a)[14] = complexMul((a)[14], (float2)(-w2, dir*w2)); \\n"

```

```

00131     "    (a)[15] = complexMul((a)[15], (float2)(-w0, dir*-w1)); \\n"
00132     "    fftKernel4((a), dir); \\n"
00133     "    fftKernel4((a) + 4, dir); \\n"
00134     "    fftKernel4((a) + 8, dir); \\n"
00135     "    fftKernel4((a) + 12, dir); \\n"
00136     "    bitreverse4x4((a)); \\n"
00137     "}\\n"
00138     "\\n"
00139     "#define bitreverse32(a) \\n"
00140     "{ \\n"
00141     "    float2 c1, c2; \\n"
00142     "    c1 = (a)[2]; (a)[2] = (a)[1]; c2 = (a)[4]; (a)[4] = c1; c1 = (a)[8]; (a)[8] = c2;
c2 = (a)[16]; (a)[16] = c1; (a)[1] = c2; \\n"
00143     "    c1 = (a)[6]; (a)[6] = (a)[3]; c2 = (a)[12]; (a)[12] = c1; c1 = (a)[24]; (a)[24] = c2;
c2 = (a)[17]; (a)[17] = c1; (a)[3] = c2; \\n"
00144     "    c1 = (a)[10]; (a)[10] = (a)[5]; c2 = (a)[20]; (a)[20] = c1; c1 = (a)[9]; (a)[9] = c2;
c2 = (a)[18]; (a)[18] = c1; (a)[5] = c2; \\n"
00145     "    c1 = (a)[14]; (a)[14] = (a)[7]; c2 = (a)[28]; (a)[28] = c1; c1 = (a)[25]; (a)[25] = c2;
c2 = (a)[19]; (a)[19] = c1; (a)[7] = c2; \\n"
00146     "    c1 = (a)[22]; (a)[22] = (a)[11]; c2 = (a)[13]; (a)[13] = c1; c1 = (a)[26]; (a)[26] = c2;
c2 = (a)[21]; (a)[21] = c1; (a)[11] = c2; \\n"
00147     "    c1 = (a)[30]; (a)[30] = (a)[15]; c2 = (a)[29]; (a)[29] = c1; c1 = (a)[27]; (a)[27] = c2;
c2 = (a)[23]; (a)[23] = c1; (a)[15] = c2; \\n"
00148     "}\\n"
00149     "\\n"
00150     "#define fftKernel32(a,dir) \\n"
00151     "{ \\n"
00152     "    fftKernel2S((a)[0], (a)[16], dir); \\n"
00153     "    fftKernel2S((a)[1], (a)[17], dir); \\n"
00154     "    fftKernel2S((a)[2], (a)[18], dir); \\n"
00155     "    fftKernel2S((a)[3], (a)[19], dir); \\n"
00156     "    fftKernel2S((a)[4], (a)[20], dir); \\n"
00157     "    fftKernel2S((a)[5], (a)[21], dir); \\n"
00158     "    fftKernel2S((a)[6], (a)[22], dir); \\n"
00159     "    fftKernel2S((a)[7], (a)[23], dir); \\n"
00160     "    fftKernel2S((a)[8], (a)[24], dir); \\n"
00161     "    fftKernel2S((a)[9], (a)[25], dir); \\n"
00162     "    fftKernel2S((a)[10], (a)[26], dir); \\n"
00163     "    fftKernel2S((a)[11], (a)[27], dir); \\n"
00164     "    fftKernel2S((a)[12], (a)[28], dir); \\n"
00165     "    fftKernel2S((a)[13], (a)[29], dir); \\n"
00166     "    fftKernel2S((a)[14], (a)[30], dir); \\n"
00167     "    fftKernel2S((a)[15], (a)[31], dir); \\n"
00168     "    (a)[17] = complexMul((a)[17], (float2)(0x1.f6297cp-1f, dir*0x1.8f8b84p-3f)); \\n"
00169     "    (a)[18] = complexMul((a)[18], (float2)(0x1.d906bcp-1f, dir*0x1.87de2ap-2f)); \\n"
00170     "    (a)[19] = complexMul((a)[19], (float2)(0x1.a9b662p-1f, dir*0x1.1c73b4p-1f)); \\n"
00171     "    (a)[20] = complexMul((a)[20], (float2)(0x1.6a09e6p-1f, dir*0x1.6a09e6p-1f)); \\n"
00172     "    (a)[21] = complexMul((a)[21], (float2)(0x1.1c73b4p-1f, dir*0x1.a9b662p-1f)); \\n"
00173     "    (a)[22] = complexMul((a)[22], (float2)(0x1.87de2ap-2f, dir*0x1.d906bcp-1f)); \\n"
00174     "    (a)[23] = complexMul((a)[23], (float2)(0x1.8f8b84p-3f, dir*0x1.f6297cp-1f)); \\n"
00175     "    (a)[24] = complexMul((a)[24], (float2)(0x0p+0f, dir*0x1p+0f)); \\n"
00176     "    (a)[25] = complexMul((a)[25], (float2)(-0x1.8f8b84p-3f, dir*0x1.f6297cp-1f)); \\n"
00177     "    (a)[26] = complexMul((a)[26], (float2)(-0x1.87de2ap-2f, dir*0x1.d906bcp-1f)); \\n"
00178     "    (a)[27] = complexMul((a)[27], (float2)(-0x1.1c73b4p-1f, dir*0x1.a9b662p-1f)); \\n"
00179     "    (a)[28] = complexMul((a)[28], (float2)(-0x1.6a09e6p-1f, dir*0x1.6a09e6p-1f)); \\n"
00180     "    (a)[29] = complexMul((a)[29], (float2)(-0x1.a9b662p-1f, dir*0x1.1c73b4p-1f)); \\n"
00181     "    (a)[30] = complexMul((a)[30], (float2)(-0x1.d906bcp-1f, dir*0x1.87de2ap-2f)); \\n"
00182     "    (a)[31] = complexMul((a)[31], (float2)(-0x1.f6297cp-1f, dir*0x1.8f8b84p-3f)); \\n"
00183     "    fftKernel16((a), dir); \\n"
00184     "    fftKernel16((a) + 16, dir); \\n"
00185     "    bitreverse32((a)); \\n"
00186     "}\\n\\n");
00187
00188 static string twistKernelInterleaved = string(
00189     "__kernel void \\n"
00190     "clFFT_IDTwistInterleaved(__global float2 *in, unsigned int startRow, unsigned int numCols,
unsigned int N, unsigned int numRowsToProcess, int dir) \\n"
00191     "{ \\n"
00192     "    float2 a, w; \\n"
00193     "    float ang; \\n"
00194     "    unsigned int j; \\n"
00195     "    unsigned int i = get_global_id(0); \\n"
00196     "    unsigned int startIndex = i; \\n"
00197     "    \\n"
00198     "    if(i < numCols) \\n"
00199     "    { \\n"
00200     "        for(j = 0; j < numRowsToProcess; j++) \\n"
00201     "        { \\n"
00202     "            a = in[startIndex]; \\n"
00203     "            ang = 2.0f * M_PI * dir * i * (startRow + j) / N; \\n"
00204     "            w = (float2)(native_cos(ang), native_sin(ang)); \\n"
00205     "            a = complexMul(a, w); \\n"
00206     "            in[startIndex] = a; \\n"
00207     "            startIndex += numCols; \\n"
00208     "        } \\n"
00209     "    } \\n"
00210     "} \\n\\n");

```

```

00211
00212 static string twistKernelPlannar = string(
00213     "__kernel void \\\n"
00214     "clFFT_1DTwistSplit(__global float *in_real, __global float *in_imag , unsigned int startRow,
    unsigned int numCols, unsigned int N, unsigned int numRowsToProcess, int dir) \\\n"
00215     "{ \\\n"
00216     "    float2 a, w; \\\n"
00217     "    float ang; \\\n"
00218     "    unsigned int j; \\\n"
00219     "    unsigned int i = get_global_id(0); \\\n"
00220     "    unsigned int startIndex = i; \\\n"
00221     "    \\\n"
00222     "    if(i < numCols) \\\n"
00223     "    { \\\n"
00224     "        for(j = 0; j < numRowsToProcess; j++) \\\n"
00225     "        { \\\n"
00226     "            a = (float2)(in_real[startIndex], in_imag[startIndex]); \\\n"
00227     "            ang = 2.0f * M_PI * dir * i * (startRow + j) / N; \\\n"
00228     "            w = (float2)(native_cos(ang), native_sin(ang)); \\\n"
00229     "            a = complexMul(a, w); \\\n"
00230     "            in_real[startIndex] = a.x; \\\n"
00231     "            in_imag[startIndex] = a.y; \\\n"
00232     "            startIndex += numCols; \\\n"
00233     "        } \\\n"
00234     "    } \\\n"
00235     "} \\\n");
00236
00237
00238 #endif

```

## 11.192 fft\_internal.h File Reference

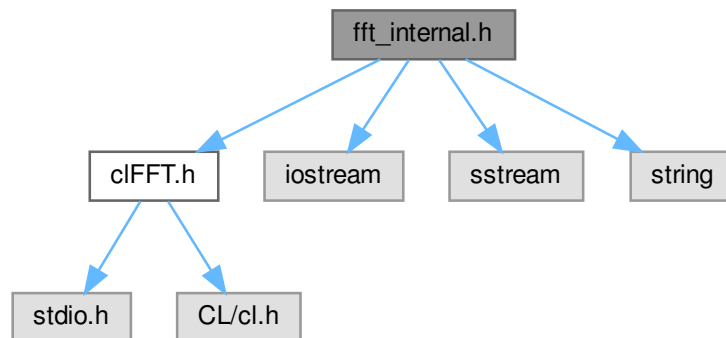
Internals of FFT for OpenCL.

```

#include "clFFT.h"
#include <iostream>
#include <sstream>
#include <string>

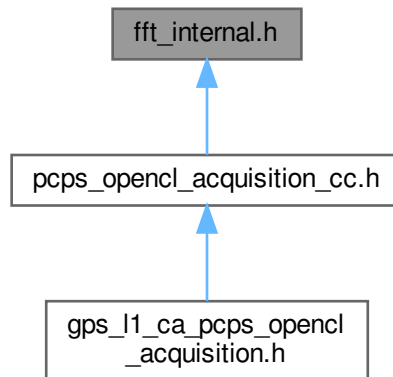
```

Include dependency graph for fft\_internal.h:





This graph shows which files directly or indirectly include this file:



### Classes

- struct `kernel_info_t`
- struct `cl_fft_plan`

### Typedefs

- typedef enum `kernel_dir_t` `cl_fft_kernel_dir`
- typedef struct `kernel_info_t` `cl_fft_kernel_info`

### Enumerations

- enum `kernel_dir_t` { `cl_fft_kernel_x` , `cl_fft_kernel_y` , `cl_fft_kernel_z` }

### Functions

- void `FFT1D` (`cl_fft_plan` \*plan, `cl_fft_kernel_dir` dir)

## 11.192.1 Detailed Description

Internals of FFT for OpenCL.

Version: <1.0>

Copyright ( C ) 2008 Apple Inc. All Rights Reserved. SPDX-License-Identifier: LicenseRef-Apple-Permissive

Definition in file `fft_internal.h`.

## 11.192.2 Enumeration Type Documentation

### 11.192.2.1 `kernel_dir_t`

enum `kernel_dir_t`

Definition at line 25 of file `fft_internal.h`.

## 11.193 `fft_internal.h`

[Go to the documentation of this file.](#)

00001 /\*!

```

00002  * \file fft_internal.h
00003  * \brief Internals of FFT for OpenCL
00004  *
00005  *
00006  * Version:    <1.0>
00007  *
00008  * Copyright ( C ) 2008 Apple Inc. All Rights Reserved.
00009  * SPDX-License-Identifier: LicenseRef-Apple-Permissive
00010  *
00011  *
00012  */
00013
00014
00015 #ifndef __CLFFT_INTERNAL_H
00016 #define __CLFFT_INTERNAL_H
00017
00018 #include "clFFT.h"
00019 #include <iostream>
00020 #include <sstream>
00021 #include <string>
00022
00023 using namespace std;
00024
00025 typedef enum kernel_dir_t
00026 {
00027     cl_fft_kernel_x,
00028     cl_fft_kernel_y,
00029     cl_fft_kernel_z
00030 } cl_fft_kernel_dir;
00031
00032 typedef struct kernel_info_t
00033 {
00034     cl_kernel kernel;
00035     char *kernel_name;
00036     unsigned lmem_size;
00037     unsigned num_workgroups;
00038     unsigned num_xforms_per_workgroup;
00039     unsigned num_workitems_per_workgroup;
00040     cl_fft_kernel_dir dir;
00041     int in_place_possible;
00042     kernel_info_t *next;
00043 } cl_fft_kernel_info;
00044
00045 typedef struct
00046 {
00047     // context in which fft resources are created and kernels are executed
00048     cl_context context;
00049
00050     // size of signal
00051     clFFT_Dim3 n;
00052
00053     // dimension of transform ... must be either 1D, 2D or 3D
00054     clFFT_Dimension dim;
00055
00056     // data format ... must be either interleaved or plannar
00057     clFFT_DataFormat format;
00058
00059     // string containing kernel source. Generated at runtime based on
00060     // n, dim, format and other parameters
00061     string *kernel_string;
00062
00063     // CL program containing source and kernel this particular
00064     // n, dim, data format
00065     cl_program program;
00066
00067     // linked list of kernels which needs to be executed for this fft
00068     cl_fft_kernel_info *kernel_info;
00069
00070     // number of kernels
00071     int num_kernels;
00072
00073     // twist kernel for virtualizing fft of very large sizes that do not
00074     // fit in GPU global memory
00075     cl_kernel twist_kernel;
00076
00077     // flag indicating if temporary intermediate buffer is needed or not.
00078     // this depends on fft kernels being executed and if transform is
00079     // in-place or out-of-place. e.g. Local memory fft (say 1D 1024 ...
00080     // one that does not require global transpose do not need temporary buffer)
00081     // 2D 1024x1024 out-of-place fft however do require intermediate buffer.
00082     // If temp buffer is needed, its allocation is lazy i.e. its not allocated
00083     // until its needed
00084     cl_int temp_buffer_needed;
00085
00086     // Batch size is runtime parameter and size of temporary buffer (if needed)
00087     // depends on batch size. Allocation of temporary buffer is lazy i.e. its
00088     // only created when needed. Once its created at first call of clFFT_Executexxx

```

```

00089 // it is not allocated next time if next time clFFT_Executexxx is called with
00090 // batch size different than the first call. last_batch_size caches the last
00091 // batch size with which this plan is used so that we dont keep allocating/deallocating
00092 // temp buffer if same batch size is used again and again.
00093 unsigned last_batch_size;
00094
00095 // temporary buffer for interleaved plan
00096 cl_mem tempmemobj;
00097
00098 // temporary buffer for planner plan. Only one of tempmemobj or
00099 // (tempmemobj_real, tempmemobj_imag) pair is valid (allocated) depending
00100 // data format of plan (plannar or interleaved)
00101 cl_mem tempmemobj_real, tempmemobj_imag;
00102
00103 // Maximum size of signal for which local memory transposed based
00104 // fft is sufficient i.e. no global mem transpose (communication)
00105 // is needed
00106 unsigned max_localmem_fft_size;
00107
00108 // Maximum work items per work group allowed. This, along with max_radix below controls
00109 // maximum local memory being used by fft kernels of this plan. Set to 256 by default
00110 unsigned max_work_item_per_workgroup;
00111
00112 // Maximum base radix for local memory fft ... this controls the maximum register
00113 // space used by work items. Currently defaults to 16
00114 unsigned max_radix;
00115
00116 // Device depended parameter that tells how many work-items need to be read consecutive
00117 // values to make sure global memory access by work-items of a work-group result in
00118 // coalesced memory access to utilize full bandwidth e.g. on NVidia tesla, this is 16
00119 unsigned min_mem_coalesce_width;
00120
00121 // Number of local memory banks. This is used to geneate kernel with local memory
00122 // transposes with appropriate padding to avoid bank conflicts to local memory
00123 // e.g. on NVidia it is 16.
00124 unsigned num_local_mem_banks;
00125 } cl_fft_plan;
00126
00127 void FFT1D(cl_fft_plan *plan, cl_fft_kernel_dir dir);
00128
00129 #endif

```

## 11.194 pass\_through.h File Reference

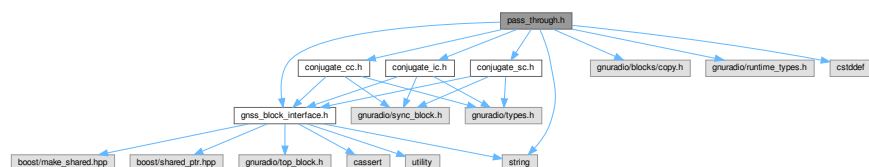
Interface of a block that just puts its input in its output.

```

#include "conjugate_cc.h"
#include "conjugate_ic.h"
#include "conjugate_sc.h"
#include "gnss_block_interface.h"
#include <gnuradio/blocks/copy.h>
#include <gnuradio/runtime_types.h>
#include <cstdint>
#include <string>

```

Include dependency graph for pass\_through.h:



## Classes

- class [Pass\\_Through](#)

*This class implements a block that connects input and output (does nothing)*

### 11.194.1 Detailed Description

Interface of a block that just puts its input in its output.

Author

Carlos Aviles, 2010. carlos.avilesr(at)gmail.com

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [pass\\_through.h](#).

---

## 11.195 pass\_through.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file pass_through.h
00003  * \brief Interface of a block that just puts its input in its
00004  *       output.
00005  * \author Carlos Aviles, 2010. carlos.avilesr(at)gmail.com
00006  *
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019 #ifndef GNSS_SDR_PASS_THROUGH_H
00020 #define GNSS_SDR_PASS_THROUGH_H
00021
00022 #include "conjugate_cc.h"
00023 #include "conjugate_ic.h"
00024 #include "conjugate_sc.h"
00025 #include "gnss_block_interface.h"
00026 #include <gnuradio/blocks/copy.h>
00027 #include <gnuradio/runtime_types.h>
00028 #include <cstdint>
00029 #include <string>
00030
00031 /** \addtogroup Algorithms_Library
00032  *  \{ */
00033 /** \addtogroup Algorithm_libs algorithms_libs
00034  *  \{ */
00035
00036
00037 class ConfigurationInterface;
00038
00039 /*!
00040 * \brief This class implements a block that connects input and output (does nothing)
00041 */
00042 class Pass_Through : public GNSSBlockInterface
00043 {
00044 public:
00045     Pass_Through(const ConfigurationInterface* configuration,
00046                 const std::string& role,
00047                 unsigned int in_stream,
00048                 unsigned int out_stream);
00049
00050     ~Pass_Through() = default;
00051
00052     inline std::string role() override
00053     {
00054         return role_;
00055     }
00056
00057     /*! Returns "Pass_Through"
00058     inline std::string implementation() override
00059     {
00060         return "Pass_Through";
00061     }
00062
00063     inline std::string item_type() const
00064     {
00065         return item_type_;
00066     }
00067 
```

## 11.196 rtklib.h File Reference

```
#include "MATH_CONSTANTS.h"
#include "gnss_frequencies.h"
#include "gnss_obs_codes.h"
#include <cctype>
#include <cmath>
#include <cstdlib>
#include <stdint>
#include <stdlib.h>
#include <netinet/in.h>
#include <pthread.h>
#include <string>
```

- struct `gtime_t`
- struct `obsd_t`
- struct `obs_t`
- struct `erpd_t`

- struct [erp\\_t](#)
- struct [pcv\\_t](#)
- struct [pcvs\\_t](#)
- struct [alm\\_t](#)
- struct [eph\\_t](#)
- struct [geph\\_t](#)
- struct [peph\\_t](#)
- struct [pclk\\_t](#)
- struct [seph\\_t](#)
- struct [tled\\_t](#)
- struct [tle\\_t](#)
- struct [tec\\_t](#)
- struct [fcbd\\_t](#)
- struct [sbsmsg\\_t](#)
- struct [sbs\\_t](#)
- struct [sbsfcrr\\_t](#)
- struct [sbslcorr\\_t](#)
- struct [sbssatp\\_t](#)
- struct [sbssat\\_t](#)
- struct [sbsigp\\_t](#)
- struct [sbsigpband\\_t](#)
- struct [sbsion\\_t](#)
- struct [dgps\\_t](#)
- struct [ssr\\_t](#)
- struct [lexmsg\\_t](#)
- struct [lex\\_t](#)
- struct [lexeph\\_t](#)
- struct [lexion\\_t](#)
- struct [stec\\_t](#)
- struct [trop\\_t](#)
- struct [pppcorr\\_t](#)
- struct [nav\\_t](#)
- struct [sta\\_t](#)
- struct [sol\\_t](#)
- struct [solbuf\\_t](#)
- struct [solstat\\_t](#)
- struct [solstatbuf\\_t](#)
- struct [rtcm\\_t](#)
- struct [url\\_t](#)
- struct [opt\\_t](#)
- struct [exterr\\_t](#)
- struct [snrmask\\_t](#)
- struct [prcopt\\_t](#)
- struct [solopt\\_t](#)
- struct [ssat\\_t](#)
- struct [ambc\\_t](#)
- struct [rtk\\_t](#)
- struct [half\\_cyc\\_tag](#)
- struct [stream\\_t](#)
- struct [serial\\_t](#)
- struct [file\\_t](#)
- struct [tcp\\_t](#)
- struct [tcpsvr\\_t](#)
- struct [tcpcli\\_t](#)
- struct [ntrip\\_t](#)

- struct [ftp\\_t](#)
- struct [raw\\_t](#)
- struct [rtksvr\\_t](#)
- struct [msm\\_h\\_t](#)

## Macros

- #define [dev\\_t](#) int
- #define [socket\\_t](#) int
- #define [closesocket](#) close
- #define [lock\\_t](#) pthread\_mutex\_t
- #define [initlock](#)(f)
- #define [rtk\\_lock](#)(f)
- #define [rtk\\_unlock](#)(f)
- #define [VER\\_RTKLIB](#) "2.4.2"
- #define [NTRIP\\_AGENT](#) "RTKLIB/" VER\_RTKLIB
- #define [NTRIP\\_CLI\\_PORT](#) 2101 /\* default ntrip-client connection port \*/
- #define [NTRIP\\_SVR\\_PORT](#) 80 /\* default ntrip-server connection port \*/
- #define [NTRIP\\_MAXRSP](#) 32768 /\* max size of ntrip response \*/
- #define [NTRIP\\_MAXSTR](#) 256 /\* max length of mountpoint string \*/
- #define [NTRIP\\_RSP\\_OK\\_CLI](#) "ICY 200 OK\r\n" /\* ntrip response: client \*/
- #define [NTRIP\\_RSP\\_OK\\_SVR](#) "OK\r\n" /\* ntrip response: server \*/
- #define [NTRIP\\_RSP\\_SRCTBL](#) "SOURCETABLE 200 OK\r\n" /\* ntrip response: source table \*/
- #define [NTRIP\\_RSP\\_TBLEND](#) "ENDSOURCETABLE"
- #define [NTRIP\\_RSP\\_HTTP](#) "HTTP/" /\* ntrip response: http \*/
- #define [NTRIP\\_RSP\\_ERROR](#) "ERROR" /\* ntrip response: error \*/
- #define [FTP\\_CMD](#) "wget" /\* ftp/http command \*/
- #define [ENAGLO](#)
- #define [ENABDS](#)
- #define [STR\\_MODE\\_R](#) 0x1 /\* stream mode: read \*/
- #define [STR\\_MODE\\_W](#) 0x2 /\* stream mode: write \*/
- #define [STR\\_MODE\\_RW](#) 0x3 /\* stream mode: read/write \*/
- #define [STR\\_NONE](#) 0 /\* stream type: none \*/
- #define [STR\\_SERIAL](#) 1 /\* stream type: serial \*/
- #define [STR\\_FILE](#) 2 /\* stream type: file \*/
- #define [STR\\_TCPSVR](#) 3 /\* stream type: TCP server \*/
- #define [STR\\_TCPCLI](#) 4 /\* stream type: TCP client \*/
- #define [STR\\_UDP](#) 5 /\* stream type: UDP stream \*/
- #define [STR\\_NTRIPSVR](#) 6 /\* stream type: NTRIP server \*/
- #define [STR\\_NTRIPCLI](#) 7 /\* stream type: NTRIP client \*/
- #define [STR\\_FTP](#) 8 /\* stream type: ftp \*/
- #define [STR\\_HTTP](#) 9 /\* stream type: http \*/
- #define [NP\\_PPP](#)(opt)
- #define [IC\\_PPP](#)(s, opt)
- #define [IT\\_PPP](#)(opt)
- #define [NR\\_PPP](#)(opt)
- #define [IB\\_PPP](#)(s, opt)
- #define [NX\\_PPP](#)(opt)
- #define [NF\\_RTK](#)(opt)
- #define [NP\\_RTK](#)(opt)
- #define [NI\\_RTK](#)(opt)
- #define [NT\\_RTK](#)(opt)
- #define [NL\\_RTK](#)(opt)
- #define [NB\\_RTK](#)(opt)
- #define [NR\\_RTK](#)(opt)
- #define [NX\\_RTK](#)(opt)

## Typedefs

- using `fatalfunc_t` = void(const char \*)  
*fatal callback function type*
- typedef struct `half_cyc_tag` `half_cyc_t`

## Variables

- const int `TINTACT` = 200  
*period for stream active (ms)*
- const int `SERIBUFFSIZE` = 4096  
*serial buffer size (bytes)*
- const int `TIMETAGH_LEN` = 64  
*time tag file header length*
- const int `MAXCLI` = 32  
*max client connection for tcp svr*
- const int `MAXSTATMSG` = 32  
*max length of status message*
- const int `FTP_TIMEOUT` = 30  
*ftp/http timeout (s)*
- const int `MAXRAWLEN` = 4096  
*max length of receiver raw message*
- const int `MAXSOLBUF` = 256  
*max number of solution buffer*
- const int `MAXSBMSG` = 32  
*max number of SBAS msg in RTK server*
- const int `MAXOBSBUF` = 128  
*max number of observation data buffer*
- const int `FILEPATHSEP` = '/'
- const double `RE_WGS84` = 6378137.0  
*earth semimajor axis (WGS84) (m)*
- const double `FE_WGS84` = (1.0 / 298.257223563)  
*earth flattening (WGS84)*
- const double `HION` = 350000.0  
*ionosphere height (m)*
- const double `PRN_HWBIAS` = 1e-6  
*process noise of h/w bias (m/MHz/sqrt(s))*
- const double `INT_SWAP_STAT` = 86400.0  
*swap interval of solution status file (s)*
- const double `INT_SWAP_TRAC` = 86400.0  
*swap interval of trace file (s)*
- const unsigned int `POLYCRC32` = 0xEDB88320u  
*CRC32 polynomial.*
- const unsigned int `POLYCRC24Q` = 0x1864CFBu  
*CRC24Q polynomial.*
- const int `PMODE_SINGLE` = 0  
*positioning mode: single*
- const int `PMODE_DGPS` = 1  
*positioning mode: DGPS/DGNSS*
- const int `PMODE_KINEMA` = 2  
*positioning mode: kinematic*
- const int `PMODE_STATIC` = 3



- positioning mode: static*
- const int [PMODE\\_MOVEB](#) = 4
- positioning mode: moving-base*
- const int [PMODE\\_FIXED](#) = 5
- positioning mode: fixed*
- const int [PMODE\\_PPP\\_KINEMA](#) = 6
- positioning mode: PPP-kinematic*
- const int [PMODE\\_PPP\\_STATIC](#) = 7
- positioning mode: PPP-static*
- const int [PMODE\\_PPP\\_FIXED](#) = 8
- positioning mode: PPP-fixed*
- const int [SOLF\\_LLH](#) = 0
- solution format: lat/lon/height*
- const int [SOLF\\_XYZ](#) = 1
- solution format: x/y/z-ecef*
- const int [SOLF\\_ENU](#) = 2
- solution format: e/n/u-baseline*
- const int [SOLF\\_NMEA](#) = 3
- solution format: NMEA-183*
- const int [SOLF\\_STAT](#) = 4
- solution format: solution status*
- const int [SOLF\\_GSIF](#) = 5
- solution format: GSI F1/F2*
- const int [SOLQ\\_NONE](#) = 0
- solution status: no solution*
- const int [SOLQ\\_FIX](#) = 1
- solution status: fix*
- const int [SOLQ\\_FLOAT](#) = 2
- solution status: float*
- const int [SOLQ\\_SBAS](#) = 3
- solution status: SBAS*
- const int [SOLQ\\_DGPS](#) = 4
- solution status: DGPS/DGNSS*
- const int [SOLQ\\_SINGLE](#) = 5
- solution status: single*
- const int [SOLQ\\_PPP](#) = 6
- solution status: PPP*
- const int [SOLQ\\_DR](#) = 7
- solution status: dead reckoning*
- const int [MAXSOLQ](#) = 7
- max number of solution status*
- const int [TIMES\\_GPST](#) = 0
- time system: gps time*
- const int [TIMES\\_UTC](#) = 1
- time system: utc*
- const int [TIMES\\_JST](#) = 2
- time system: jst*
- const double [ERR\\_SAAS](#) = 0.3
- saastamoinen model error std (m)*
- const double [ERR\\_BRDCI](#) = 0.5
- broadcast iono model error factor*

- const double `ERR_CBIAS` = 0.3  
*code bias error std (m)*
- const double `REL_HUMI` = 0.7  
*relative humidity for saastamoinen model*
- const double `GAP_RESION` = 120  
*default gap to reset ionos parameters (ep)*
- const int `MAXFREQ` = 7  
*max NFREQ*
- const int `MAXLEAPS` = 64  
*max number of leap seconds table*
- const double `DTTOL` = 0.005  
*tolerance of time difference (s)*
- const int `NFREQ` = 3  
*number of carrier frequencies*
- const int `NFREQGLO` = 2  
*number of carrier frequencies of GLONASS*
- const int `NEXOBS` = 0  
*number of extended obs codes*
- const int `MAXANT` = 64  
*max length of station name/antenna type*
- const int `MINPRNGPS` = 1  
*min satellite PRN number of GPS*
- const int `MAXPRNGPS` = 32  
*max satellite PRN number of GPS*
- const int `NSATGPS` = (`MAXPRNGPS` - `MINPRNGPS` + 1)  
*number of GPS satellites*
- const int `NSYSGPS` = 1
- const int `SYS_NONE` = 0x00  
*navigation system: none*
- const int `SYS_GPS` = 0x01  
*navigation system: GPS*
- const int `SYS_SBS` = 0x02  
*navigation system: SBAS*
- const int `SYS_GLO` = 0x04  
*navigation system: GLONASS*
- const int `SYS_GAL` = 0x08  
*navigation system: Galileo*
- const int `SYS_QZS` = 0x10  
*navigation system: QZSS*
- const int `SYS_BDS` = 0x20  
*navigation system: BeiDou*
- const int `SYS_IRN` = 0x40  
*navigation system: IRNS*
- const int `SYS_LEO` = 0x80  
*navigation system: LEO*
- const int `SYS_ALL` = 0xFF  
*navigation system: all*
- const int `MINPRNGLO` = 1  
*min satellite slot number of GLONASS*
- const int `MAXPRNGLO` = 27  
*max satellite slot number of GLONASS*

- const int `NSATGLO` = (`MAXPRNGLO` - `MINPRNGLO` + 1)  
*number of GLONASS satellites*
- const int `NSYSGLO` = 1
- const int `MINPRNGAL` = 1  
*min satellite PRN number of Galileo*
- const int `MAXPRNGAL` = 36  
*max satellite PRN number of Galileo*
- const int `NSATGAL` = (`MAXPRNGAL` - `MINPRNGAL` + 1)  
*number of Galileo satellites*
- const int `NSYSGAL` = 1
- const int `MAXPRNQZS` = 199  
*max satellite PRN number of QZSS*
- const int `MINPRNQZS` = 193  
*min satellite PRN number of QZSS*
- const int `MINPRNQZS_S` = 0
- const int `NSATQZS` = 0
- const int `NSYSQZS` = 0
- const int `MINPRNBDS` = 1  
*min satellite sat number of BeiDou*
- const int `MAXPRNBDS` = 63  
*max satellite sat number of BeiDou*
- const int `NSATBDS` = (`MAXPRNBDS` - `MINPRNBDS` + 1)  
*number of BeiDou satellites*
- const int `NSYSBDS` = 1
- const int `MINPRNIRN` = 1  
*min satellite sat number of IRNSS*
- const int `MAXPRNIRN` = 7  
*max satellite sat number of IRNSS*
- const int `NSATIRN` = 0
- const int `NSYSIRN` = 0
- const int `MINPRNLEO` = 1  
*min satellite sat number of LEO*
- const int `MAXPRNLEO` = 10  
*max satellite sat number of LEO \*/*
- const int `NSATLEO` = 0
- const int `NSYSLEO` = 0
- const int `NSYS` = (`NSYSGPS` + `NSYSGLO` + `NSYSGAL` + `NSYSQZS` + `NSYSBDS` + `NSYSIRN` + `NSYSLEO`)  
*number of systems*
- const int `MINPRNSBS` = 120  
*min satellite PRN number of SBAS*
- const int `MAXPRNSBS` = 142  
*max satellite PRN number of SBAS*
- const int `NSATSBS` = (`MAXPRNSBS` - `MINPRNSBS` + 1)  
*number of SBAS satellites*
- const int `MAXSAT` = (`NSATGPS` + `NSATGLO` + `NSATGAL` + `NSATQZS` + `NSATBDS` + `NSATIRN` + `NSATSBS` + `NSATLEO`)
- const int `MAXSTA` = 255
- const int `MAXOBS` = 64  
*max number of obs in an epoch*
- const int `MAXRCV` = 64  
*max receiver number (1 to MAXRCV)*
- const int `MAXOBSTYPE` = 64

- max number of obs type in RINEX*
- const double `MAXDToe` = 7200.0
- max time difference to GPS Toe (s)*
- const double `MAXDToe_QZS` = 7200.0
- max time difference to QZSS Toe (s)*
- const double `MAXDToe_GAL` = 10800.0
- max time difference to Galileo Toe (s)*
- const double `MAXDToe_BDS` = 21600.0
- max time difference to BeiDou Toe (s)*
- const double `MAXDToe_GLO` = 1800.0
- max time difference to GLONASS Toe (s)*
- const double `MAXDToe_SBS` = 360.0
- max time difference to SBAS Toe (s)*
- const double `MAXDToe_S` = 86400.0
- max time difference to ephemeris toe (s) for other*
- const double `MAXGDOP` = 300.0
- max GDOP*
- const int `MAXSBSURA` = 8
- max URA of SBAS satellite*
- const int `MAXBAND` = 10
- max SBAS band of IGP*
- const int `MAXNIGP` = 201
- max number of IGP in SBAS band*
- const int `MAXNGEO` = 4
- max number of GEO satellites*
- const int `MAXSOLMSG` = 8191
- max length of solution message*
- const int `MAXERRMSG` = 4096
- max length of error/warning message*
- const int `IONOOPT_OFF` = 0
- ionosphere option: correction off*
- const int `IONOOPT_BRDC` = 1
- ionosphere option: broadcast model*
- const int `IONOOPT_SBAS` = 2
- ionosphere option: SBAS model*
- const int `IONOOPT_IFLC` = 3
- ionosphere option: L1/L2 or L1/L5 iono-free LC*
- const int `IONOOPT_EST` = 4
- ionosphere option: estimation*
- const int `IONOOPT_TEC` = 5
- ionosphere option: IONEX TEC model*
- const int `IONOOPT_QZS` = 6
- ionosphere option: QZSS broadcast model*
- const int `IONOOPT_LEX` = 7
- ionosphere option: QZSS LEX ionosphere*
- const int `IONOOPT_STEC` = 8
- ionosphere option: SLANT TEC model*
- const int `TROPOPT_OFF` = 0
- troposphere option: correction off*
- const int `TROPOPT_SAAS` = 1
- troposphere option: Saastamoinen model*

- const int [TROPOPT\\_SBAS](#) = 2  
*troposphere option: SBAS model*
- const int [TROPOPT\\_EST](#) = 3  
*troposphere option: ZTD estimation*
- const int [TROPOPT\\_ESTG](#) = 4  
*troposphere option: ZTD+grad estimation*
- const int [TROPOPT\\_COR](#) = 5  
*troposphere option: ZTD correction*
- const int [TROPOPT\\_CORG](#) = 6  
*troposphere option: ZTD+grad correction*
- const int [EPHOPT\\_BRDC](#) = 0  
*ephemeris option: broadcast ephemeris*
- const int [EPHOPT\\_PREC](#) = 1  
*ephemeris option: precise ephemeris*
- const int [EPHOPT\\_SBAS](#) = 2  
*ephemeris option: broadcast + SBAS*
- const int [EPHOPT\\_SSRAPC](#) = 3  
*ephemeris option: broadcast + SSR\_APC*
- const int [EPHOPT\\_SSRCOM](#) = 4  
*ephemeris option: broadcast + SSR\_COM*
- const int [EPHOPT\\_LEX](#) = 5  
*ephemeris option: QZSS LEX ephemeris*
- const double [EFACT\\_GPS](#) = 1.0  
*error factor: GPS*
- const double [EFACT\\_GLO](#) = 1.5  
*error factor: GLONASS*
- const double [EFACT\\_GAL](#) = 1.0  
*error factor: Galileo*
- const double [EFACT\\_QZS](#) = 1.0  
*error factor: QZSS*
- const double [EFACT\\_BDS](#) = 1.0  
*error factor: BeiDou*
- const double [EFACT\\_IRN](#) = 1.5  
*error factor: IRNSS*
- const double [EFACT\\_SBS](#) = 3.0  
*error factor: SBAS*
- const int [MAXEXFILE](#) = 1024  
*max number of expanded files*
- const double [MAXSBSAGEF](#) = 30.0  
*max age of SBAS fast correction (s)*
- const double [MAXSBSAGEL](#) = 1800.0  
*max age of SBAS long term corr (s)*
- const int [ARMODE\\_OFF](#) = 0  
*AR mode: off.*
- const int [ARMODE\\_CONT](#) = 1  
*AR mode: continuous.*
- const int [ARMODE\\_INST](#) = 2  
*AR mode: instantaneous.*
- const int [ARMODE\\_FIXHOLD](#) = 3  
*AR mode: fix and hold.*
- const int [ARMODE\\_PPPAR](#) = 4

- AR mode: PPP-AR.*
- const int `ARMODE_PPPAR_ILS` = 5
- AR mode: AR mode: PPP-AR ILS.*
- const int `ARMODE_WLNL` = 6
- const int `ARMODE_TCAR` = 7
- const int `POSOPT_RINEX` = 3
- pos option: rinex header pos*
- const int `MAXSTRPATH` = 1024
- max length of stream path*
- const int `MAXSTRMSG` = 1024
- max length of stream message*
- const double `CHISQR` [100]
- const double `LAM_CARR` [`MAXFREQ`]
- const int `STRFMT_RTCM2` = 0
- const int `STRFMT_RTCM3` = 1
- const int `STRFMT_SP3` = 16
- const int `STRFMT_RNXCLK` = 17
- const int `STRFMT_SBAS` = 18
- const int `STRFMT_NMEA` = 19
- const int `MAXSTRRTK` = 8

### 11.196.1 Detailed Description

main header file for the rtklib library

#### Authors

- 2007-2013, T. Takasu
- 2017, Javier Arribas
- 2017-2023, Carles Fernandez

This is a derived work from RTKLIB <http://www.rtklib.com/> The original source code at <https://github.com/tomojitakasu/RTKLIB> is released under the BSD 2-clause license with an additional exclusive clause that does not apply here. This additional clause is reproduced below:

" The software package includes some companion executive binaries or shared libraries necessary to execute APs on Windows. These licenses succeed to the original ones of these software. "

Neither the executive binaries nor the shared libraries are required by, used or included in GNSS-SDR.

Copyright (C) 2007-2013, T. Takasu Copyright (C) 2017, Javier Arribas Copyright (C) 2017-2023, Carles Fernandez All rights reserved.

SPDX-License-Identifier: BSD-2-Clause  
Definition in file [rtklib.h](#).

## 11.197 rtklib.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file rtklib.h
00003  * \brief main header file for the rtklib library
00004  * \authors <ul>
00005  *         <li> 2007-2013, T. Takasu
00006  *         <li> 2017, Javier Arribas
00007  *         <li> 2017-2023, Carles Fernandez
00008  *         </ul>
00009  *
00010  * This is a derived work from RTKLIB http://www.rtklib.com/
00011  * The original source code at https://github.com/tomojitakasu/RTKLIB is
00012  * released under the BSD 2-clause license with an additional exclusive clause
00013  * that does not apply here. This additional clause is reproduced below:
00014  *
00015  * " The software package includes some companion executive binaries or shared
00016  * libraries necessary to execute APs on Windows. These licenses succeed to the
00017  * original ones of these software. "
```

```

00018  *
00019  * Neither the executive binaries nor the shared libraries are required by, used
00020  * or included in GNSS-SDR.
00021  *
00022  * -----
00023  * Copyright (C) 2007-2013, T. Takasu
00024  * Copyright (C) 2017, Javier Arribas
00025  * Copyright (C) 2017-2023, Carles Fernandez
00026  * All rights reserved.
00027  *
00028  * SPDX-License-Identifier: BSD-2-Clause
00029  *
00030  * -----
00031  */
00032
00033 #ifndef GNSS_SDR_RTKLIB_H
00034 #define GNSS_SDR_RTKLIB_H
00035
00036 #include "MATH_CONSTANTS.h"
00037 #include "gnss_frequencies.h"
00038 #include "gnss_obs_codes.h"
00039 #include <cctype>
00040 #include <cmath>
00041 #include <cstdarg>
00042 #include <stdint>
00043 #include <stdlib>
00044 #include <netinet/in.h>
00045 #include <pthread.h>
00046 #include <string>
00047
00048 /** \addtogroup PVT
00049  * \{ */
00050 /** \addtogroup RTKLIB_Library algorithms_libs_rtklib
00051  * Our version of the RTKLIB core library (see http://www.rtklib.com/)
00052  * \{ */
00053
00054
00055 /* macros -----*/
00056
00057 #define dev_t int
00058 #define socket_t int
00059 #define closesocket close
00060 #define lock_t pthread_mutex_t
00061 #define initlock(f) pthread_mutex_init(f, NULL)
00062 #define rtk_lock(f) pthread_mutex_lock(f)
00063 #define rtk_unlock(f) pthread_mutex_unlock(f)
00064
00065 #define VER_RTKLIB "2.4.2"
00066 #define NTRIP_AGENT "RTKLIB/" VER_RTKLIB
00067 #define NTRIP_CLI_PORT 2101 /* default ntrip-client connection port */
00068 #define NTRIP_SVR_PORT 80 /* default ntrip-server connection port */
00069 #define NTRIP_MAXRSP 32768 /* max size of ntrip response */
00070 #define NTRIP_MAXSTR 256 /* max length of mountpoint string */
00071 #define NTRIP_RSP_OK_CLI "ICY 200 OK\r\n" /* ntrip response: client */
00072 #define NTRIP_RSP_OK_SVR "OK\r\n" /* ntrip response: server */
00073 #define NTRIP_RSP_SRCTBL "SOURCETABLE 200 OK\r\n" /* ntrip response: source table */
00074 #define NTRIP_RSP_TBLEND "ENDSOURCETABLE"
00075 #define NTRIP_RSP_HTTP "HTTP/" /* ntrip response: http */
00076 #define NTRIP_RSP_ERROR "ERROR" /* ntrip response: error */
00077
00078 #define FTP_CMD "wget" /* ftp/http command */
00079
00080 const int TINTACT = 200; /*!< period for stream active (ms)
00081 const int SERIBUFFSIZE = 4096; /*!< serial buffer size (bytes)
00082 const int TIMETAGH_LEN = 64; /*!< time tag file header length
00083 const int MAXCLI = 32; /*!< max client connection for tcp svr
00084 const int MAXSTATMSG = 32; /*!< max length of status message
00085
00086 const int FTP_TIMEOUT = 30; /*!< ftp/http timeout (s)
00087 const int MAXRAWLEN = 4096; /*!< max length of receiver raw message
00088 const int MAXSOLBUF = 256; /*!< max number of solution buffer
00089 const int MAXSBMSG = 32; /*!< max number of SBAS msg in RTK server
00090 const int MAXOBSBUF = 128; /*!< max number of observation data buffer
00091
00092 const int FILEPATHSEP = '/';
00093 const double RE_WGS84 = 6378137.0; /*!< earth semimajor axis (WGS84) (m)
00094 const double FE_WGS84 = (1.0 / 298.257223563); /*!< earth flattening (WGS84)
00095
00096 const double HION = 350000.0; /*!< ionosphere height (m)
00097 const double PRN_HWBIAIS = 1e-6; /*!< process noise of h/w bias (m/MHz/sqrt(s))
00098
00099 const double INT_SWAP_STAT = 86400.0; /*!< swap interval of solution status file (s)
00100 const double INT_SWAP_TRAC = 86400.0; /*!< swap interval of trace file (s)
00101
00102 const unsigned int POLYCRC32 = 0xEDB88320u; /*!< CRC32 polynomial
00103 const unsigned int POLYCRC24Q = 0x1864CFBu; /*!< CRC24Q polynomial
00104

```

```

00105 const int PMODE_SINGLE = 0;      //!< positioning mode: single
00106 const int PMODE_DGPS = 1;         //!< positioning mode: DGPS/DGNSS
00107 const int PMODE_KINEMA = 2;       //!< positioning mode: kinematic
00108 const int PMODE_STATIC = 3;       //!< positioning mode: static
00109 const int PMODE_MOVEB = 4;        //!< positioning mode: moving-base
00110 const int PMODE_FIXED = 5;        //!< positioning mode: fixed
00111 const int PMODE_PPP_KINEMA = 6;    //!< positioning mode: PPP-kinematic
00112 const int PMODE_PPP_STATIC = 7;    //!< positioning mode: PPP-static
00113 const int PMODE_PPP_FIXED = 8;     //!< positioning mode: PPP-fixed
00114
00115 const int SOLF_LLH = 0;            //!< solution format: lat/lon/height
00116 const int SOLF_XYZ = 1;           //!< solution format: x/y/z-ecef
00117 const int SOLF_ENU = 2;           //!< solution format: e/n/u-baseline
00118 const int SOLF_NMEA = 3;          //!< solution format: NMEA-183
00119 const int SOLF_STAT = 4;           //!< solution format: solution status
00120 const int SOLF_GSIF = 5;           //!< solution format: GSI F1/F2
00121
00122 const int SOLQ_NONE = 0;           //!< solution status: no solution
00123 const int SOLQ_FIX = 1;            //!< solution status: fix
00124 const int SOLQ_FLOAT = 2;          //!< solution status: float
00125 const int SOLQ_SBAS = 3;           //!< solution status: SBAS
00126 const int SOLQ_DGPS = 4;           //!< solution status: DGPS/DGNSS
00127 const int SOLQ_SINGLE = 5;         //!< solution status: single
00128 const int SOLQ_PPP = 6;            //!< solution status: PPP
00129 const int SOLQ_DR = 7;             //!< solution status: dead reckoning
00130 const int MAXSOLQ = 7;             //!< max number of solution status
00131
00132 const int TIMES_GPST = 0;          //!< time system: gps time
00133 const int TIMES_UTC = 1;           //!< time system: utc
00134 const int TIMES_JST = 2;           //!< time system: jst
00135
00136
00137 const double ERR_SAAS = 0.3;        //!< saastamoinen model error std (m)
00138 const double ERR_BRDCI = 0.5;      //!< broadcast iono model error factor
00139 const double ERR_CBIAS = 0.3;      //!< code bias error std (m)
00140 const double REL_HUMI = 0.7;        //!< relative humidity for saastamoinen model
00141 const double GAP_RESION = 120;      //!< default gap to reset ionos parameters (ep)
00142
00143 const int MAXFREQ = 7;              //!< max NFREQ
00144
00145 const int MAXLEAPS = 64;            //!< max number of leap seconds table
00146 const double DTTOL = 0.005;        //!< tolerance of time difference (s)
00147
00148 const int NFREQ = 3;               //!< number of carrier frequencies
00149 const int NFREQGLO = 2;            //!< number of carrier frequencies of GLONASS
00150 const int NEXOBS = 0;              //!< number of extended obs codes
00151 const int MAXANT = 64;             //!< max length of station name/antenna type
00152
00153 const int MINPRNGPS = 1;           //!< min satellite PRN number of GPS
00154 const int MAXPRNGPS = 32;          //!< max satellite PRN number of GPS
00155 const int NSATGPS = (MAXPRNGPS - MINPRNGPS + 1); //!< number of GPS satellites
00156 const int NSYSGPS = 1;
00157
00158 const int SYS_NONE = 0x00;         //!< navigation system: none
00159 const int SYS_GPS = 0x01;          //!< navigation system: GPS
00160 const int SYS_SBS = 0x02;          //!< navigation system: SBAS
00161 const int SYS_GLO = 0x04;          //!< navigation system: GLONASS
00162 const int SYS_GAL = 0x08;          //!< navigation system: Galileo
00163 const int SYS_QZSS = 0x10;         //!< navigation system: QZSS
00164 const int SYS_BDS = 0x20;          //!< navigation system: BeiDou
00165 const int SYS_IRN = 0x40;          //!< navigation system: IRNS
00166 const int SYS_LEO = 0x80;          //!< navigation system: LEO
00167 const int SYS_ALL = 0xFF;          //!< navigation system: all
00168
00169
00170 #define ENAGLO
00171 #ifdef ENAGLO
00172 const int MINPRNGLO = 1;           //!< min satellite slot number of GLONASS
00173 const int MAXPRNGLO = 27;          //!< max satellite slot number of GLONASS
00174 const int NSATGLO = (MAXPRNGLO - MINPRNGLO + 1); //!< number of GLONASS satellites
00175 const int NSYSGLO = 1;
00176 #else
00177 const int MINPRNGLO = 0;
00178 const int MAXPRNGLO = 0;
00179 const int NSATGLO = 0;
00180 const int NSYSGLO = 0;
00181 #endif
00182
00183
00184 const int MINPRNGAL = 1;           //!< min satellite PRN number of Galileo
00185 const int MAXPRNGAL = 36;          //!< max satellite PRN number of Galileo
00186 const int NSATGAL = (MAXPRNGAL - MINPRNGAL + 1); //!< number of Galileo satellites
00187 const int NSYSGAL = 1;
00188
00189 const int MAXPRNQZSS = 199;        //!< max satellite PRN number of QZSS
00190 const int MINPRNQZSS = 193;        //!< min satellite PRN number of QZSS
00191 #ifdef ENAQZSS

```



```

00192 const int MINPRNQZS_S = 183;          //!< min satellite PRN number of QZSS SAIF
00193 const int MAXPRNQZS_S = 189;          //!< max satellite PRN number of QZSS SAIF
00194 const int NSATQZS = (MAXPRNQZS - MINPRNQZS + 1); //!< number of QZSS satellites
00195 const int NSYSQZS = 1;
00196 #else
00197 const int MINPRNQZS_S = 0;
00198 const int NSATQZS = 0;
00199 const int NSYSQZS = 0;
00200 #endif
00201
00202 #define ENABDS
00203 #ifndef ENABDS
00204 const int MINPRNBDS = 1;          //!< min satellite sat number of BeiDou
00205 const int MAXPRNBDS = 63;        //!< max satellite sat number of BeiDou
00206 const int NSATBDS = (MAXPRNBDS - MINPRNBDS + 1); //!< number of BeiDou satellites
00207 const int NSYSBDS = 1;
00208 #else
00209 const int MINPRNBDS = 0;
00210 const int MAXPRNBDS = 0;
00211 const int NSATBDS = 0;
00212 const int NSYSBDS = 0;
00213 #endif
00214
00215 const int MINPRNIRN = 1; //!< min satellite sat number of IRNSS
00216 const int MAXPRNIRN = 7; //!< max satellite sat number of IRNSS
00217 #ifndef ENAIRN
00218 const int NSATIRN = (MAXPRNIRN - MINPRNIRN + 1); //!< number of IRNSS satellites
00219 const int NSYSIRN = 1;
00220 #else
00221 const int NSATIRN = 0;
00222 const int NSYSIRN = 0;
00223 #endif
00224
00225 const int MINPRNLEO = 1; //!< min satellite sat number of LEO
00226 const int MAXPRNLEO = 10; //!< max satellite sat number of LEO */
00227 #ifndef ENALEO
00228 const int NSATLEO = (MAXPRNLEO - MINPRNLEO + 1); //!< number of LEO satellites
00229 const int NSYSLEO = 1;
00230 #else
00231 const int NSATLEO = 0;
00232 const int NSYSLEO = 0;
00233 #endif
00234
00235 const int NSYS = (NSYSGPS + NSYSGLO + NSYS GAL + NSYSQZS + NSYSBDS + NSYSIRN + NSYSLEO); //!< number
of systems
00236
00237 const int MINPRNSBS = 120;          //!< min satellite PRN number of SBAS
00238 const int MAXPRNSBS = 142;          //!< max satellite PRN number of SBAS
00239 const int NSATSBS = (MAXPRNSBS - MINPRNSBS + 1); //!< number of SBAS satellites
00240
00241 const int MAXSAT = (NSATGPS + NSATGLO + NSAT GAL + NSATQZS + NSATBDS + NSATIRN + NSATSBS + NSATLEO);
00242
00243 const int MAXSTA = 255;
00244
00245 #ifndef MAXOBS
00246 const int MAXOBS = 64; //!< max number of obs in an epoch
00247 #endif
00248
00249 const int MAXRCV = 64;          //!< max receiver number (1 to MAXRCV)
00250 const int MAXOBS_TYPE = 64;    //!< max number of obs type in RINEX
00251 const double MAXDTOE = 7200.0; //!< max time difference to GPS Toe (s)
00252 const double MAXDTOE_QZS = 7200.0; //!< max time difference to QZSS Toe (s)
00253 const double MAXDTOE_GAL = 10800.0; //!< max time difference to Galileo Toe (s)
00254 const double MAXDTOE_BDS = 21600.0; //!< max time difference to BeiDou Toe (s)
00255 const double MAXDTOE_GLO = 1800.0; //!< max time difference to GLONASS Toe (s)
00256 const double MAXDTOE_SBS = 360.0; //!< max time difference to SBAS Toe (s)
00257 const double MAXDTOE_S = 86400.0; //!< max time difference to ephemeris toe (s) for other
00258 const double MAXGDOP = 300.0;   //!< max GDOP
00259
00260 const int MAXSBSURA = 8; //!< max URA of SBAS satellite
00261 const int MAXBAND = 10;   //!< max SBAS band of IGP
00262 const int MAXNIGP = 201;  //!< max number of IGP in SBAS band
00263 const int MAXNGEO = 4;    //!< max number of GEO satellites
00264
00265 const int MAXSOLMSG = 8191; //!< max length of solution message
00266 const int MAXERRMSG = 4096; //!< max length of error/warning message
00267
00268 const int IONOOPT_OFF = 0;   //!< ionosphere option: correction off
00269 const int IONOOPT_BRDC = 1;  //!< ionosphere option: broadcast model
00270 const int IONOOPT_SBAS = 2;  //!< ionosphere option: SBAS model
00271 const int IONOOPT_IFLC = 3;  //!< ionosphere option: L1/L2 or L1/L5 iono-free LC
00272 const int IONOOPT_EST = 4;   //!< ionosphere option: estimation
00273 const int IONOOPT_TEC = 5;   //!< ionosphere option: IONEX TEC model
00274 const int IONOOPT_QZS = 6;   //!< ionosphere option: QZSS broadcast model
00275 const int IONOOPT_LEX = 7;   //!< ionosphere option: QZSS LEX ionosphere
00276 const int IONOOPT_STEC = 8;  //!< ionosphere option: SLANT TEC model
00277

```

```

00278 const int TROPOPT_OFF = 0;    //!< troposphere option: correction off
00279 const int TROPOPT_SAAS = 1;    //!< troposphere option: Saastamoinen model
00280 const int TROPOPT_SBAS = 2;    //!< troposphere option: SBAS model
00281 const int TROPOPT_EST = 3;    //!< troposphere option: ZTD estimation
00282 const int TROPOPT_ESTG = 4;    //!< troposphere option: ZTD+grad estimation
00283 const int TROPOPT_COR = 5;    //!< troposphere option: ZTD correction
00284 const int TROPOPT_CORG = 6;    //!< troposphere option: ZTD+grad correction
00285
00286
00287 const int EPHOPT_BRDC = 0;    //!< ephemeris option: broadcast ephemeris
00288 const int EPHOPT_PREC = 1;    //!< ephemeris option: precise ephemeris
00289 const int EPHOPT_SBAS = 2;    //!< ephemeris option: broadcast + SBAS
00290 const int EPHOPT_SSRAPC = 3;   //!< ephemeris option: broadcast + SSR_APC
00291 const int EPHOPT_SSRCOM = 4;   //!< ephemeris option: broadcast + SSR_COM
00292 const int EPHOPT_LEX = 5;     //!< ephemeris option: QZSS LEX ephemeris
00293
00294 const double EFACT_GPS = 1.0;  //!< error factor: GPS
00295 const double EFACT_GLO = 1.5;  //!< error factor: GLONASS
00296 const double EFACT_GAL = 1.0;  //!< error factor: Galileo
00297 const double EFACT_QZSS = 1.0;  //!< error factor: QZSS
00298 const double EFACT_BDS = 1.0;  //!< error factor: BeiDou
00299 const double EFACT_IRNSS = 1.5;  //!< error factor: IRNSS
00300 const double EFACT_SBS = 3.0;  //!< error factor: SBAS
00301
00302 const int MAXEXFILE = 1024;    //!< max number of expanded files
00303 const double MAXSBSAGEF = 30.0;  //!< max age of SBAS fast correction (s)
00304 const double MAXSBSAGEL = 1800.0;  //!< max age of SBAS long term corr (s)
00305
00306 const int ARMODE_OFF = 0;      //!< AR mode: off
00307 const int ARMODE_CONT = 1;     //!< AR mode: continuous
00308 const int ARMODE_INST = 2;     //!< AR mode: instantaneous
00309 const int ARMODE_FIXHOLD = 3;  //!< AR mode: fix and hold
00310 const int ARMODE_PPPAR = 4;    //!< AR mode: PPP-AR
00311 const int ARMODE_PPPAR_ILS = 5;  //!< AR mode: AR mode: PPP-AR ILS
00312 const int ARMODE_WLNL = 6;
00313 const int ARMODE_TCAR = 7;
00314
00315
00316 const int POSOPT_RINEX = 3;    //!< pos option: rinex header pos
00317 const int MAXSTPATH = 1024;    //!< max length of stream path
00318 const int MAXSTRMSG = 1024;    //!< max length of stream message
00319
00320 using fatalfunc_t = void(const char *);  //!< fatal callback function type
00321
00322 // clang-format off
00323 #define STR_MODE_R 0x1 /* stream mode: read */
00324 #define STR_MODE_W 0x2 /* stream mode: write */
00325 #define STR_MODE_RW 0x3 /* stream mode: read/write */
00326
00327 #define STR_NONE 0 /* stream type: none */
00328 #define STR_SERIAL 1 /* stream type: serial */
00329 #define STR_FILE 2 /* stream type: file */
00330 #define STR_TCPSVR 3 /* stream type: TCP server */
00331 #define STR_TCPCLI 4 /* stream type: TCP client */
00332 #define STR_UDP 5 /* stream type: UDP stream */
00333 #define STR_NTRIPSVR 6 /* stream type: NTRIP server */
00334 #define STR_NTRIPCLI 7 /* stream type: NTRIP client */
00335 #define STR_FTP 8 /* stream type: ftp */
00336 #define STR_HTTP 9 /* stream type: http */
00337
00338 #define NP_PPP(opt) ((opt)->dynamics ? 9 : 3)
/* number of pos solution */
00339 #define IC_PPP(s, opt) (NP_PPP(opt) + (s))
/* state index of clocks (s=0:gps,1:glo) */
00340 #define IT_PPP(opt) (IC_PPP(0, opt) + NSYS)
/* state index of tropo */
00341 #define NR_PPP(opt) (IT_PPP(opt) + ((opt)->tropopt < TROPOPT_EST ? 0 : ((opt)->tropopt == TROPOPT_EST
? 1 : 3))) /* number of solutions */
00342 #define IB_PPP(s, opt) (NR_PPP(opt) + (s)-1)
/* state index of phase bias */
00343 #define NX_PPP(opt) (IB_PPP(MAXSAT, opt) + 1)
/* number of estimated states */
00344 // clang-format on
00345
00346 #define NF_RTK(opt) ((opt)->ionoopt == IONOOPT_IFLC ? 1 : (opt)->nf)
00347 #define NP_RTK(opt) ((opt)->dynamics == 0 ? 3 : 9)
00348 #define NI_RTK(opt) ((opt)->ionoopt != IONOOPT_EST ? 0 : MAXSAT)
00349 #define NT_RTK(opt) ((opt)->tropopt < TROPOPT_EST ? 0 : ((opt)->tropopt < TROPOPT_ESTG ? 2 : 6))
00350 #define NL_RTK(opt) ((opt)->glomodear != 2 ? 0 : NFREQGLO)
00351 #define NB_RTK(opt) ((opt)->mode <= PMODE_DGPS ? 0 : MAXSAT * NF_RTK(opt))
00352 #define NR_RTK(opt) (NP_RTK(opt) + NI_RTK(opt) + NT_RTK(opt) + NL_RTK(opt))
00353 #define NX_RTK(opt) (NR_RTK(opt) + NB_RTK(opt))
00354
00355 typedef struct
00356 {
00357     time_t time; /* time (s) expressed by standard time_t */
00358     double sec; /* fraction of second under 1 s */

```

```

00359 } gtime_t;
00360
00361
00362 typedef struct
00363 {
00364     gtime_t time; /* observation data record */
00365     unsigned char sat, rcv; /* receiver sampling time (GPST) */
00366     unsigned char SNR[NFREQ + NEXOBS]; /* satellite/receiver number */
00367     unsigned char LLI[NFREQ + NEXOBS]; /* signal strength (0.25 dBHz) */
00368     unsigned char code[NFREQ + NEXOBS]; /* loss of lock indicator */
00369     double L[NFREQ + NEXOBS]; /* code indicator (CODE_???) */
00370     double P[NFREQ + NEXOBS]; /* observation data carrier-phase (cycle) */
00371     float D[NFREQ + NEXOBS]; /* observation data pseudorange (m) */
00372 } obsd_t; /* observation data doppler frequency (Hz) */
00373
00374
00375 typedef struct
00376 {
00377     int n, nmax; /* observation data */
00378     obsd_t *data; /* number of observation data/allocated */
00379 } obs_t; /* observation data records */
00380
00381
00382 typedef struct
00383 {
00384     double mjd; /* earth rotation parameter data type */
00385     double xp, yp; /* mjd (days) */
00386     double xpr, ypr; /* pole offset (rad) */
00387     double utl_utc; /* pole offset rate (rad/day) */
00388     double lod; /* utl-utc (s) */
00389 } erpd_t; /* length of day (s/day) */
00390
00391
00392 typedef struct
00393 {
00394     int n, nmax; /* earth rotation parameter type */
00395     erpd_t *data; /* number and max number of data */
00396 } erp_t; /* earth rotation parameter data */
00397
00398
00399 typedef struct
00400 {
00401     int sat; /* antenna parameter type */
00402     char type[MAXANT]; /* satellite number (0:receiver) */
00403     char code[MAXANT]; /* antenna type */
00404     gtime_t ts, te; /* serial number or satellite code */
00405     double off[NFREQ][3]; /* valid time start and end */
00406     double var[NFREQ][19]; /* phase center offset e/n/u or x/y/z (m) */
00407     /* phase center variation (m) */
00408     /* el=90,85,...,0 or nadir=0,1,2,3,... (deg) */
00409 } pcv_t;
00410
00411 typedef struct
00412 {
00413     int n, nmax; /* antenna parameters type */
00414     pcv_t *pcv; /* number of data/allocated */
00415 } pcvs_t; /* antenna parameters data */
00416
00417
00418 typedef struct
00419 {
00420     int sat; /* almanac type */
00421     int svh; /* satellite number */
00422     int svconf; /* sv health (0:ok) */
00423     int week; /* as and sv config */
00424     gtime_t toa; /* GPS/QZS: gps week, GAL: galileo week */
00425     /* SV orbit parameters */
00426     double A, e, i0, OMG0, omg, M0, OMGd;
00427     double toas; /* Toa (s) in week */
00428     double f0, f1; /* SV clock parameters (af0,af1) */
00429 } alm_t;
00430
00431
00432 typedef struct
00433 {
00434     int sat; /* GPS/QZS/GAL broadcast ephemeris type */
00435     int iode, iodc; /* satellite number */
00436     int sva; /* IODE,IODC */
00437     int svh; /* SV accuracy (URA index) */
00438     int week; /* SV health (0:ok) */
00439     int code; /* GPS/QZS: gps week, GAL: galileo week */
00440     int flag; /* GPS/QZS: code on L2, GAL/BDS: data sources */
00441     gtime_t toe, toc, ttr; /* GPS/QZS: L2 P data flag, BDS: nav type */
00442     /* Toe,Toc,T_trans */
00443     double A, e, i0, OMG0, omg, M0, deln, OMGd, idot;
00444     double crc, crs, cuc, cus, cic, cis;
00445     double toes; /* SV orbit parameters */
00446     /* Toe (s) in week */

```

```

00446     double fit;                                /* fit interval (h) */
00447     double f0, f1, f2;                          /* SV clock parameters (af0,af1,af2) */
00448     double tgd[4];                              /* group delay parameters */
00449                                           /* GPS/QZS:tgd[0]=TGD */
00450                                           /* GAL      :tgd[0]=BGD E5a/E1,tgd[1]=BGD E5b/E1 */
00451                                           /* BDS      :tgd[0]=BGD1,tgd[1]=BGD2 */
00452     double isc[4];                              /* GPS      :isc[0]=ISCL1, isc[1]=ISCL2, isc[2]=ISCL5I,
isc[3]=ISCL5Q */
00453     double Adot, ndot;                          /* Adot,ndot for CNAV */
00454     float has_clock_correction_m;                /* Galileo High Accuracy Service clock correction, in
[m] */
00455     float has_orbit_radial_correction_m;          /* Galileo High Accuracy Service orbit radial
correction, in [m] */
00456     float has_orbit_in_track_correction_m;        /* Galileo High Accuracy Service orbit in-track
correction, in [m] */
00457     float has_orbit_cross_track_correction_m;     /* Galileo High Accuracy Service orbit cross-track
correction, in [m] */
00458     bool apply_has_corrections;
00459 } eph_t;
00460
00461
00462 typedef struct
00463 {
00464     int sat;                                     /* GLONASS broadcast ephemeris type */
00465     int iode;                                    /* satellite number */
00466     int frq;                                     /* IODE (0-6 bit of tb field) */
00467     int svh, sva, age;                          /* satellite frequency number */
00468     gtime_t toe;                                /* satellite health, accuracy, age of operation */
00469     gtime_t tof;                                /* epoch of ephemerides (gpst) */
00470     double pos[3];                              /* message frame time (gpst) */
00471     double vel[3];                              /* satellite position (ecef) (m) */
00472     double acc[3];                              /* satellite velocity (ecef) (m/s) */
00473     double taun, gamn;                          /* satellite acceleration (ecef) (m/s^2) */
00474     double dtaun;                               /* SV clock bias (s)/relative freq bias */
00475 } gepht_t;
00476
00477
00478 typedef struct
00479 {
00480     gtime_t time;                               /* precise ephemeris type */
00481     int index;                                  /* time (GPST) */
00482     double pos[MAXSAT][4];                     /* ephemeris index for multiple files */
00483     float std[MAXSAT][4];                      /* satellite position/clock (ecef) (m/s) */
00484     double vel[MAXSAT][4];                     /* satellite position/clock std (m/s) */
00485     float vst[MAXSAT][4];                      /* satellite velocity/clk-rate (m/s/s/s) */
00486     float cov[MAXSAT][3];                     /* satellite velocity/clk-rate std (m/s/s/s) */
00487     float vco[MAXSAT][3];                     /* satellite position covariance (m^2) */
00488     float vco[MAXSAT][3];                     /* satellite velocity covariance (m^2) */
00489 } pepht_t;
00490
00491 typedef struct
00492 {
00493     gtime_t time;                               /* precise clock type */
00494     int index;                                  /* time (GPST) */
00495     double clk[MAXSAT][1];                     /* clock index for multiple files */
00496     float std[MAXSAT][1];                     /* satellite clock (s) */
00497     float std[MAXSAT][1];                     /* satellite clock std (s) */
00498 } pclk_t;
00499
00500 typedef struct
00501 {
00502     int sat;                                     /* SBAS ephemeris type */
00503     gtime_t t0;                                 /* satellite number */
00504     gtime_t tof;                                /* reference epoch time (GPST) */
00505     int sva;                                    /* time of message frame (GPST) */
00506     int svh;                                    /* SV accuracy (URA index) */
00507     double pos[3];                             /* SV health (0:ok) */
00508     double vel[3];                             /* satellite position (m) (ecef) */
00509     double acc[3];                             /* satellite velocity (m/s) (ecef) */
00510     double af0, afl;                           /* satellite acceleration (m/s^2) (ecef) */
00511     double af0, afl;                           /* satellite clock-offset/drift (s,s/s) */
00512 } seph_t;
00513
00514 typedef struct
00515 {
00516     char name[32];                             /* norad two line element data type */
00517     char alias[32];                            /* common name */
00518     char satno[16];                           /* alias name */
00519     char satclass;                            /* satellite catalog number */
00520     char desig[16];                           /* classification */
00521     gtime_t epoch;                            /* international designator */
00522     double ndot;                              /* element set epoch (UTC) */
00523     double nddot;                             /* 1st derivative of mean motion */
00524     double bstar;                             /* 2st derivative of mean motion */
00525     int etype;                                /* B* drag term */
00526     int eleno;                                /* element set type */
00527     double inc;                              /* element number */
00528     double inc;                              /* orbit inclination (deg) */

```

```

00528     double OMG;      /* right ascension of ascending node (deg) */
00529     double ecc;        /* eccentricity */
00530     double omg;        /* argument of perigee (deg) */
00531     double M;          /* mean anomaly (deg) */
00532     double n;          /* mean motion (rev/day) */
00533     int rev;           /* revolution number at epoch */
00534 } tled_t;
00535
00536
00537 typedef struct
00538 {
00539     /* norad two line element type */
00539     int n, nmax; /* number/max number of two line element data */
00540     tled_t *data; /* norad two line element data */
00541 } tle_t;
00542
00543
00544 typedef struct
00545 {
00546     /* TEC grid type */
00546     gtime_t time; /* epoch time (GPST) */
00547     int ndata[3]; /* TEC grid data size {nlat,nlon,nhgt} */
00548     double rb; /* earth radius (km) */
00549     double lats[3]; /* latitude start/interval (deg) */
00550     double lons[3]; /* longitude start/interval (deg) */
00551     double hgts[3]; /* heights start/interval (km) */
00552     double *data; /* TEC grid data (tecu) */
00553     float *rms; /* RMS values (tecu) */
00554 } tec_t;
00555
00556
00557 typedef struct
00558 {
00559     /* satellite fcb data type */
00559     gtime_t ts, te; /* start/end time (GPST) */
00560     double bias[MAXSAT][3]; /* fcb value (cyc) */
00561     double std[MAXSAT][3]; /* fcb std-dev (cyc) */
00562 } fcbd_t;
00563
00564
00565 typedef struct
00566 {
00567     /* SBAS message type */
00567     int week, tow; /* reception time */
00568     int prn; /* SBAS satellite PRN number */
00569     unsigned char msg[29]; /* SBAS message (226bit) padded by 0 */
00570 } sbmsg_t;
00571
00572
00573 typedef struct
00574 {
00575     /* SBAS messages type */
00575     int n, nmax; /* number of SBAS messages/allocated */
00576     sbmsg_t *msgs; /* SBAS messages */
00577 } sbs_t;
00578
00579
00580 typedef struct
00581 {
00582     /* SBAS fast correction type */
00582     gtime_t t0; /* time of applicability (TOF) */
00583     double prc; /* pseudorange correction (PRC) (m) */
00584     double rrc; /* range-rate correction (RRC) (m/s) */
00585     double dt; /* range-rate correction delta-time (s) */
00586     int iodf; /* IODF (issue of date fast corr) */
00587     short udre; /* UDRE+1 */
00588     short ai; /* degradation factor indicator */
00589 } sbsfcrr_t;
00590
00591
00592 typedef struct
00593 {
00594     /* SBAS long term satellite error correction type */
00594     gtime_t t0; /* correction time */
00595     int iode; /* IODE (issue of date ephemeris) */
00596     double dpos[3]; /* delta position (m) (ecef) */
00597     double dvel[3]; /* delta velocity (m/s) (ecef) */
00598     double da0, da1; /* delta clock-offset/drift (s,s/s) */
00599 } sbslcorr_t;
00600
00601
00602 typedef struct
00603 {
00604     /* SBAS satellite correction type */
00604     int sat; /* satellite number */
00605     sbsfcrr_t fcorr; /* fast correction */
00606     sbslcorr_t lcorr; /* long term correction */
00607 } sbssatp_t;
00608
00609
00610 typedef struct
00611 {
00612     /* SBAS satellite corrections type */
00612     int iodp; /* IODP (issue of date mask) */
00613     int nsat; /* number of satellites */
00614     int tlat; /* system latency (s) */

```

```

00615     sbssatp_t sat[MAXSAT]; /* satellite correction */
00616 } sbssat_t;
00617
00618
00619 typedef struct
00620 {
00621     /* SBAS ionospheric correction type */
00622     gtime_t t0; /* correction time */
00623     short lat, lon; /* latitude/longitude (deg) */
00624     short give; /* GIVI+1 */
00625     float delay; /* vertical delay estimate (m) */
00626 } sbsigp_t;
00627
00628 typedef struct
00629 {
00630     /* IGP band type */
00631     short x; /* longitude/latitude (deg) */
00632     const short *y; /* latitudes/longitudes (deg) */
00633     unsigned char bits; /* IGP mask start bit */
00634     unsigned char bite; /* IGP mask end bit */
00635 } sbsigpband_t;
00636
00637 typedef struct
00638 {
00639     /* SBAS ionospheric corrections type */
00640     int iodi; /* IODI (issue of date ionos corr) */
00641     int nigp; /* number of igps */
00642     sbsigp_t igp[MAXNIGP]; /* ionospheric correction */
00643 } sbsion_t;
00644
00645 typedef struct
00646 {
00647     /* DGPS/GNSS correction type */
00648     gtime_t t0; /* correction time */
00649     double prc; /* pseudorange correction (PRC) (m) */
00650     double rrc; /* range rate correction (RRC) (m/s) */
00651     int iod; /* issue of data (IOD) */
00652     double udre; /* UDRE */
00653 } dgps_t;
00654
00655 typedef struct
00656 {
00657     /* SSR correction type */
00658     gtime_t t0[6]; /* epoch time (GPST) {eph,clk,hrcclk,ura,bias,pbias} */
00659     double udi[6]; /* SSR update interval (s) */
00660     int iod[6]; /* iod ssr {eph,clk,hrcclk,ura,bias,pbias} */
00661     int iode; /* issue of data */
00662     int iodicrc; /* issue of data crc for beidou/sbas */
00663     int ura; /* URA indicator */
00664     int refd; /* sat ref datum (0:ITRF,1:regional) */
00665     double ddep[3]; /* delta orbit {radial,along,cross} (m) */
00666     double ddep[3]; /* dot delta orbit {radial,along,cross} (m/s) */
00667     double dclk[3]; /* delta clock {c0,c1,c2} (m,m/s,m/s^2) */
00668     double hrcclk; /* high-rate clock correction (m) */
00669     float cbias[MAXCODE]; /* code biases (m) */
00670     double pbias[MAXCODE]; /* phase biases (m) */
00671     float stdpb[MAXCODE]; /* std-dev of phase biases (m) */
00672     double yaw_ang, yaw_rate; /* yaw angle and yaw rate (deg,deg/s) */
00673     unsigned char update; /* update flag (0:no update,1:update) */
00674 } ssr_t;
00675
00676 typedef struct
00677 {
00678     /* QZSS LEX message type */
00679     int prn; /* satellite PRN number */
00680     int type; /* message type */
00681     int alert; /* alert flag */
00682     unsigned char stat; /* signal tracking status */
00683     unsigned char snr; /* signal C/N0 (0.25 dBHz) */
00684     unsigned int ttt; /* tracking time (ms) */
00685     unsigned char msg[212]; /* LEX message data part 1695 bits */
00686 } lexmsg_t;
00687
00688 typedef struct
00689 {
00690     /* QZSS LEX messages type */
00691     int n, nmax; /* number of LEX messages and allocated */
00692     lexmsg_t *msgs; /* LEX messages */
00693 } lex_t;
00694
00695 typedef struct
00696 {
00697     /* QZSS LEX ephemeris type */
00698     gtime_t toe; /* epoch time (GPST) */
00699     gtime_t tof; /* message frame time (GPST) */
00700     int sat; /* satellite number */
00701     unsigned char health; /* signal health (L1,L2,L1C,L5,LEX) */
00702     unsigned char ura; /* URA index */

```

```

00702     double pos[3];           /* satellite position (m) */
00703     double vel[3];           /* satellite velocity (m/s) */
00704     double acc[3];           /* satellite acceleration (m/s2) */
00705     double jerk[3];          /* satellite jerk (m/s3) */
00706     double af0, afl;         /* satellite clock bias and drift (s,s/s) */
00707     double tgd;              /* TGD */
00708     double isc[8];           /* ISC */
00709 } lexeph_t;
00710
00711
00712 typedef struct
00713 {
00714     /* QZSS LEX ionosphere correction type */
00715     gtime_t t0;              /* epoch time (GPST) */
00716     double tspan;            /* valid time span (s) */
00717     double pos0[2];          /* reference position {lat,lon} (rad) */
00718     double coef[3][2];       /* coefficients lat x lon (3 x 2) */
00719 } lexion_t;
00720
00721 typedef struct
00722 {
00723     /* stec data type */
00724     gtime_t time;            /* time (GPST) */
00725     unsigned char sat;        /* satellite number */
00726     double ion;              /* slant ionos delay (m) */
00727     float std;               /* std-dev (m) */
00728     float azel[2];           /* azimuth/elevation (rad) */
00729     unsigned char flag;       /* fix flag */
00730 } stec_t;
00731
00732 typedef struct
00733 {
00734     /* trop data type */
00735     gtime_t time;            /* time (GPST) */
00736     double trp[3];           /* zenith tropos delay/gradient (m) */
00737     float std[3];            /* std-dev (m) */
00738 } trop_t;
00739
00740 typedef struct
00741 {
00742     /* ppp corrections type */
00743     int nsta;                 /* number of stations */
00744     char stas[MAXSTA][8];     /* station names */
00745     double rr[MAXSTA][3];     /* station ecef positions (m) */
00746     int ns[MAXSTA], nsmax[MAXSTA]; /* number of stec data */
00747     int nt[MAXSTA], ntmax[MAXSTA]; /* number of trop data */
00748     stec_t *stec[MAXSTA];     /* stec data */
00749     trop_t *trop[MAXSTA];     /* trop data */
00750 } pppcorr_t;
00751
00752 typedef struct
00753 {
00754     /* navigation data type */
00755     int n, nmax;              /* number of broadcast ephemeris */
00756     int ng, ngmax;            /* number of glonass ephemeris */
00757     int ns, nsmax;            /* number of sbas ephemeris */
00758     int ne, nemax;            /* number of precise ephemeris */
00759     int nc, ncmax;            /* number of precise clock */
00760     int na, namax;            /* number of almanac data */
00761     int nt, ntmax;            /* number of tec grid data */
00762     int nf, nfmax;            /* number of satellite fcb data */
00763     eph_t *eph;               /* GPS/QZS/GAL ephemeris */
00764     geeph_t *geeph;           /* GLONASS ephemeris */
00765     seeph_t *seeph;           /* SBAS ephemeris */
00766     peeph_t *peeph;           /* precise ephemeris */
00767     pclk_t *pclk;             /* precise clock */
00768     alm_t *alm;               /* almanac data */
00769     tec_t *tec;               /* tec grid data */
00770     fcbd_t *fcb;              /* satellite fcb data */
00771     erp_t erp;                /* earth rotation parameters */
00772     double utc_gps[4];         /* GPS delta-UTC parameters {A0,A1,T,W} */
00773     double utc_glo[4];         /* GLONASS UTC GPS time parameters */
00774     double utc_gal[4];         /* Galileo UTC GPS time parameters */
00775     double utc_qzs[4];         /* QZS UTC GPS time parameters */
00776     double utc_cmp[4];         /* BeiDou UTC parameters */
00777     double utc_irn[4];         /* IRNSS UTC parameters */
00778     double utc_sbs[4];         /* SBAS UTC parameters */
00779     double ion_gps[8];         /* GPS iono model parameters {a0,a1,a2,a3,b0,b1,b2,b3} */
00780     double ion_gal[4];         /* Galileo iono model parameters {ai0,ai1,ai2,0} */
00781     double ion_qzs[8];         /* QZSS iono model parameters {a0,a1,a2,a3,b0,b1,b2,b3} */
00782     double ion_cmp[8];         /* BeiDou iono model parameters {a0,a1,a2,a3,b0,b1,b2,b3} */
00783     double ion_irn[8];         /* IRNSS iono model parameters {a0,a1,a2,a3,b0,b1,b2,b3} */
00784     int leaps;                /* leap seconds (s) */
00785     double lam[MAXSAT][NFREQ]; /* carrier wave lengths (m) */
00786     double cbias[MAXSAT][3];   /* satellite dcb (0:p1-p2,1:p1-c1,2:p2-c2) (m) */
00787     double rbias[MAXRCV][2][3]; /* receiver dcb (0:p1-p2,1:p1-c1,2:p2-c2) (m) */
00788     double wlbias[MAXSAT];     /* wide-lane bias (cycle) */
00789     double glo_cpbias[4];       /* glonass code-phase bias {1C,1P,2C,2P} (m) */

```



```

00789     char glo_fcn[MAXPRNGLO + 1]; /* glonass frequency channel number + 8 */
00790     pcv_t pcvs[MAXSAT];          /* satellite antenna pcv */
00791     sbssat_t sbssat;             /* SBAS satellite corrections */
00792     sbsion_t sbsion[MAXBAND + 1]; /* SBAS ionosphere corrections */
00793     dgps_t dgps[MAXSAT];         /* DGPS corrections */
00794     ssr_t ssr[MAXSAT];           /* SSR corrections */
00795     lexeph_t lexeph[MAXSAT];      /* LEX ephemeris */
00796     lexion_t lexion;             /* LEX ionosphere correction */
00797     pppcorr_t pppcorr;           /* ppp corrections */
00798 } nav_t;
00799
00800
00801 typedef struct
00802 {
00803     char name[MAXANT];           /* station parameter type */
00804     char marker[MAXANT];         /* marker name */
00805     char antdes[MAXANT];         /* marker number */
00806     char antsno[MAXANT];         /* antenna descriptor */
00807     char rectype[MAXANT];        /* antenna serial number */
00808     char recver[MAXANT];         /* receiver type descriptor */
00809     char recsno[MAXANT];         /* receiver firmware version */
00810     int antsetup;                /* receiver serial number */
00811     int itrfr;                  /* antenna setup id */
00812     int deltype;                 /* ITRF realization year */
00813     double pos[3];               /* antenna delta type (0:enu,1:xyz) */
00814     double del[3];               /* station position (ecef) (m) */
00815     double hgt;                 /* antenna position delta (e/n/u or x/y/z) (m) */
00816     double hgt;                 /* antenna height (m) */
00817 } sta_t;
00818
00819 typedef struct
00820 {
00821     gtime_t time;                /* solution type */
00822     double rr[6];                /* time (GPST) */
00823     float qr[6];                 /* position/velocity (m|m/s) */
00824     double dtr[6];               /* {x,y,z,vx,vy,vz} or {e,n,u,ve,vn,vu} */
00825     unsigned char type;           /* position variance/covariance (m^2) */
00826     unsigned char stat;           /* {c_xx,c_yy,c_zz,c_xy,c_yz,c_zx} or */
00827     unsigned char ns;            /* {c_ee,c_nn,c_uu,c_en,c_nu,c_ue} */
00828     float age;                   /* receiver clock bias to time systems (s) */
00829     float ratio;                 /* type (0:xyz-ecef,1:enu-baseline) */
00830     float thres;                 /* solution status (SOLQ_???) */
00831     float thres;                 /* number of valid satellites */
00832     float thres;                 /* age of differential (s) */
00833     float thres;                 /* AR ratio factor for validation */
00834     float thres;                 /* AR ratio threshold for validation */
00835 } sol_t;
00836
00837 typedef struct
00838 {
00839     int n, nmax;                 /* solution buffer type */
00840     int cyclic;                  /* number of solution/max number of buffer */
00841     int start, end;              /* cyclic buffer flag */
00842     gtime_t time;                /* start/end index */
00843     sol_t *data;                 /* current solution time */
00844     double rb[3];                /* solution data */
00845     unsigned char buff[MAXSOLMSG + 1]; /* reference position {x,y,z} (ecef) (m) */
00846     int nb;                      /* message buffer */
00847     int nb;                      /* number of byte in message buffer */
00848 } solbuf_t;
00849
00850 typedef struct
00851 {
00852     gtime_t time;                /* solution status type */
00853     unsigned char sat;           /* time (GPST) */
00854     unsigned char frq;           /* satellite number */
00855     float az, el;                /* frequency (1:L1,2:L2,...) */
00856     float resp;                  /* azimuth/elevation angle (rad) */
00857     float resc;                  /* pseudorange residual (m) */
00858     unsigned char flag;          /* carrier-phase residual (m) */
00859     unsigned char snr;           /* flags: (vsat<5)+(slip<3)+fix */
00860     unsigned short lock;         /* signal strength (0.25 dBHz) */
00861     unsigned short outc;         /* lock counter */
00862     unsigned short slipc;        /* outage counter */
00863     unsigned short rejcc;        /* slip counter */
00864     unsigned short rejcc;        /* reject counter */
00865 } solstat_t;
00866
00867 typedef struct
00868 {
00869     int n, nmax;                 /* solution status buffer type */
00870     solstat_t *data;             /* number of solution/max number of buffer */
00871     solstat_t *data;             /* solution status data */
00872 } solstatbuf_t;
00873
00874 typedef struct
00875 {
00876     /* RTCM control struct type */

```



```

00876     int staid;                /* station id */
00877     int stah;                /* station health */
00878     int seqno;               /* sequence number for rtcm 2 or iods msm */
00879     int outtype;             /* output message type */
00880     gtime_t time;            /* message time */
00881     gtime_t time_s;          /* message start time */
00882     obs_t obs;               /* observation data (uncorrected) */
00883     nav_t nav;               /* satellite ephemerides */
00884     sta_t sta;               /* station parameters */
00885     dgps_t *dgps;            /* output of dgps corrections */
00886     ssr_t ssr[MAXSAT];       /* output of ssr corrections */
00887     char msg[128];           /* special message */
00888     char msgtype[256];        /* last message type */
00889     char msmttype[6][128];    /* msm signal types */
00890     int obsflag;             /* obs data complete flag (1:ok,0:not complete) */
00891     int ephsat;              /* update satellite of ephemeris */
00892     double cp[MAXSAT][NFREQ + NEXOBS]; /* carrier-phase measurement */
00893     unsigned short lock[MAXSAT][NFREQ + NEXOBS]; /* lock time */
00894     unsigned short loss[MAXSAT][NFREQ + NEXOBS]; /* loss of lock count */
00895     gtime_t lltime[MAXSAT][NFREQ + NEXOBS]; /* last lock time */
00896     int nbyte;               /* number of bytes in message buffer */
00897     int nbit;                /* number of bits in word buffer */
00898     int len;                 /* message length (bytes) */
00899     unsigned char buff[1200]; /* message buffer */
00900     unsigned int word;        /* word buffer for rtcm 2 */
00901     unsigned int nmsg2[100];  /* message count of RTCM 2 (1-99:1-99,0:other) */
00902     unsigned int nmsg3[400];  /* message count of RTCM 3
(1-299:1001-1299,300-399:2000-2099,0:other) */
00903     char opt[256];           /* RTCM dependent options */
00904 } rtcm_t;
00905
00906
00907 typedef struct
00908 {
00909     /* download url type */
00909     char type[32];           /* data type */
00910     char path[1024];         /* url path */
00911     char dir[1024];          /* local directory */
00912     double tint;             /* time interval (s) */
00913 } url_t;
00914
00915
00916 typedef struct
00917 {
00918     /* option type */
00918     const char *name;        /* option name */
00919     int format;              /* option format (0:int,1:double,2:string,3:enum) */
00920     void *var;               /* pointer to option variable */
00921     const char *comment;     /* option comment/enum labels/unit */
00922 } opt_t;
00923
00924
00925 typedef struct
00926 {
00927     /* extended receiver error model */
00927     int ena[4];              /* model enabled */
00928     double cerr[4][NFREQ * 2]; /* code errors (m) */
00929     double perr[4][NFREQ * 2]; /* carrier-phase errors (m) */
00930     double gpsglob[NFREQ];     /* gps-glonass h/w bias (m) */
00931     double gloicb[NFREQ];      /* glonass interchannel bias (m/fn) */
00932 } exterr_t;
00933
00934
00935 typedef struct
00936 {
00937     /* SNR mask type */
00937     int ena[2];              /* enable flag {rover,base} */
00938     double mask[NFREQ][9];    /* mask (dBHz) at 5,10,...85 deg */
00939 } snrmask_t;
00940
00941
00942 typedef struct
00943 {
00944     /* processing options type */
00944     int mode;                /* positioning mode (PMODE_???) */
00945     int soltype;              /* solution type (0:forward,1:backward,2:combined) */
00946     int nf;                   /* number of frequencies (1:L1,2:L1+L2,3:L1+L2+L5) */
00947     int navsys;               /* navigation system */
00948     double elmin;             /* elevation mask angle (rad) */
00949     snrmask_t snrmask;        /* SNR mask */
00950     int sateph;               /* satellite ephemeris/clock (EPHOPT_???) */
00951     int modear;               /* AR mode (0:off,1:continuous,2:instantaneous,3:fix and
hold,4:ppp-ar) */
00952     int glomodear;            /* GLONASS AR mode (0:off,1:on,2:auto cal,3:ext cal) */
00953     int bdsmodear;            /* BeiDou AR mode (0:off,1:on) */
00954     int maxout;               /* obs outage count to reset bias */
00955     int minlock;              /* min lock count to fix ambiguity */
00956     int minfix;               /* min fix count to hold ambiguity */
00957     int armaxiter;            /* max iteration to resolve ambiguity */
00958     int ionoopt;              /* ionosphere option (IONOOPT_???) */
00959     int tropopt;              /* troposphere option (TROPOPT_???) */
00960     int dynamics;            /* dynamics model (0:none,1:velociy,2:accel) */

```

```

00961     int tidecorr;          /* earth tide correction (0:off,1:solid,2:solid+otl+pole) */
00962     int niter;             /* number of filter iteration */
00963     int codesmooth;        /* code smoothing window size (0:none) */
00964     int intpref;           /* interpolate reference obs (for post mission) */
00965     int sbascorr;          /* SBAS correction options */
00966     int sbassatsel;        /* SBAS satellite selection (0:all) */
00967     int rovpos;            /* rover position for fixed mode */
00968     int refpos;            /* base position for relative mode */
00969                             /* (0:pos in prcopt, 1:average of single pos, */
00970                             /* 2:read from file, 3:rinex header, 4:rtcm pos) */
00971     double eratio[NFREQ];  /* code/phase error ratio */
00972     double err[5];         /* measurement error factor */
00973                             /* [0]:reserved */
00974                             /* [1-3]:error factor a/b/c of phase (m) */
00975                             /* [4]:doppler frequency (hz) */
00976     double std[3];         /* initial-state std [0]bias,[1]iono [2]trop */
00977     double prn[6];         /* process-noise std [0]bias,[1]iono [2]trop [3]acch [4]accv [5] pos
*/
00978     double sclkstab;       /* satellite clock stability (sec/sec) */
00979     double thresar[8];     /* AR validation threshold */
00980     double elmaskar;       /* elevation mask of AR for rising satellite (deg) */
00981     double elmaskhold;     /* elevation mask to hold ambiguity (deg) */
00982     double thresslip;      /* slip threshold of geometry-free phase (m) */
00983     double maxtdiff;       /* max difference of time (sec) */
00984     double maxinno;        /* reject threshold of innovation (m) */
00985     double maxgdp;         /* reject threshold of gdop */
00986     double baseline[2];    /* baseline length constraint {const,sigma} (m) */
00987     double ru[3];          /* rover position for fixed mode {x,y,z} (ecef) (m) */
00988     double rb[3];          /* base position for relative mode {x,y,z} (ecef) (m) */
00989     char anttype[2][MAXANT]; /* antenna types {rover,base} */
00990     double antdel[2][3];   /* antenna delta {(rov_e,rov_n,rov_u},{ref_e,ref_n,ref_u)} */
00991     pcv_t pcvr[2];         /* receiver antenna parameters {rov,base} */
00992     unsigned char exsats[MAXSAT]; /* excluded satellites (1:excluded,2:included) */
00993     int maxaveep;          /* max averaging epoches */
00994     int inirst;            /* initialize by restart */
00995     int outsingle;         /* output single by dgps/float/fix/ppp outage */
00996     char rxopt[2][256];    /* rinex options {rover,base} */
00997     int posopt[6];         /* positioning options */
00998     int syncsol;           /* solution sync mode (0:off,1:on) */
00999     double odisp[2][6 * 11]; /* ocean tide loading parameters {rov,base} */
01000     exterr_t exterr;       /* extended receiver error model */
01001     int freqopt;           /* disable L2-AR */
01002     char ppopt[256];       /* ppp option */
01003     bool bancroft_init;    /* enable Bancroft initialization for the first iteration of the PVT
computation */
01004 } prcopt_t;
01005
01006
01007 typedef struct
01008 {
01009     /* solution options type */
01010     int posf;              /* solution format (SOLF_???) */
01011     int times;             /* time system (TIMES_???) */
01012     int timef;             /* time format (0:sssss.s,1:yyyy/mm/dd hh:mm:ss.s) */
01013     int timeu;             /* time digits under decimal point */
01014     int degf;              /* latitude/longitude format (0:ddd.ddd,1:ddd mm ss) */
01015     int outhead;           /* output header (0:no,1:yes) */
01016     int outopt;            /* output processing options (0:no,1:yes) */
01017     int datum;             /* datum (0:WGS84,1:Tokyo) */
01018     int height;            /* height (0:ellipsoidal,1:geodetic) */
01019     int geoid;             /* geoid model (0:EGM96,1:JGD2000) */
01020     int solstatic;         /* solution of static mode (0:all,1:single) */
01021     int sstat;             /* solution statistics level (0:off,1:states,2:residuals) */
01022     int trace;             /* debug trace level (0:off,1-5:debug) */
01023     double nmeaintv[2];    /* nmea output interval (s) (<0:no,0:all) */
01024                             /* nmeaintv[0]:gprmc,gpgga,nmeaintv[1]:gpgsv */
01025     char sep[64];          /* field separator */
01026     char prog[64];         /* program name */
01027     double maxsolstd;      /* max std-dev for solution output (m) (0:all) */
01028 } solopt_t;
01029
01030 typedef struct
01031 {
01032     /* satellite status type */
01033     unsigned char sys;     /* navigation system */
01034     unsigned char vs;      /* valid satellite flag single */
01035     double azel[2];        /* azimuth/elevation angles {az,el} (rad) */
01036     double resp[NFREQ];    /* residuals of pseudorange (m) */
01037     double resc[NFREQ];    /* residuals of carrier-phase (m) */
01038     unsigned char vsat[NFREQ]; /* valid satellite flag */
01039     unsigned char snr[NFREQ]; /* signal strength (0.25 dBHz) */
01040     unsigned char fix[NFREQ]; /* ambiguity fix flag (1:fix,2:float,3:hold) */
01041     unsigned char slip[NFREQ]; /* cycle-slip flag */
01042     unsigned char half[NFREQ]; /* half-cycle valid flag */
01043     int lock[NFREQ];        /* lock counter of phase */
01044     unsigned int outc[NFREQ]; /* obs outage counter of phase */
01045     unsigned int slipc[NFREQ]; /* cycle-slip counter */
01046     unsigned int rejcc[NFREQ]; /* reject counter */

```

```

01046     double gf;                /* geometry-free phase L1-L2 (m) */
01047     double gf2;               /* geometry-free phase L1-L5 (m) */
01048     double mw;                /* MW-LC (m) */
01049     double phw;               /* phase windup (cycle) */
01050     gtime_t pt[2][NFREQ];     /* previous carrier-phase time */
01051     double ph[2][NFREQ];      /* previous carrier-phase observable (cycle) */
01052 } ssat_t;
01053
01054
01055 typedef struct
01056 {
01057     gtime_t epoch[4];          /* last epoch */
01058     int n[4];                  /* number of epochs */
01059     double LC[4];              /* linear combination average */
01060     double LCv[4];             /* linear combination variance */
01061     int fixcnt;                 /* fix count */
01062     char flags[MAXSAT];        /* fix flags */
01063 } ambc_t;
01064
01065
01066 typedef struct
01067 {
01068     sol_t sol;                 /* RTK control/result type */
01069     double rb[6];              /* RTK solution */
01070     int nx, na;                /* base position/velocity (ecef) (m|m/s) */
01071     double tt;                 /* number of float states/fixed states */
01072     double *x, *P;             /* time difference between current and previous (s) */
01073     double *xa, *Pa;           /* float states and their covariance */
01074     int nfix;                  /* fixed states and their covariance */
01075     ambc_t ambc[MAXSAT];       /* number of continuous fixes of ambiguity */
01076     ssat_t ssat[MAXSAT];       /* ambiguity control */
01077     int neb;                   /* satellite status */
01078     char errbuf[MAXERRMSG];    /* bytes in error message buffer */
01079     prcopt_t opt;              /* error message buffer */
01080 } rtk_t;
01081
01082
01083 typedef struct half_cyc_tag
01084 {
01085     unsigned char sat;         /* processing options */
01086     unsigned char freq;        /* half-cycle correction list type */
01087     unsigned char valid;       /* satellite number */
01088     char corr;                 /* frequency number (0:L1,1:L2,2:L5) */
01089     gtime_t ts, te;            /* half-cycle valid flag */
01090     struct half_cyc_tag *next; /* half-cycle corrected (x 0.5 cyc) */
01091 } half_cyc_t;
01092
01093
01094 typedef struct
01095 {
01096     int type;                  /* half-cycle time start, time end */
01097     int mode;                  /* stream type */
01098     int state;                 /* type (STR_???) */
01099     unsigned int inb, inr;      /* mode (STR_MODE_?) */
01100     unsigned int outb, outr;    /* state (-1:error,0:close,1:open) */
01101     unsigned int tick, tact;    /* input bytes/rate */
01102     unsigned int inbt, outbt;   /* output bytes/rate */
01103     lock_t lock;               /* tick/active tick */
01104     void *port;                /* input/output bytes at tick */
01105     char path[MAXSTRPATH];      /* lock flag */
01106     char msg[MAXSTRMSG];        /* type dependent port control struct */
01107 } stream_t;
01108
01109
01110 typedef struct
01111 {
01112     dev_t dev;                 /* serial control type */
01113     int error;                 /* serial device */
01114 } serial_t;
01115
01116
01117 struct file_t
01118 {
01119     FILE *fp = nullptr;        /* file control type */
01120     FILE *fp_tag = nullptr;    /* file pointer */
01121     FILE *fp_tmp = nullptr;    /* file pointer of tag file */
01122     FILE *fp_tag_tmp = nullptr; /* temporary file pointer for swap */
01123     std::string path;          /* temporary file pointer of tag file for swap */
01124     std::string openpath;      /* file path */
01125     int mode = 0;              /* open file path */
01126     int timetag;               /* file mode */
01127     int repmode = 0;           /* time tag flag (0:off,1:on) */
01128     int offset = 0;            /* replay mode (0:master,1:slave) */
01129     gtime_t time = {};         /* time offset (ms) for slave */
01130     gtime_t wtime = {};       /* start time */
01131     unsigned int tick = 0;     /* write time */
01132     unsigned int tick_f = 0;   /* start tick */

```

```

01133     unsigned int fpos = 0;          /* current file position */
01134     double start = 0;               /* start offset (s) */
01135     double speed = 0;               /* replay speed (time factor) */
01136     double swapintv = 0;            /* swap interval (hr) (0: no swap) */
01137     lock_t lock;                    /* lock flag */
01138 };
01139
01140
01141 typedef struct
01142 {
01143     int state;                       /* tcp control type */
01144     char saddr[256];                 /* state (0:close,1:wait,2:connect) */
01145     int port;                        /* address string */
01146     struct sockaddr_in addr;          /* port */
01147     socket_t sock;                   /* address resolved */
01148     int tcon;                        /* socket descriptor */
01149     unsigned int tact;                /* reconnect time (ms) (-1:never,0:now) */
01150     unsigned int tdis;                /* data active tick */
01151 } tcp_t;
01152
01153
01154 typedef struct
01155 {
01156     tcp_t svr;                       /* tcp server type */
01157     tcp_t cli[MAXCLI];                /* tcp server control */
01158 } tcpsvr_t;
01159
01160
01161 typedef struct
01162 {
01163     tcp_t svr;                       /* tcp client type */
01164     int toinact;                     /* tcp server control */
01165     int tirecon;                     /* inactive timeout (ms) (0:no timeout) */
01166 } tcpcli_t;
01167
01168
01169 typedef struct
01170 {
01171     int state;                       /* ntrip control type */
01172     int type;                        /* state (0:close,1:wait,2:connect) */
01173     int nb;                          /* type (0:server,1:client) */
01174     char url[256];                   /* response buffer size */
01175     char mntpnt[256];                /* url for proxy */
01176     char user[256];                  /* mountpoint */
01177     char passwd[256];                /* user */
01178     char str[NTRIP_MAXSTR];           /* password */
01179     unsigned char buff[NTRIP_MAXRSP]; /* mountpoint string for server */
01180     tcpcli_t *tcp;                   /* response buffer */
01181 } ntrip_t;
01182
01183
01184 typedef struct
01185 {
01186     int state;                       /* ftp download control type */
01187     int proto;                       /* state (0:close,1:download,2:complete,3:error) */
01188     int error;                       /* protocol (0:ftp,1:http) */
01189     /* 11:no temp dir,12:uncompact error */
01190     char addr[1024];                 /* error code (0:no error,1-10:wget error, */
01191     char file[1024];                 /* 11:no temp dir,12:uncompact error) */
01192     char user[256];                  /* download address */
01193     char passwd[256];                /* download file path */
01194     char local[1024];                /* user for ftp */
01195     int topts[4];                    /* password for ftp */
01196     gtime_t tnext;                   /* local file path */
01197     pthread_t thread;                /* time options {poff,tint,toff,tretry} (s) */
01198 } ftp_t;
01199
01200
01201 typedef struct
01202 {
01203     gtime_t time;                    /* receiver raw data control type */
01204     gtime_t tobs;                    /* message time */
01205     obs_t obs;                       /* observation data time */
01206     obs_t obuf;                      /* observation data */
01207     nav_t nav;                       /* observation data buffer */
01208     sta_t sta;                       /* satellite ephemerides */
01209     int ephsat;                      /* station parameters */
01210     sbmsg_t sbmsg;                   /* sat number of update ephemeris (0:no satellite) */
01211     char msgtype[256];                /* SBAS message */
01212     unsigned char subfrm[MAXSAT][380]; /* last message type */
01213     lexmsg_t lexmsg;                 /* subframe buffer */
01214     double lockt[MAXSAT][NFREQ + NEXOBS]; /* LEX message */
01215     double icpp[MAXSAT], off[MAXSAT], icpc; /* lock time (s) */
01216     double prCA[MAXSAT], dpCA[MAXSAT]; /* carrier params for ss2 */
01217     unsigned char halfc[MAXSAT][NFREQ + NEXOBS]; /* L1/CA pseudorange/doppler for javad */
01218     char freqn[MAXOBS];                /* half-cycle add flag */
01219     int nbyte;                       /* frequency number for javad */
01220 }
01221

```

```

01220     int len;                                /* message length (bytes) */
01221     int iod;                                /* issue of data */
01222     int tod;                                /* time of day (ms) */
01223     int tbase;                              /* time base (0:gpst,1:utc(usno),2:glonass,3:utc(su)
*/
01224     int flag;                              /* general purpose flag */
01225     int outtype;                           /* output message type */
01226     unsigned char buff[MAXRAWLEN];         /* message buffer */
01227     char opt[256];                         /* receiver dependent options */
01228     double receive_time;                  /* RT17: Reiceve time of week for week rollover
detection */
01229     unsigned int plen;                    /* RT17: Total size of packet to be read */
01230     unsigned int pbyte;                  /* RT17: How many packet bytes have been read so far
*/
01231     unsigned int page;                   /* RT17: Last page number */
01232     unsigned int reply;                 /* RT17: Current reply number */
01233     int week;                          /* RT17: week number */
01234     unsigned char pbuff[255 + 4 + 2];    /* RT17: Packet buffer */
01235 } raw_t;
01236
01237
01238 typedef struct
01239 {
01240     int state;                          /* RTK server type */
01241     int cycle;                          /* server state (0:stop,1:running) */
01242     int nmeacycle;                      /* processing cycle (ms) */
01243     int nmeareq;                        /* NMEA request cycle (ms) (0:no req) */
01244     double nmeapos[3];                 /* NMEA request position (ecef) (m) */
01245     int buffsize;                      /* input buffer size (bytes) */
01246     int format[3];                     /* input format {rov,base,corr} */
01247     solopt_t solopt[2];                 /* output solution options {sol1,sol2} */
01248     int navsel;                        /* ephemeris select (0:all,1:rover,2:base,3:corr) */
01249     int nsbs;                          /* number of sbas message */
01250     int nsol;                          /* number of solution buffer */
01251     rtk_t rtk;                         /* RTK control/result struct */
01252     int nb[3];                         /* bytes in input buffers {rov,base} */
01253     int nsb[2];                        /* bytes in soulution buffers */
01254     int npb[3];                        /* bytes in input peek buffers */
01255     unsigned char *buff[3];            /* input buffers {rov,base,corr} */
01256     unsigned char *sbuf[2];            /* output buffers {sol1,sol2} */
01257     unsigned char *pbuf[3];            /* peek buffers {rov,base,corr} */
01258     sol_t solbuf[MAXSOLBUF];           /* solution buffer */
01259     unsigned int nmsg[3][10];          /* input message counts */
01260     raw_t raw[3];                      /* receiver raw control {rov,base,corr} */
01261     rtkcm_t rtkcm[3];                  /* RTCM control {rov,base,corr} */
01262     gtime_t ftime[3];                  /* download time {rov,base,corr} */
01263     char files[3][MAXSTRPATH];         /* download paths {rov,base,corr} */
01264     obs_t obs[3][MAXOBSBUF];           /* observation data {rov,base,corr} */
01265     nav_t nav;                         /* navigation data */
01266     sbasmsg_t sbasmsg[MAXSBMSG];       /* SBAS message buffer */
01267     stream_t stream[8];                /* streams {rov,base,corr,sol1,sol2,logr,logb,logc} */
01268     stream_t *moni;                   /* monitor stream */
01269     unsigned int tick;                 /* start tick */
01270     pthread_t thread;                 /* server thread */
01271     int cputime;                       /* CPU time (ms) for a processing cycle */
01272     int prcout;                       /* missing observation data count */
01273     lock_t lock;                      /* lock flag */
01274 } rtksvr_t;
01275
01276 typedef struct
01277 {
01278     unsigned char iod;                 /* multi-signal-message header type */
01279     unsigned char time_s;              /* issue of data station */
01280     unsigned char clk_str;             /* cumulative session transmitting time */
01281     unsigned char clk_ext;             /* clock steering indicator */
01282     unsigned char smooth;             /* external clock indicator */
01283     unsigned char tint_s;             /* divergence free smoothing indicator */
01284     unsigned char nsat, nsig;          /* soothing interval */
01285     unsigned char sats[64];           /* number of satellites/signals */
01286     unsigned char sigs[32];           /* satellites */
01287     unsigned char cellmask[64];       /* signals */
01288     unsigned char cellmask[64];       /* cell mask */
01289 } msm_h_t;
01290
01291 const double CHISQR[100] = { /* chi-sqr(n) (alpha=0.001) */
01292     10.8, 13.8, 16.3, 18.5, 20.5, 22.5, 24.3, 26.1, 27.9, 29.6,
01293     31.3, 32.9, 34.5, 36.1, 37.7, 39.3, 40.8, 42.3, 43.8, 45.3,
01294     46.8, 48.3, 49.7, 51.2, 52.6, 54.1, 55.5, 56.9, 58.3, 59.7,
01295     61.1, 62.5, 63.9, 65.2, 66.6, 68.0, 69.3, 70.7, 72.1, 73.4,
01296     74.7, 76.0, 77.3, 78.6, 80.0, 81.3, 82.6, 84.0, 85.4, 86.7,
01297     88.0, 89.3, 90.6, 91.9, 93.3, 94.7, 96.0, 97.4, 98.7, 100,
01298     101, 102, 103, 104, 105, 107, 108, 109, 110, 112,
01299     113, 114, 115, 116, 118, 119, 120, 122, 123, 125,
01300     126, 127, 128, 129, 131, 132, 133, 134, 135, 137,
01301     138, 139, 140, 142, 143, 144, 145, 147, 148, 149};
01302
01303

```

```

01304 const double LAM_CARR[MAXFREQ] = { /* carrier wave length (m) */
01305     SPEED_OF_LIGHT_M_S / FREQ1, SPEED_OF_LIGHT_M_S / FREQ2, SPEED_OF_LIGHT_M_S / FREQ5,
01306     SPEED_OF_LIGHT_M_S / FREQ6, SPEED_OF_LIGHT_M_S / FREQ7,
01307     SPEED_OF_LIGHT_M_S / FREQ8, SPEED_OF_LIGHT_M_S / FREQ9};
01308 const int STRFMT_RTCM2 = 0; /* stream format: RTCM 2 */
01309 const int STRFMT_RTCM3 = 1; /* stream format: RTCM 3 */
01310 const int STRFMT_SP3 = 16; /* stream format: SP3 */
01311 const int STRFMT_RNXCLK = 17; /* stream format: RINEX CLK */
01312 const int STRFMT_SBAS = 18; /* stream format: SBAS messages */
01313 const int STRFMT_NMEA = 19; /* stream format: NMEA 0183 */
01314 // const solopt_t solopt_default; /* default solution output options */
01315
01316 const int MAXSTRRTK = 8; /* max number of stream in RTK server */
01317
01318 /** \} */
01319 /** \} */
01320 #endif // GNSS_SDR_RTKLIB_H

```

## 11.198 rtklib\_conversions.h File Reference

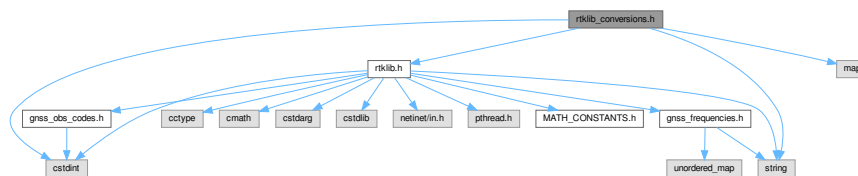
GNSS-SDR to RTKLIB data structures conversion functions.

```

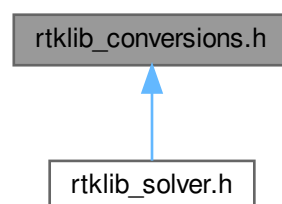
#include "rtklib.h"
#include <stdint>
#include <map>
#include <string>

```

Include dependency graph for rtklib\_conversions.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [HAS\\_clock\\_corrections](#)
- class [HAS\\_orbit\\_corrections](#)
- class [HAS\\_obs\\_corrections](#)

### Functions

- [eph\\_t eph\\_to\\_rtklib](#) (const [Galileo\\_Ephemeris](#) &gal\_eph)

- `eph_t eph_to_rtklib` (const [Galileo\\_Ephemeris](#) &gal\_eph, const std::map< int, [HAS\\_orbit\\_corrections](#) > &orbit\_correction\_map, const std::map< int, [HAS\\_clock\\_corrections](#) > &clock\_correction\_map)
- `eph_t eph_to_rtklib` (const [Gps\\_Ephemeris](#) &gps\_eph, bool pre\_2009\_file=false)
- `eph_t eph_to_rtklib` (const [Gps\\_Ephemeris](#) &gps\_eph, const std::map< int, [HAS\\_orbit\\_corrections](#) > &orbit\_correction\_map, const std::map< int, [HAS\\_clock\\_corrections](#) > &clock\_correction\_map, bool pre\_2009\_file=false)
- `eph_t eph_to_rtklib` (const [Gps\\_CNAV\\_Ephemeris](#) &gps\_cnav\_eph)
- `eph_t eph_to_rtklib` (const [Beidou\\_Dnav\\_Ephemeris](#) &bei\_eph)
- `alm_t alm_to_rtklib` (const [Gps\\_Almanac](#) &gps\_alm)
- `alm_t alm_to_rtklib` (const [Galileo\\_Almanac](#) &gal\_alm)
- `geph_t eph_to_rtklib` (const [Glonass\\_Gnav\\_Ephemeris](#) &glonass\_gnav\_eph, const [Glonass\\_Gnav\\_Utc\\_Model](#) &gnav\_clock\_model)

*Transforms a [Glonass\\_Gnav\\_Ephemeris](#) to its RTKLIB counterpart.*

- `obsd_t insert_obs_to_rtklib` (`obsd_t` &rtklib\_obs, const [Gnss\\_Synchro](#) &gnss\_synchro, const std::map< std::string, std::map< int, [HAS\\_obs\\_corrections](#) > > &has\_obs\_corr, int week, int band, bool pre\_2009\_file=false)
- `obsd_t insert_obs_to_rtklib` (`obsd_t` &rtklib\_obs, const [Gnss\\_Synchro](#) &gnss\_synchro, int week, int band, bool pre\_2009\_file=false)

### 11.198.1 Detailed Description

GNSS-SDR to RTKLIB data structures conversion functions.

Author

2017, Javier Arribas

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [rtklib\\_conversions.h](#).

---

## 11.199 rtklib\_conversions.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file rtklib_conversions.h
00003  * \brief GNSS-SDR to RTKLIB data structures conversion functions
00004  * \author 2017, Javier Arribas
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_RTKLIB_CONVERSIONS_H
00018 #define GNSS_SDR_RTKLIB_CONVERSIONS_H
00019
00020 #include "rtklib.h"
00021 #include <stdint>
00022 #include <map>
00023 #include <string>
00024
00025 /** \addtogroup PVT
00026  * \{ */
00027 /** \addtogroup RTKLIB_Library
00028  * \{ */
00029
00030
00031 class Beidou_Dnav_Ephemeris;
00032 class Galileo_Almanac;
00033 class Galileo_Ephemeris;
00034 class Glonass_Gnav_Ephemeris;
00035 class Glonass_Gnav_Utc_Model;
00036 class Gnss_Synchro;
```

```

00037 class Gps_Almanac;
00038 class Gps_CNAV_Ephemeris;
00039 class Gps_Ephemeris;
00040
00041 class HAS_clock_corrections
00042 {
00043 public:
00044     HAS_clock_corrections() = default;
00045     float clock_correction_m{};
00046     uint32_t valid_until{};
00047 };
00048
00049 class HAS_orbit_corrections
00050 {
00051 public:
00052     HAS_orbit_corrections() = default;
00053     float radial_m{};
00054     float in_track_m{};
00055     float cross_track_m{};
00056     uint32_t valid_until{};
00057     uint16_t iod{};
00058 };
00059
00060 class HAS_obs_corrections
00061 {
00062 public:
00063     HAS_obs_corrections() = default;
00064     float code_bias_m{};
00065     float phase_bias_cycle{};
00066 };
00067
00068
00069 eph_t eph_to_rtklib(const Galileo_Ephemeris& gal_eph);
00070
00071 eph_t eph_to_rtklib(const Galileo_Ephemeris& gal_eph,
00072     const std::map<int, HAS_orbit_corrections>& orbit_correction_map,
00073     const std::map<int, HAS_clock_corrections>& clock_correction_map);
00074
00075 eph_t eph_to_rtklib(const Gps_Ephemeris& gps_eph,
00076     bool pre_2009_file = false);
00077
00078 eph_t eph_to_rtklib(const Gps_Ephemeris& gps_eph,
00079     const std::map<int, HAS_orbit_corrections>& orbit_correction_map,
00080     const std::map<int, HAS_clock_corrections>& clock_correction_map,
00081     bool pre_2009_file = false);
00082
00083 eph_t eph_to_rtklib(const Gps_CNAV_Ephemeris& gps_cnav_eph);
00084 eph_t eph_to_rtklib(const Beidou_Dnav_Ephemeris& bei_eph);
00085
00086 alm_t alm_to_rtklib(const Gps_Almanac& gps_alm);
00087 alm_t alm_to_rtklib(const Galileo_Almanac& gal_alm);
00088
00089 /*!
00090 * \brief Transforms a Glonass_Gnav_Ephemeris to its RTKLIB counterpart
00091 * \param glonass_gnav_eph GLONASS GNAV Ephemeris structure
00092 * \return Ephemeris structure for RTKLIB parsing
00093 */
00094 gep_t eph_to_rtklib(const Glonass_Gnav_Ephemeris& glonass_gnav_eph, const Glonass_Gnav_Utc_Model&
    gnav_clock_model);
00095
00096 obsd_t insert_obs_to_rtklib(obsd_t& rtklib_obs,
00097     const Gnss_Synchro& gnss_synchro,
00098     const std::map<std::string, std::map<int, HAS_obs_corrections>& has_obs_corr,
00099     int week,
00100     int band,
00101     bool pre_2009_file = false);
00102
00103 obsd_t insert_obs_to_rtklib(obsd_t& rtklib_obs, const Gnss_Synchro& gnss_synchro, int week, int band,
    bool pre_2009_file = false);
00104
00105
00106 /** \} */
00107 /** \} */
00108 #endif // GNSS_SDR_RTKLIB_CONVERSIONS_H

```

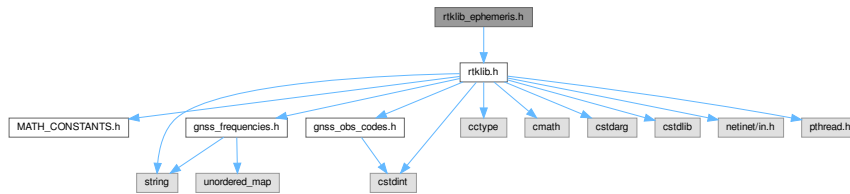
## 11.200 rtklib\_ephemeris.h File Reference

satellite ephemeris and clock functions



```
#include "rtklib.h"
```

Include dependency graph for rtklib\_ephemeris.h:



## Functions

- double **var\_uraeph** (int ura)
- double **var\_urassr** (int ura)
- void **alm2pos** (gtime\_t time, const alm\_t \*alm, double \*rs, double \*dts)
- double **eph2clk** (gtime\_t time, const eph\_t \*eph)
- void **eph2pos** (gtime\_t time, const eph\_t \*eph, double \*rs, double \*dts, double \*var)
- void **deq** (const double \*x, double \*xdot, const double \*acc)
- void **glorbit** (double t, double \*x, const double \*acc)
- double **geph2clk** (gtime\_t time, const geph\_t \*geph)
- void **geph2pos** (gtime\_t time, const geph\_t \*geph, double \*rs, double \*dts, double \*var)
- double **seph2clk** (gtime\_t time, const seph\_t \*seph)
- void **seph2pos** (gtime\_t time, const seph\_t \*seph, double \*rs, double \*dts, double \*var)
- eph\_t \* **seleph** (gtime\_t time, int sat, int iode, const nav\_t \*nav)
- geph\_t \* **selgeph** (gtime\_t time, int sat, int iode, const nav\_t \*nav)
- seph\_t \* **selseph** (gtime\_t time, int sat, const nav\_t \*nav)
- int **ephclk** (gtime\_t time, gtime\_t teph, int sat, const nav\_t \*nav, double \*dts)
- int **ephpos** (gtime\_t time, gtime\_t teph, int sat, const nav\_t \*nav, int iode, double \*rs, double \*dts, double \*var, int \*svh)
- int **satpos\_sbass** (gtime\_t time, gtime\_t teph, int sat, const nav\_t \*nav, double \*rs, double \*dts, double \*var, int \*svh)
- int **satpos\_ssr** (gtime\_t time, gtime\_t teph, int sat, const nav\_t \*nav, int opt, double \*rs, double \*dts, double \*var, int \*svh)
- int **satpos** (gtime\_t time, gtime\_t teph, int sat, int ephopt, const nav\_t \*nav, double \*rs, double \*dts, double \*var, int \*svh)
- void **satposs** (gtime\_t teph, const obsd\_t \*obs, int n, const nav\_t \*nav, int ephopt, double \*rs, double \*dts, double \*var, int \*svh)

### 11.200.1 Detailed Description

satellite ephemeris and clock functions

#### Authors

- 2007-2013, T. Takasu
- 2017, Javier Arribas
- 2017, Carles Fernandez

This is a derived work from RTKLIB <http://www.rtklib.com/> The original source code at <https://github.com/tomojitakasu/RTKLIB> is released under the BSD 2-clause license with an additional exclusive clause that does not apply here. This additional clause is reproduced below:

" The software package includes some companion executive binaries or shared libraries necessary to execute APs on Windows. These licenses succeed to the original ones of these software. "

Neither the executive binaries nor the shared libraries are required by, used or included in GNSS-SDR.

Copyright (C) 2007-2013, T. Takasu Copyright (C) 2017, Javier Arribas Copyright (C) 2017, Carles Fernandez All rights reserved.

SPDX-License-Identifier: BSD-2-Clause  
Definition in file [rtklib\\_ephemeris.h](#).

---

## 11.201 rtklib\_ephemeris.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file rtklib_ephemeris.h
00003  * \brief satellite ephemeris and clock functions
00004  * \authors <ul>
00005  *     <li> 2007-2013, T. Takasu
00006  *     <li> 2017, Javier Arribas
00007  *     <li> 2017, Carles Fernandez
00008  * </ul>
00009  *
00010  * This is a derived work from RTKLIB http://www.rtklib.com/
00011  * The original source code at https://github.com/tomojitakasu/RTKLIB is
00012  * released under the BSD 2-clause license with an additional exclusive clause
00013  * that does not apply here. This additional clause is reproduced below:
00014  *
00015  * " The software package includes some companion executive binaries or shared
00016  * libraries necessary to execute APs on Windows. These licenses succeed to the
00017  * original ones of these software. "
00018  *
00019  * Neither the executive binaries nor the shared libraries are required by, used
00020  * or included in GNSS-SDR.
00021  *
00022  * -----
00023  * Copyright (C) 2007-2013, T. Takasu
00024  * Copyright (C) 2017, Javier Arribas
00025  * Copyright (C) 2017, Carles Fernandez
00026  * All rights reserved.
00027  *
00028  * SPDX-License-Identifier: BSD-2-Clause
00029  *
00030  * -----
00031  */
00032
00033 #ifndef GNSS_SDR_RTKLIB_EPHEMERIS_H
00034 #define GNSS_SDR_RTKLIB_EPHEMERIS_H
00035
00036 #include "rtklib.h"
00037
00038 double var_uraeph(int ura);
00039 double var_urassr(int ura);
00040 void alm2pos(gtime_t time, const alm_t *alm, double *rs, double *dts);
00041 double eph2clk(gtime_t time, const eph_t *eph);
00042 void eph2pos(gtime_t time, const eph_t *eph, double *rs, double *dts,
00043             double *var);
00044 void deq(const double *x, double *xdot, const double *acc);
00045 void glorbit(double t, double *x, const double *acc);
00046 double gep2clk(gtime_t time, const gep_t *geph);
00047
00048 void gep2pos(gtime_t time, const gep_t *geph, double *rs, double *dts,
00049             double *var);
00050 double seph2clk(gtime_t time, const seph_t *seph);
00051 void seph2pos(gtime_t time, const seph_t *seph, double *rs, double *dts,
00052             double *var);
00053 eph_t *seleph(gtime_t time, int sat, int iode, const nav_t *nav);
00054 gep_t *selgeph(gtime_t time, int sat, int iode, const nav_t *nav);
00055 seph_t *selseph(gtime_t time, int sat, const nav_t *nav);
00056 int ephclk(gtime_t time, gtime_t teph, int sat, const nav_t *nav,
00057           double *dts);
00058 // satellite position and clock by broadcast ephemeris
00059 int ephpos(gtime_t time, gtime_t teph, int sat, const nav_t *nav,
00060           int iode, double *rs, double *dts, double *var, int *svh);
00061 int satpos_sbas(gtime_t time, gtime_t teph, int sat, const nav_t *nav,
00062               double *rs, double *dts, double *var, int *svh);
00063 int satpos_ssr(gtime_t time, gtime_t teph, int sat, const nav_t *nav,
00064               int opt, double *rs, double *dts, double *var, int *svh);
00065
00066 int satpos(gtime_t time, gtime_t teph, int sat, int ephopt,
00067           const nav_t *nav, double *rs, double *dts, double *var,
00068           int *svh);
00069 void satpos(gtime_t teph, const obsd_t *obs, int n, const nav_t *nav,
00070           int ephopt, double *rs, double *dts, double *var, int *svh);
00071
00072 #endif // GNSS_SDR_RTKLIB_EPHEMERIS_H

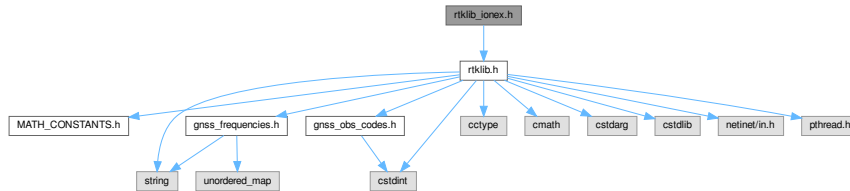
```

## 11.202 rtklib\_ionex.h File Reference

ionex functions

```
#include "rtklib.h"
```

Include dependency graph for rtklib\_ionex.h:



### Functions

- int **getindex** (double value, const double \*range)
- int **nitem** (const double \*range)
- int **dataindex** (int i, int j, int k, const int \*ndata)
- **tec\_t** \* **addtec** (const double \*lats, const double \*lons, const double \*hgts, double rb, **nav\_t** \*nav)
- void **readionexdcb** (FILE \*fp, double \*dcb, double \*rms)
- double **readionexh** (FILE \*fp, double \*lats, double \*lons, double \*hgts, double \*rb, double \*nexp, double \*dcb, double \*rms)
- int **readionexb** (FILE \*fp, const double \*lats, const double \*lons, const double \*hgts, double rb, double nexp, **nav\_t** \*nav)
- void **combtec** (**nav\_t** \*nav)
- void **readtec** (const char \*file, **nav\_t** \*nav, int opt)
- int **interptec** (const **tec\_t** \*tec, int k, const double \*posp, double \*value, double \*rms)
- int **iondelay** (**gtime\_t** time, const **tec\_t** \*tec, const double \*pos, const double \*azel, int opt, double \*delay, double \*var)
- int **iontec** (**gtime\_t** time, const **nav\_t** \*nav, const double \*pos, const double \*azel, int opt, double \*delay, double \*var)

### Variables

- const double **VAR\_NOTEC** = 30.0 \* 30.0
- const double **MIN\_EL** = 0.0
- const double **MIN\_HGT** = -1000.0

### 11.202.1 Detailed Description

ionex functions

#### Authors

- 2007-2013, T. Takasu
- 2017, Javier Arribas
- 2017, Carles Fernandez

This is a derived work from RTKLIB <http://www.rtklib.com/> The original source code at <https://github.com/tomojitakasu/RTKLIB> is released under the BSD 2-clause license with an additional exclusive clause that does not apply here. This additional clause is reproduced below:

" The software package includes some companion executive binaries or shared libraries necessary to execute APs on Windows. These licenses succeed to the original ones of these software. "

Neither the executive binaries nor the shared libraries are required by, used or included in GNSS-SDR.

Copyright (C) 2007-2013, T. Takasu Copyright (C) 2017, Javier Arribas Copyright (C) 2017, Carles Fernandez All rights reserved.

SPDX-License-Identifier: BSD-2-Clause

References: [1] S.Schear, W.Gurtner and J.Feltens, IONEX: The IONosphere Map EXchange Format Version 1, February 25, 1998 [2] S.Schaer, R.Markus, B.Gerhard and A.S.Timon, Daily Global Ionosphere Maps based on GPS Carrier Phase Data Routinely produced by CODE Analysis Center, Proceeding of the IGS Analysis Center Workshop, 1996

Definition in file [rtklib\\_ionex.h](#).

## 11.202.2 Variable Documentation

### 11.202.2.1 MIN\_EL

```
const double MIN_EL = 0.0
```

Definition at line 46 of file [rtklib\\_ionex.h](#).

### 11.202.2.2 MIN\_HGT

```
const double MIN_HGT = -1000.0
```

Definition at line 47 of file [rtklib\\_ionex.h](#).

### 11.202.2.3 VAR\_NOTEC

```
const double VAR_NOTEC = 30.0 * 30.0
```

Definition at line 45 of file [rtklib\\_ionex.h](#).

## 11.203 rtklib\_ionex.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file rtklib_ionex.h
00003  * \brief ionex functions
00004  * \authors <ul>
00005  *         <li> 2007-2013, T. Takasu
00006  *         <li> 2017, Javier Arribas
00007  *         <li> 2017, Carles Fernandez
00008  *         </ul>
00009  *
00010  * This is a derived work from RTKLIB http://www.rtklib.com/
00011  * The original source code at https://github.com/tomojitakasu/RTKLIB is
00012  * released under the BSD 2-clause license with an additional exclusive clause
00013  * that does not apply here. This additional clause is reproduced below:
00014  *
00015  * " The software package includes some companion executive binaries or shared
00016  * libraries necessary to execute APs on Windows. These licenses succeed to the
00017  * original ones of these software. "
00018  *
00019  * Neither the executive binaries nor the shared libraries are required by, used
00020  * or included in GNSS-SDR.
00021  *
00022  * -----
00023  * Copyright (C) 2007-2013, T. Takasu
00024  * Copyright (C) 2017, Javier Arribas
00025  * Copyright (C) 2017, Carles Fernandez
00026  * All rights reserved.
00027  *
00028  * SPDX-License-Identifier: BSD-2-Clause
00029  *
00030  * References:
00031  * [1] S.Schear, W.Gurtner and J.Feltens, IONEX: The IONosphere Map EXchange
00032  *      Format Version 1, February 25, 1998
00033  * [2] S.Schaer, R.Markus, B.Gerhard and A.S.Timon, Daily Global Ionosphere
00034  *      Maps based on GPS Carrier Phase Data Routinely produced by CODE
00035  *      Analysis Center, Proceeding of the IGS Analysis Center Workshop, 1996
00036  *
00037  * -----
00038  */
00039
00040 #ifndef GNSS_SDR_RTKLIB_IONEX_H
00041 #define GNSS_SDR_RTKLIB_IONEX_H
00042
00043 #include "rtklib.h"
00044
00045 const double VAR_NOTEC = 30.0 * 30.0; /* variance of no tec */
00046 const double MIN_EL = 0.0;           /* min elevation angle (rad) */
00047 const double MIN_HGT = -1000.0;      /* min user height (m) */
```

```

00048
00049 int getindex(double value, const double *range);
00050
00051 int nitem(const double *range);
00052 int dataindex(int i, int j, int k, const int *ndata);
00053 tec_t *addtec(const double *lats, const double *lons, const double *hgts,
00054   double rb, nav_t *nav);
00055 void readionexdcb(FILE *fp, double *dcb, double *rms);
00056 double readionexh(FILE *fp, double *lats, double *lons, double *hgts,
00057   double *rb, double *nexp, double *dcb, double *rms);
00058 int readionexb(FILE *fp, const double *lats, const double *lons,
00059   const double *hgts, double rb, double nexp, nav_t *nav);
00060 void combtec(nav_t *nav);
00061 void readtec(const char *file, nav_t *nav, int opt);
00062 int interptec(const tec_t *tec, int k, const double *posp, double *value,
00063   double *rms);
00064
00065 int iondelay(gtime_t time, const tec_t *tec, const double *pos,
00066   const double *azel, int opt, double *delay, double *var);
00067 int iontec(gtime_t time, const nav_t *nav, const double *pos,
00068   const double *azel, int opt, double *delay, double *var);
00069
00070 #endif // GNSS_SDR_RTKLIB_IONEX_H

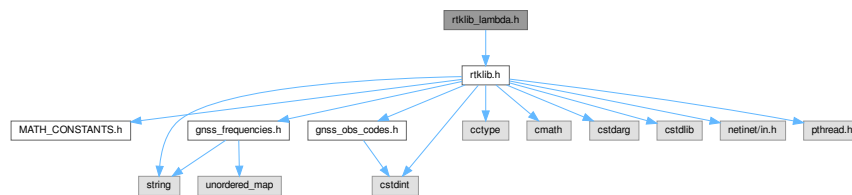
```

## 11.204 rtklib\_lambda.h File Reference

Integer ambiguity resolution.

#include "rtklib.h"

Include dependency graph for rtklib\_lambda.h:



### Macros

- #define [SGN\\_LAMBDA](#)(x)
- #define [ROUND\\_LAMBDA](#)(x)
- #define [SWAP\\_LAMBDA](#)(x, y)

### Functions

- int **LD** (int n, const double \*Q, double \*L, double \*D)
- void **gauss** (int n, double \*L, double \*Z, int i, int j)
- void **perm** (int n, double \*L, double \*D, int j, double del, double \*Z)
- void **reduction** (int n, double \*L, double \*D, double \*Z)
- int **search** (int n, int m, const double \*L, const double \*D, const double \*zs, double \*zn, double \*s)
- int **lambda** (int n, int m, const double \*a, const double \*Q, double \*F, double \*s)
- int **lambda\_reduction** (int n, const double \*Q, double \*Z)
- int **lambda\_search** (int n, int m, const double \*a, const double \*Q, double \*F, double \*s)

### Variables

- const int [LOOPMAX](#) = 10000

### 11.204.1 Detailed Description

Integer ambiguity resolution.

## Authors

- 2007-2008, T. Takasu
- 2017, Javier Arribas
- 2017, Carles Fernandez

This is a derived work from RTKLIB <http://www.rtklib.com/> The original source code at <https://github.com/tomojitakasu/RTKLIB> is released under the BSD 2-clause license with an additional exclusive clause that does not apply here. This additional clause is reproduced below:

" The software package includes some companion executive binaries or shared libraries necessary to execute APs on Windows. These licenses succeed to the original ones of these software. "

Neither the executive binaries nor the shared libraries are required by, used or included in GNSS-SDR.

Copyright (C) 2007-2008, T. Takasu Copyright (C) 2017, Javier Arribas Copyright (C) 2017, Carles Fernandez All rights reserved.

SPDX-License-Identifier: BSD-2-Clause

References: [1] P.J.G.Teunissen, The least-square ambiguity decorrelation adjustment: a method for fast GPS ambiguity estimation, J.Geodesy, Vol.70, 65-82, 1995 [2] X.-W.Chang, X.Yang, T.Zhou, MLAMBDA: A modified LAMBDA method for integer least-squares estimation, J.Geodesy, Vol.79, 552-565, 2005

Definition in file [rtklib\\_lambda.h](#).

## 11.204.2 Macro Definition Documentation

### 11.204.2.1 ROUND\_LAMBDA

```
#define ROUND_LAMBDA(  
    x)
```

#### Value:

```
(floor((x) + 0.5))
```

Definition at line 49 of file [rtklib\\_lambda.h](#).

### 11.204.2.2 SGN\_LAMBDA

```
#define SGN_LAMBDA(  
    x)
```

#### Value:

```
((x) <= 0.0 ? -1.0 : 1.0)
```

Definition at line 48 of file [rtklib\\_lambda.h](#).

### 11.204.2.3 SWAP\_LAMBDA

```
#define SWAP_LAMBDA(  
    x,  
    y)
```

#### Value:

```
do  
{  
    double tmp_  
    tmp_ = x;  
    x = y;  
    y = tmp_  
}  
while (0)
```

Definition at line 50 of file [rtklib\\_lambda.h](#).

## 11.204.3 Variable Documentation

### 11.204.3.1 LOOPMAX

```
const int LOOPMAX = 10000
```

Definition at line 47 of file [rtklib\\_lambda.h](#).

## 11.205 rtklib\_lambda.h

[Go to the documentation of this file.](#)

```

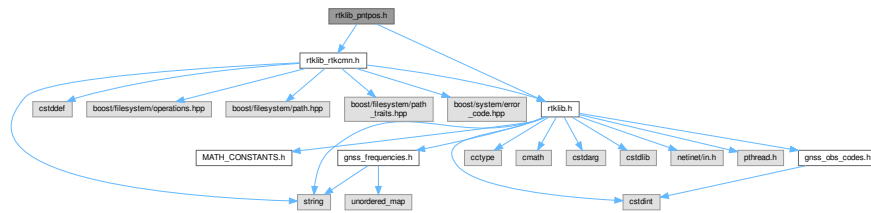
00001  /*!
00002  * \file rtklib_lambda.h
00003  * \brief Integer ambiguity resolution
00004  * \authors <ul>
00005  *         <li> 2007-2008, T. Takasu
00006  *         <li> 2017, Javier Arribas
00007  *         <li> 2017, Carles Fernandez
00008  *         </ul>
00009  *
00010  * This is a derived work from RTKLIB http://www.rtklib.com/
00011  * The original source code at https://github.com/tomojitakasu/RTKLIB is
00012  * released under the BSD 2-clause license with an additional exclusive clause
00013  * that does not apply here. This additional clause is reproduced below:
00014  *
00015  * " The software package includes some companion executive binaries or shared
00016  * libraries necessary to execute APs on Windows. These licenses succeed to the
00017  * original ones of these software. "
00018  *
00019  * Neither the executive binaries nor the shared libraries are required by, used
00020  * or included in GNSS-SDR.
00021  *
00022  * -----
00023  * Copyright (C) 2007-2008, T. Takasu
00024  * Copyright (C) 2017, Javier Arribas
00025  * Copyright (C) 2017, Carles Fernandez
00026  * All rights reserved.
00027  *
00028  * SPDX-License-Identifier: BSD-2-Clause
00029  *
00030  * References:
00031  * [1] P.J.G.Teunissen, The least-square ambiguity decorrelation adjustment:
00032  *      a method for fast GPS ambiguity estimation, J.Geodesy, Vol.70, 65-82,
00033  *      1995
00034  * [2] X.-W.Chang, X.Yang, T.Zhou, MLAMBDA: A modified LAMBDA method for
00035  *      integer least-squares estimation, J.Geodesy, Vol.79, 552-565, 2005
00036  *
00037  * -----
00038  */
00039
00040 #ifndef GNSS_SDR_RTKLIB_LAMBDA_H
00041 #define GNSS_SDR_RTKLIB_LAMBDA_H
00042
00043
00044 #include "rtklib.h"
00045
00046 /* constants/macros -----*/
00047 const int LOOPMAX = 10000; /* maximum count of search loop */
00048 #define SGN_LAMBDA(x) ((x) <= 0.0 ? -1.0 : 1.0)
00049 #define ROUND_LAMBDA(x) (floor((x) + 0.5))
00050 #define SWAP_LAMBDA(x, y) \
00051     do \
00052     { \
00053         double tmp_; \
00054         tmp_ = x; \
00055         x = y; \
00056         y = tmp_; \
00057     } \
00058     while (0)
00059
00060 int LD(int n, const double *Q, double *L, double *D);
00061 void gauss(int n, double *L, double *Z, int i, int j);
00062 void perm(int n, double *L, double *D, int j, double del, double *Z);
00063 void reduction(int n, double *L, double *D, double *Z);
00064 int search(int n, int m, const double *L, const double *D,
00065           const double *zs, double *zn, double *s);
00066
00067 int lambda(int n, int m, const double *a, const double *Q, double *F, double *s);
00068
00069 int lambda_reduction(int n, const double *Q, double *Z);
00070
00071 int lambda_search(int n, int m, const double *a, const double *Q,
00072                 double *F, double *s);
00073
00074
00075 #endif

```

## 11.206 rtklib\_pntpos.h File Reference

standard code-based positioning

```
#include "rtklib.h"
#include "rtklib_rtkcmn.h"
Include dependency graph for rtklib_pntpos.h:
```



## Functions

- double **varerr** (const [prcopt\\_t](#) \*opt, double el, int sys)
- double **gettgd** (int sat, const [nav\\_t](#) \*nav)
- double **getiscl1** (int sat, const [nav\\_t](#) \*nav)
- double **getiscl2** (int sat, const [nav\\_t](#) \*nav)
- double **getiscl5i** (int sat, const [nav\\_t](#) \*nav)
- double **getiscl5q** (int sat, const [nav\\_t](#) \*nav)
- double **prange** (const [obsd\\_t](#) \*obs, const [nav\\_t](#) \*nav, const double \*azel, int iter, const [prcopt\\_t](#) \*opt, double \*var)
- int **ionocorr** ([gtime\\_t](#) time, const [nav\\_t](#) \*nav, int sat, const double \*pos, const double \*azel, int ionoopt, double \*ion, double \*var)
- int **tropcorr** ([gtime\\_t](#) time, const [nav\\_t](#) \*nav, const double \*pos, const double \*azel, int tropopt, double \*trp, double \*var)
- int **rescode** (int iter, const [obsd\\_t](#) \*obs, int n, const double \*rs, const double \*dts, const double \*vare, const int \*svh, const [nav\\_t](#) \*nav, const double \*x, const [prcopt\\_t](#) \*opt, double \*v, double \*H, double \*var, double \*azel, int \*vsat, double \*resp, int \*ns)
- int **valsol** (const double \*azel, const int \*vsat, int n, const [prcopt\\_t](#) \*opt, const double \*v, int nv, int nx, char \*msg)
- int **estpos** (const [obsd\\_t](#) \*obs, int n, const double \*rs, const double \*dts, const double \*vare, const int \*svh, const [nav\\_t](#) \*nav, const [prcopt\\_t](#) \*opt, [sol\\_t](#) \*sol, double \*azel, int \*vsat, double \*resp, char \*msg)
- int **raim\_fde** (const [obsd\\_t](#) \*obs, int n, const double \*rs, const double \*dts, const double \*vare, const int \*svh, const [nav\\_t](#) \*nav, const [prcopt\\_t](#) \*opt, [sol\\_t](#) \*sol, double \*azel, int \*vsat, double \*resp, char \*msg)
- int **resdop** (const [obsd\\_t](#) \*obs, int n, const double \*rs, const double \*dts, const [nav\\_t](#) \*nav, const double \*rr, const double \*x, const double \*azel, const int \*vsat, double \*v, double \*H)
- void **estvel** (const [obsd\\_t](#) \*obs, int n, const double \*rs, const double \*dts, const [nav\\_t](#) \*nav, const [prcopt\\_t](#) \*opt, [sol\\_t](#) \*sol, const double \*azel, const int \*vsat)
- int **pntpos** (const [obsd\\_t](#) \*obs, int n, const [nav\\_t](#) \*nav, const [prcopt\\_t](#) \*opt, [sol\\_t](#) \*sol, double \*azel, [ssat\\_t](#) \*ssat, char \*msg)

*single-point positioning compute receiver position, velocity, clock bias by single-point positioning with pseudorange and doppler observables* args : [obsd\\_t](#) \*obs I observation data int n I number of observation data [nav\\_t](#) \*nav I navigation data [prcopt\\_t](#) \*opt I processing options [sol\\_t](#) \*sol IO solution double \*azel IO azimuth/elevation angle (rad) (NULL: no output) [ssat\\_t](#) \*ssat IO satellite status (NULL: no output) char \*msg O error message for error exit return : status(1:ok,0:error) notes : assuming sbas-gps, galileo-gps, qzss-gps, compass-gps time offset and receiver bias are negligible (only involving glonass-gps time offset and receiver bias)

## Variables

- const int **NX** = 4 + 3
- const int **MAXITER** = 10  
*max number of iteration for point pos*
- const double **ERR\_ION** = 5.0  
*ionospheric delay std (m)*



- const double `ERR_TROP` = 3.0  
tropospheric delay std (m)

## 11.206.1 Detailed Description

standard code-based positioning

### Authors

- 2007-2013, T. Takasu
- 2017, Javier Arribas
- 2017, Carles Fernandez

This is a derived work from RTKLIB <http://www.rtklib.com/> The original source code at <https://github.com/tomohitakasu/RTKLIB> is released under the BSD 2-clause license with an additional exclusive clause that does not apply here. This additional clause is reproduced below:

" The software package includes some companion executive binaries or shared libraries necessary to execute APs on Windows. These licenses succeed to the original ones of these software. "

Neither the executive binaries nor the shared libraries are required by, used or included in GNSS-SDR.

Copyright (C) 2007-2013, T. Takasu Copyright (C) 2017, Javier Arribas Copyright (C) 2017, Carles Fernandez All rights reserved.

SPDX-License-Identifier: BSD-2-Clause

Definition in file [rtklib\\_pntpos.h](#).

## 11.206.2 Function Documentation

### 11.206.2.1 pntpos()

```
int pntpos (
    const obsd_t * obs,
    int n,
    const nav_t * nav,
    const prcopt_t * opt,
    sol_t * sol,
    double * azel,
    ssat_t * ssat,
    char * msg)
```

single-point positioning compute receiver position, velocity, clock bias by single-point positioning with pseudorange and doppler observables args : `obsd_t` \*obs I observation data `int n` I number of observation data `nav_t` \*nav I navigation data `prcopt_t` \*opt I processing options `sol_t` \*sol IO solution double \*azel IO azimuth/elevation angle (rad) (NULL: no output) `ssat_t` \*ssat IO satellite status (NULL: no output) `char *msg` O error message for error exit return : status(1:ok,0:error) notes : assuming sbas-gps, galileo-gps, qzss-gps, compass-gps time offset and receiver bias are negligible (only involving glonass-gps time offset and receiver bias)

## 11.206.3 Variable Documentation

### 11.206.3.1 ERR\_ION

```
const double ERR_ION = 5.0
ionospheric delay std (m)
Definition at line 42 of file rtklib\_pntpos.h.
```

### 11.206.3.2 ERR\_TROP

```
const double ERR_TROP = 3.0
tropospheric delay std (m)
Definition at line 43 of file rtklib\_pntpos.h.
```

### 11.206.3.3 MAXITR

```
const int MAXITR = 10
```

max number of iteration for point pos

Definition at line 41 of file [rtklib\\_pntpos.h](#).

### 11.206.3.4 NX

```
const int NX = 4 + 3
```

## 11.206.4 of estimated parameters

Definition at line 40 of file [rtklib\\_pntpos.h](#).

## 11.207 rtklib\_pntpos.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file rtklib_pntpos.h
00003  * \brief standard code-based positioning
00004  * \authors <ul>
00005  *         <li> 2007-2013, T. Takasu
00006  *         <li> 2017, Javier Arribas
00007  *         <li> 2017, Carles Fernandez
00008  *         </ul>
00009  *
00010  * This is a derived work from RTKLIB http://www.rtklib.com/
00011  * The original source code at https://github.com/tomojitakasu/RTKLIB is
00012  * released under the BSD 2-clause license with an additional exclusive clause
00013  * that does not apply here. This additional clause is reproduced below:
00014  *
00015  * " The software package includes some companion executive binaries or shared
00016  * libraries necessary to execute APs on Windows. These licenses succeed to the
00017  * original ones of these software. "
00018  *
00019  * Neither the executive binaries nor the shared libraries are required by, used
00020  * or included in GNSS-SDR.
00021  *
00022  * -----
00023  * Copyright (C) 2007-2013, T. Takasu
00024  * Copyright (C) 2017, Javier Arribas
00025  * Copyright (C) 2017, Carles Fernandez
00026  * All rights reserved.
00027  *
00028  * SPDX-License-Identifier: BSD-2-Clause
00029  *
00030  * -----
00031  */
00032
00033 #ifndef GNSS_SDR_RTKLIB_PNTPOS_H
00034 #define GNSS_SDR_RTKLIB_PNTPOS_H
00035
00036 #include "rtklib.h"
00037 #include "rtklib_rtkcmn.h"
00038
00039 /* constants -----*/
00040 const int NX = 4 + 3;          //!< # of estimated parameters
00041 const int MAXITR = 10;        //!< max number of iteration for point pos
00042 const double ERR_ION = 5.0;    //!< ionospheric delay std (m)
00043 const double ERR_TROP = 3.0;   //!< tropospheric delay std (m)
00044
00045
00046 /* pseudorange measurement error variance -----*/
00047 double varerr(const prcopt_t *opt, double el, int sys);
00048
00049 /* get tgd parameter (m) -----*/
00050 double gettgd(int sat, const nav_t *nav);
00051
00052 /* get isc parameter (m) -----*/
00053 double getisc1(int sat, const nav_t *nav);
00054 double getisc2(int sat, const nav_t *nav);
00055 double getisc15i(int sat, const nav_t *nav);
00056 double getisc15q(int sat, const nav_t *nav);
00057
00058 /* pseudorange with code bias correction -----*/
00059 double prange(const obsd_t *obs, const nav_t *nav, const double *azel,
00060              int iter, const prcopt_t *opt, double *var);
00061
```

```

00062 /* ionospheric correction -----*/
00063 * compute ionospheric correction
00064 * args : gtime_t time      I   time
00065 *         nav_t *nav        I   navigation data
00066 *         int  sat          I   satellite number
00067 *         double *pos       I   receiver position {lat,lon,h} (rad|m)
00068 *         double *azel      I   azimuth/elevation angle {az,el} (rad)
00069 *         int  ionoopt      I   ionospheric correction option (IONOOPT_???)
00070 *         double *ion       O   ionospheric delay (L1) (m)
00071 *         double *var       O   ionospheric delay (L1) variance (m^2)
00072 * return : status(1:ok,0:error)
00073 *-----*/
00074 int ionocorr(gtime_t time, const nav_t *nav, int sat, const double *pos,
00075             const double *azel, int ionoopt, double *ion, double *var);
00076 /* tropospheric correction -----*/
00077 * compute tropospheric correction
00078 * args : gtime_t time      I   time
00079 *         nav_t *nav        I   navigation data
00080 *         double *pos       I   receiver position {lat,lon,h} (rad|m)
00081 *         double *azel      I   azimuth/elevation angle {az,el} (rad)
00082 *         int  tropopt      I   tropospheric correction option (TROPOPT_???)
00083 *         double *trp       O   tropospheric delay (m)
00084 *         double *var       O   tropospheric delay variance (m^2)
00085 * return : status(1:ok,0:error)
00086 *-----*/
00087 int tropcorr(gtime_t time, const nav_t *nav, const double *pos,
00088             const double *azel, int tropopt, double *trp, double *var);
00089
00090 /* pseudorange residuals -----*/
00091 int rescode(int iter, const obsd_t *obs, int n, const double *rs,
00092             const double *dts, const double *vare, const int *svh,
00093             const nav_t *nav, const double *x, const prcopt_t *opt,
00094             double *v, double *H, double *var, double *azel, int *vsat,
00095             double *resp, int *ns);
00096
00097 /* validate solution -----*/
00098 int valsol(const double *azel, const int *vsat, int n,
00099             const prcopt_t *opt, const double *v, int nv, int nx,
00100             char *msg);
00101
00102 /* estimate receiver position -----*/
00103 int estpos(const obsd_t *obs, int n, const double *rs, const double *dts,
00104             const double *vare, const int *svh, const nav_t *nav,
00105             const prcopt_t *opt, sol_t *sol, double *azel, int *vsat,
00106             double *resp, char *msg);
00107
00108 /* raim fde (failure detection and exclusion) -----*/
00109 int raim_fde(const obsd_t *obs, int n, const double *rs,
00110             const double *dts, const double *vare, const int *svh,
00111             const nav_t *nav, const prcopt_t *opt, sol_t *sol,
00112             double *azel, int *vsat, double *resp, char *msg);
00113
00114 /* doppler residuals -----*/
00115 int resdop(const obsd_t *obs, int n, const double *rs, const double *dts,
00116             const nav_t *nav, const double *rr, const double *x,
00117             const double *azel, const int *vsat, double *v, double *H);
00118
00119 /* estimate receiver velocity -----*/
00120 void estvel(const obsd_t *obs, int n, const double *rs, const double *dts,
00121             const nav_t *nav, const prcopt_t *opt, sol_t *sol,
00122             const double *azel, const int *vsat);
00123
00124 /*!
00125 * \brief single-point positioning
00126 * compute receiver position, velocity, clock bias by single-point positioning
00127 * with pseudorange and doppler observables
00128 * args : obsd_t *obs      I   observation data
00129 *         int  n          I   number of observation data
00130 *         nav_t *nav       I   navigation data
00131 *         prcopt_t *opt    I   processing options
00132 *         sol_t *sol       IO  solution
00133 *         double *azel     IO  azimuth/elevation angle (rad) (NULL: no output)
00134 *         ssat_t *ssat     IO  satellite status (NULL: no output)
00135 *         char  *msg       O   error message for error exit
00136 * return : status(1:ok,0:error)
00137 * notes : assuming sbas-gps, galileo-gps, qzss-gps, compass-gps time offset and
00138 *         receiver bias are negligible (only involving glonass-gps time offset
00139 *         and receiver bias)
00140 */
00141 int pntpos(const obsd_t *obs, int n, const nav_t *nav,
00142             const prcopt_t *opt, sol_t *sol, double *azel, ssat_t *ssat,
00143             char *msg);
00144
00145 #endif // GNSS_SDR_RTKLIB_PNTPOS_H

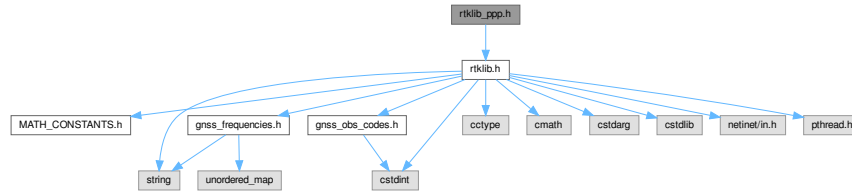
```

## 11.208 rtklib\_ppp.h File Reference

Precise Point Positioning.

```
#include "rtklib.h"
```

Include dependency graph for rtklib\_ppp.h:



### Macros

- #define [MIN\\_PPP](#)(x, y)
- #define [ROUND\\_PPP](#)(x)
- #define [SWAP\\_I](#)(x, y)
- #define [SWAP\\_D](#)(x, y)

### Functions

- double [lam\\_LC](#) (int i, int j, int k)
- double [L\\_LC](#) (int i, int j, int k, const double \*L)
- double [P\\_LC](#) (int i, int j, int k, const double \*P)
- double [var\\_LC](#) (int i, int j, int k, double sig)
- double [q\\_gamma](#) (double a, double x, double log\_gamma\_a)
- double [p\\_gamma](#) (double a, double x, double log\_gamma\_a)
- double [f\\_erfc](#) (double x)
- double [conffunc](#) (int N, double B, double sig)
- void [average\\_LC](#) (rtk\_t \*rtk, const obsd\_t \*obs, int n, const nav\_t \*nav, const double \*azel)
- int [fix\\_amb\\_WL](#) (rtk\_t \*rtk, const nav\_t \*nav, int sat1, int sat2, int \*NW)
- int [is\\_depend](#) (int sat1, int sat2, int \*flgs, int \*max\_flg)
- int [sel\\_amb](#) (int \*sat1, int \*sat2, double \*N, double \*var, int n)
- int [fix\\_sol](#) (rtk\_t \*rtk, const int \*sat1, const int \*sat2, const double \*NC, int n)
- int [fix\\_amb\\_ROUND](#) (rtk\_t \*rtk, int \*sat1, int \*sat2, const int \*NW, int n)
- int [fix\\_amb\\_ILS](#) (rtk\_t \*rtk, int \*sat1, int \*sat2, int \*NW, int n)
- int [pppamb](#) (rtk\_t \*rtk, const obsd\_t \*obs, int n, const nav\_t \*nav, const double \*azel)
- void [pppoutsolstat](#) (rtk\_t \*rtk, int level, FILE \*fp)
- void [testecclipse](#) (const obsd\_t \*obs, int n, const nav\_t \*nav, double \*rs)
- double [varerr](#) (int sat, int sys, double el, int type, const prcopt\_t \*opt)
- void [initx](#) (rtk\_t \*rtk, double xi, double var, int i)
- int [ifmeas](#) (const obsd\_t \*obs, const nav\_t \*nav, const double \*azel, const prcopt\_t \*opt, const double \*dantr, const double \*dants, double phw, double \*meas, double \*var)
- double [gettgd\\_ppp](#) (int sat, const nav\_t \*nav)
- int [corr\\_ion](#) (gtime\_t time, const nav\_t \*nav, int sat, const double \*pos, const double \*azel, int ionoopt, double \*ion, double \*var, int \*brk)
- int [corrmeas](#) (const obsd\_t \*obs, const nav\_t \*nav, const double \*pos, const double \*azel, const prcopt\_t \*opt, const double \*dantr, const double \*dants, double phw, double \*meas, double \*var, int \*brk)
- double [gfmeas](#) (const obsd\_t \*obs, const nav\_t \*nav)
- void [udpos\\_ppp](#) (rtk\_t \*rtk)
- void [udclk\\_ppp](#) (rtk\_t \*rtk)
- void [udtrop\\_ppp](#) (rtk\_t \*rtk)

- void **detslp\_ll** (rtk\_t \*rtk, const obsd\_t \*obs, int n)
- void **detslp\_gf** (rtk\_t \*rtk, const obsd\_t \*obs, int n, const nav\_t \*nav)
- void **udbias\_ppp** (rtk\_t \*rtk, const obsd\_t \*obs, int n, const nav\_t \*nav)
- void **udstate\_ppp** (rtk\_t \*rtk, const obsd\_t \*obs, int n, const nav\_t \*nav)
- void **satantpcv** (const double \*rs, const double \*rr, const pcv\_t \*pcv, double \*dant)
- double **prectrop** (gtime\_t time, const double \*pos, const double \*azel, const prcopt\_t \*opt, const double \*x, double \*dtdx, double \*var)
- int **res\_ppp** (int iter, const obsd\_t \*obs, int n, const double \*rs, const double \*dts, const double \*vare, const int \*svh, const nav\_t \*nav, const double \*x, rtk\_t \*rtk, double \*v, double \*H, double \*R, double \*azel)
- int **pppnx** (const prcopt\_t \*opt)
- void **pppos** (rtk\_t \*rtk, const obsd\_t \*obs, int n, const nav\_t \*nav)

## Variables

- const double **MIN\_ARC\_GAP** = 300.0
- const double **CONST\_AMB** = 0.001
- const double **THRES\_RES** = 0.3
- const double **LOG\_PI** = 1.14472988584940017
- const double **SQRT2** = 1.41421356237309510
- const double **VAR\_POS\_PPP** = std::pow(100.0, 2.0)
- const double **VAR\_CLK** = std::pow(100.0, 2.0)
- const double **VAR\_ZTD** = std::pow(0.3, 2.0)
- const double **VAR\_GRA\_PPP** = std::pow(0.001, 2.0)
- const double **VAR\_BIAS** = std::pow(100.0, 2.0)
- const double **VAR\_IONO\_OFF** = std::pow(10.0, 2.0)

## 11.208.1 Detailed Description

Precise Point Positioning.

### Authors

- 2007-2008, T. Takasu
- 2017, Javier Arribas
- 2017, Carles Fernandez

This is a derived work from RTKLIB <http://www.rtklib.com/> The original source code at <https://github.com/tomojitakasu/RTKLIB> is released under the BSD 2-clause license with an additional exclusive clause that does not apply here. This additional clause is reproduced below:

" The software package includes some companion executive binaries or shared libraries necessary to execute APs on Windows. These licenses succeed to the original ones of these software. "

Neither the executive binaries nor the shared libraries are required by, used or included in GNSS-SDR.

Copyright (C) 2007-2008, T. Takasu Copyright (C) 2017, Javier Arribas Copyright (C) 2017, Carles Fernandez All rights reserved.

SPDX-License-Identifier: BSD-2-Clause

Definition in file [rtklib\\_ppp.h](#).

## 11.208.2 Macro Definition Documentation

### 11.208.2.1 MIN\_PPP

```
#define MIN_PPP(  
    x,  
    y)
```

#### Value:

```
((x) <= (y) ? (x) : (y))
```

Definition at line 40 of file [rtklib\\_ppp.h](#).

### 11.208.2.2 ROUND\_PPP

```
#define ROUND_PPP(  
    x)
```

#### Value:

```
static_cast<int>(floor((x) + 0.5))
```

Definition at line 41 of file [rtklib\\_ppp.h](#).

### 11.208.2.3 SWAP\_D

```
#define SWAP_D(  
    x,  
    y)
```

#### Value:

```
do  
{  
    double _z = x;  
    x = y;  
    y = _z;  
}  
while (0)
```

Definition at line 51 of file [rtklib\\_ppp.h](#).

### 11.208.2.4 SWAP\_I

```
#define SWAP_I(  
    x,  
    y)
```

#### Value:

```
do  
{  
    int _z = x;  
    x = y;  
    y = _z;  
}  
while (0)
```

Definition at line 43 of file [rtklib\\_ppp.h](#).

## 11.208.3 Variable Documentation

### 11.208.3.1 CONST\_AMB

```
const double CONST_AMB = 0.001
```

Definition at line 61 of file [rtklib\\_ppp.h](#).

### 11.208.3.2 LOG\_PI

```
const double LOG_PI = 1.14472988584940017
```

Definition at line 63 of file [rtklib\\_ppp.h](#).

### 11.208.3.3 MIN\_ARC\_GAP

```
const double MIN_ARC_GAP = 300.0
```

Definition at line 60 of file [rtklib\\_ppp.h](#).

### 11.208.3.4 SQRT2

```
const double SQRT2 = 1.41421356237309510
```

Definition at line 64 of file [rtklib\\_ppp.h](#).

### 11.208.3.5 THRES\_RES

```
const double THRES_RES = 0.3
```

Definition at line 62 of file [rtklib\\_ppp.h](#).

**11.208.3.6 VAR\_BIAS**

```
const double VAR_BIAS = std::pow(100.0, 2.0)
```

Definition at line 70 of file [rtklib\\_ppp.h](#).

**11.208.3.7 VAR\_CLK**

```
const double VAR_CLK = std::pow(100.0, 2.0)
```

Definition at line 67 of file [rtklib\\_ppp.h](#).

**11.208.3.8 VAR\_GRA\_PPP**

```
const double VAR_GRA_PPP = std::pow(0.001, 2.0)
```

Definition at line 69 of file [rtklib\\_ppp.h](#).

**11.208.3.9 VAR\_IONO\_OFF**

```
const double VAR_IONO_OFF = std::pow(10.0, 2.0)
```

Definition at line 72 of file [rtklib\\_ppp.h](#).

**11.208.3.10 VAR\_POS\_PPP**

```
const double VAR_POS_PPP = std::pow(100.0, 2.0)
```

Definition at line 66 of file [rtklib\\_ppp.h](#).

**11.208.3.11 VAR\_ZTD**

```
const double VAR_ZTD = std::pow(0.3, 2.0)
```

Definition at line 68 of file [rtklib\\_ppp.h](#).

**11.209 rtklib\_ppp.h**

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file rtklib_ppp.h
00003  * \brief Precise Point Positioning
00004  * \authors <ul>
00005  *         <li> 2007-2008, T. Takasu
00006  *         <li> 2017, Javier Arribas
00007  *         <li> 2017, Carles Fernandez
00008  *         </ul>
00009  *
00010  * This is a derived work from RTKLIB http://www.rtklib.com/
00011  * The original source code at https://github.com/tomojitakasu/RTKLIB is
00012  * released under the BSD 2-clause license with an additional exclusive clause
00013  * that does not apply here. This additional clause is reproduced below:
00014  *
00015  * " The software package includes some companion executive binaries or shared
00016  * libraries necessary to execute APs on Windows. These licenses succeed to the
00017  * original ones of these software. "
00018  *
00019  * Neither the executive binaries nor the shared libraries are required by, used
00020  * or included in GNSS-SDR.
00021  *
00022  * -----
00023  * Copyright (C) 2007-2008, T. Takasu
00024  * Copyright (C) 2017, Javier Arribas
00025  * Copyright (C) 2017, Carles Fernandez
00026  * All rights reserved.
00027  *
00028  * SPDX-License-Identifier: BSD-2-Clause
00029  *
00030  * -----
00031  */
00032
00033
00034 #ifndef GNSS_SDR_RTKLIB_PPP_H
00035 #define GNSS_SDR_RTKLIB_PPP_H
00036
00037 #include "rtklib.h"
00038
```

```

00039
00040 #define MIN_PPP(x, y) ((x) <= (y) ? (x) : (y))
00041 #define ROUND_PPP(x) static_cast<int>(floor((x) + 0.5))
00042
00043 #define SWAP_I(x, y) \
00044     do \
00045     { \
00046         int _z = x; \
00047         x = y; \
00048         y = _z; \
00049     } \
00050     while (0)
00051 #define SWAP_D(x, y) \
00052     do \
00053     { \
00054         double _z = x; \
00055         x = y; \
00056         y = _z; \
00057     } \
00058     while (0)
00059
00060 const double MIN_ARC_GAP = 300.0; /* min arc gap (s) */
00061 const double CONST_AMB = 0.001; /* constraint to fixed ambiguity */
00062 const double THRES_RES = 0.3; /* threshold of residuals test (m) */
00063 const double LOG_PI = 1.14472988584940017; /* log(pi) */
00064 const double SQRT2 = 1.41421356237309510; /* sqrt(2) */
00065
00066 const double VAR_POS_PPP = std::pow(100.0, 2.0); /* init variance receiver position (m^2) */
00067 const double VAR_CLK = std::pow(100.0, 2.0); /* init variance receiver clock (m^2) */
00068 const double VAR_ZTD = std::pow(0.3, 2.0); /* init variance ztd (m^2) */
00069 const double VAR_GRA_PPP = std::pow(0.001, 2.0); /* init variance gradient (m^2) */
00070 const double VAR_BIAS = std::pow(100.0, 2.0); /* init variance phase-bias (m^2) */
00071
00072 const double VAR_IONO_OFF = std::pow(10.0, 2.0); /* variance of iono-model-off */
00073
00074
00075 /* functions originally included in RTKLIB/src/ppp_ar.c v2.4.2 */
00076 double lam_LC(int i, int j, int k);
00077
00078 double L_LC(int i, int j, int k, const double *L);
00079
00080 double P_LC(int i, int j, int k, const double *P);
00081
00082 double var_LC(int i, int j, int k, double sig);
00083
00084 double q_gamma(double a, double x, double log_gamma_a);
00085
00086 double p_gamma(double a, double x, double log_gamma_a);
00087
00088 double f_erfc(double x);
00089
00090 double conffunc(int N, double B, double sig);
00091
00092 void average_LC(rtk_t *rtk, const obsd_t *obs, int n, const nav_t *nav, const double *azel);
00093
00094 int fix_amb_WL(rtk_t *rtk, const nav_t *nav, int sat1, int sat2, int *NW);
00095
00096 int is_depend(int sat1, int sat2, int *flgs, int *max_flg);
00097
00098 int sel_amb(int *sat1, int *sat2, double *N, double *var, int n);
00099
00100 int fix_sol(rtk_t *rtk, const int *sat1, const int *sat2, const double *NC, int n);
00101
00102 int fix_amb_ROUND(rtk_t *rtk, int *sat1, int *sat2, const int *NW, int n);
00103
00104 int fix_amb_ILS(rtk_t *rtk, int *sat1, int *sat2, int *NW, int n);
00105
00106 int pppamb(rtk_t *rtk, const obsd_t *obs, int n, const nav_t *nav, const double *azel);
00107
00108
00109 /* functions originally included in RTKLIB/src/ppp.c v2.4.2 */
00110 void pppoutsolstat(rtk_t *rtk, int level, FILE *fp);
00111
00112 void testeclipse(const obsd_t *obs, int n, const nav_t *nav, double *rs);
00113
00114 double varerr(int sat, int sys, double el, int type, const prcopt_t *opt);
00115
00116 void initx(rtk_t *rtk, double xi, double var, int i);
00117
00118 int ifmeas(const obsd_t *obs, const nav_t *nav, const double *azel,
00119           const prcopt_t *opt, const double *dantr, const double *dants,
00120           double phw, double *meas, double *var);
00121
00122 double gettgdd_ppp(int sat, const nav_t *nav);
00123
00124 int corr_ion(gtime_t time, const nav_t *nav, int sat, const double *pos,
00125             const double *azel, int ionoopt, double *ion, double *var,

```



```

00126     int *brk);
00127
00128 int corrmeas(const obsd_t *obs, const nav_t *nav, const double *pos,
00129     const double *azel, const prcopt_t *opt,
00130     const double *dantr, const double *dants, double phw,
00131     double *meas, double *var, int *brk);
00132
00133 double gfmeas(const obsd_t *obs, const nav_t *nav);
00134
00135 void udpos_ppp(rtk_t *rtk);
00136
00137 void udcik_ppp(rtk_t *rtk);
00138
00139 void udtrop_ppp(rtk_t *rtk);
00140
00141 void detslp_ll(rtk_t *rtk, const obsd_t *obs, int n);
00142
00143 void detslp_gf(rtk_t *rtk, const obsd_t *obs, int n, const nav_t *nav);
00144
00145 void udbias_ppp(rtk_t *rtk, const obsd_t *obs, int n, const nav_t *nav);
00146
00147 void udsta_ppp(rtk_t *rtk, const obsd_t *obs, int n, const nav_t *nav);
00148
00149 void satantpcv(const double *rs, const double *rr, const pcv_t *pcv, double *dant);
00150
00151 double prectrop(gtime_t time, const double *pos, const double *azel,
00152     const prcopt_t *opt, const double *x, double *dtdx,
00153     double *var);
00154
00155 int res_ppp(int iter, const obsd_t *obs, int n, const double *rs,
00156     const double *dts, const double *vare, const int *svh,
00157     const nav_t *nav, const double *x, rtk_t *rtk, double *v,
00158     double *H, double *R, double *azel);
00159
00160 int pppnx(const prcopt_t *opt);
00161
00162 void pppos(rtk_t *rtk, const obsd_t *obs, int n, const nav_t *nav);
00163
00164 #endif

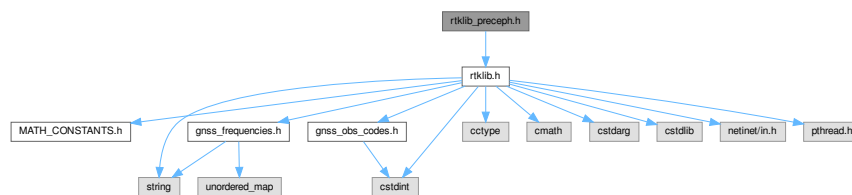
```

## 11.210 rtklib\_preceph.h File Reference

precise ephemeris and clock functions

```
#include "rtklib.h"
```

Include dependency graph for rtklib\_preceph.h:



### Functions

- int **code2sys** (char code)
- int **readsp3h** (FILE \*fp, gtime\_t \*time, char \*type, int \*sats, double \*bfact, char \*tsys)
- int **addpeph** (nav\_t \*nav, peph\_t \*peph)
- void **readsp3b** (FILE \*fp, char type, int \*sats, int ns, const double \*bfact, char \*tsys, int index, int opt, nav\_t \*nav)
- int **cmppeph** (const void \*p1, const void \*p2)
- void **combpeph** (nav\_t \*nav, int opt)
- void **readsp3** (const char \*file, nav\_t \*nav, int opt)
- int **readsap** (const char \*file, gtime\_t time, nav\_t \*nav)
- int **readdcbf** (const char \*file, nav\_t \*nav, const sta\_t \*sta)
- int **readdcb** (const char \*file, nav\_t \*nav, const sta\_t \*sta)

- double **interppl** (const double \*x, double \*y, int n)
- int **pephpos** ([gtime\\_t](#) time, int sat, const [nav\\_t](#) \*nav, double \*rs, double \*dts, double \*vare, double \*varc)
- int **pephclk** ([gtime\\_t](#) time, int sat, const [nav\\_t](#) \*nav, double \*dts, double \*varc)
- void **satantoff** ([gtime\\_t](#) time, const double \*rs, int sat, const [nav\\_t](#) \*nav, double \*dant)
- int **peph2pos** ([gtime\\_t](#) time, int sat, const [nav\\_t](#) \*nav, int opt, double \*rs, double \*dts, double \*var)

## Variables

- const int **NMAX** = 10
- const double **MAXDTE** = 900.0
- const double **EXTERR\_CLK** = 1e-3
- const double **EXTERR\_EPH** = 5e-7

## 11.210.1 Detailed Description

precise ephemeris and clock functions

### Authors

- 2007-2013, T. Takasu
- 2017, Javier Arribas
- 2017, Carles Fernandez

This is a derived work from RTKLIB <http://www.rtklib.com/> The original source code at <https://github.com/tomojitakasu/RTKLIB> is released under the BSD 2-clause license with an additional exclusive clause that does not apply here. This additional clause is reproduced below:

" The software package includes some companion executive binaries or shared libraries necessary to execute APs on Windows. These licenses succeed to the original ones of these software. "

Neither the executive binaries nor the shared libraries are required by, used or included in GNSS-SDR.

Copyright (C) 2007-2013, T. Takasu Copyright (C) 2017, Javier Arribas Copyright (C) 2017, Carles Fernandez All rights reserved.

SPDX-License-Identifier: BSD-2-Clause

References : [1] S.Hilla, The Extended Standard Product 3 Orbit Format (SP3-c), 12 February, 2007 [2] J.Ray, W. Gurtner, RINEX Extensions to Handle Clock Information, 27 August, 1998 [3] D.D.McCarthy, IERS Technical Note 21, IERS Conventions 1996, July 1996 [4] D.A.Vallado, Fundamentals of Astrodynamics and Applications 2nd ed, Space Technology Library, 2004

Definition in file [rtklib\\_preceph.h](#).

## 11.210.2 Variable Documentation

### 11.210.2.1 EXTERR\_CLK

```
const double EXTERR_CLK = 1e-3
```

Definition at line 51 of file [rtklib\\_preceph.h](#).

### 11.210.2.2 EXTERR\_EPH

```
const double EXTERR_EPH = 5e-7
```

Definition at line 52 of file [rtklib\\_preceph.h](#).

### 11.210.2.3 MAXDTE

```
const double MAXDTE = 900.0
```

Definition at line 50 of file [rtklib\\_preceph.h](#).

### 11.210.2.4 NMAX

```
const int NMAX = 10
```

Definition at line 49 of file [rtklib\\_preceph.h](#).

## 11.211 rtklib\_preceph.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file rtklib_preceph.h
00003  * \brief precise ephemeris and clock functions
00004  * \authors <ul>
00005  *         <li> 2007-2013, T. Takasu
00006  *         <li> 2017, Javier Arribas
00007  *         <li> 2017, Carles Fernandez
00008  *     </ul>
00009  *
00010  * This is a derived work from RTKLIB http://www.rtklib.com/
00011  * The original source code at https://github.com/tomojitakasu/RTKLIB is
00012  * released under the BSD 2-clause license with an additional exclusive clause
00013  * that does not apply here. This additional clause is reproduced below:
00014  *
00015  * " The software package includes some companion executive binaries or shared
00016  * libraries necessary to execute APs on Windows. These licenses succeed to the
00017  * original ones of these software. "
00018  *
00019  * Neither the executive binaries nor the shared libraries are required by, used
00020  * or included in GNSS-SDR.
00021  *
00022  * -----
00023  * Copyright (C) 2007-2013, T. Takasu
00024  * Copyright (C) 2017, Javier Arribas
00025  * Copyright (C) 2017, Carles Fernandez
00026  * All rights reserved.
00027  *
00028  * SPDX-License-Identifier: BSD-2-Clause
00029  *
00030  *
00031  * References :
00032  * [1] S.Hilla, The Extended Standard Product 3 Orbit Format (SP3-c),
00033  *     12 February, 2007
00034  * [2] J.Ray, W.Gurtner, RINEX Extensions to Handle Clock Information,
00035  *     27 August, 1998
00036  * [3] D.D.McCarthy, IERS Technical Note 21, IERS Conventions 1996, July 1996
00037  * [4] D.A.Vallado, Fundamentals of Astrodynamics and Applications 2nd ed,
00038  *     Space Technology Library, 2004
00039  *
00040  * -----
00041  */
00042
00043 #ifndef GNSS_SDR_RTKLIB_PRECEPH_H
00044 #define GNSS_SDR_RTKLIB_PRECEPH_H
00045
00046 #include "rtklib.h"
00047
00048
00049 const int NMAX = 10;          /* order of polynomial interpolation */
00050 const double MAXDTE = 900.0; /* max time difference to ephemeris time (s) */
00051 const double EXTERR_CLK = 1e-3; /* extrapolation error for clock (m/s) */
00052 const double EXTERR_EPH = 5e-7; /* extrapolation error for ephemeris (m/s^2) */
00053
00054 int code2sys(char code);
00055
00056 int readsp3h(FILE *fp, gtime_t *time, char *type, int *sats,
00057             double *bfact, char *tsys);
00058
00059 int addpeph(nav_t *nav, peph_t *peph);
00060
00061 void readsp3b(FILE *fp, char type, int *sats, int ns, const double *bfact,
00062             char *tsys, int index, int opt, nav_t *nav);
00063
00064 int cmppeph(const void *p1, const void *p2);
00065
00066 void combpeph(nav_t *nav, int opt);
00067
00068 void readsp3(const char *file, nav_t *nav, int opt);
00069
00070 int readsap(const char *file, gtime_t time, nav_t *nav);
00071
00072 int readdcbf(const char *file, nav_t *nav, const sta_t *sta);
00073
00074 int readdcb(const char *file, nav_t *nav, const sta_t *sta);
00075
00076 double interppl(const double *x, double *y, int n);
00077
00078 int pephpos(gtime_t time, int sat, const nav_t *nav, double *rs,
00079            double *dts, double *vare, double *varc);
00080
00081 int pephclk(gtime_t time, int sat, const nav_t *nav, double *dts,
00082            double *varc);
00083

```

```

00084 void satantoff(gtime_t time, const double *rs, int sat, const nav_t *nav,
00085                 double *dant);
00086
00087 int pep2pos(gtime_t time, int sat, const nav_t *nav, int opt,
00088            double *rs, double *dts, double *var);
00089
00090 #endif // GNSS_SDR_RTKLIB_PRECEPH_H

```

## 11.212 rtklib\_rtc.h File Reference

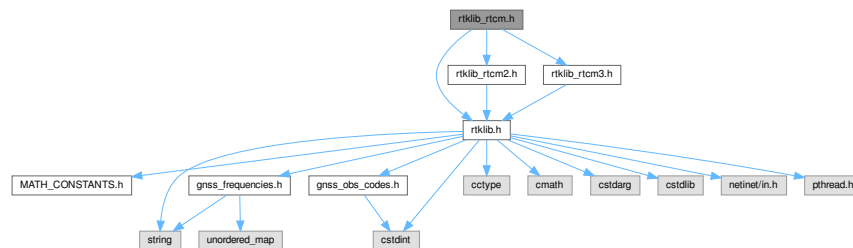
RTCM functions headers.

```
#include "rtklib.h"
```

```
#include "rtklib_rtc2.h"
```

```
#include "rtklib_rtc3.h"
```

Include dependency graph for rtklib\_rtc.h:



### Macros

- #define [RTCM2PREAMB](#) 0x66 /\* rtc ver.2 frame preamble \*/
- #define [RTCM3PREAMB](#) 0xD3 /\* rtc ver.3 frame preamble \*/

### Functions

- int [init\\_rtc](#) (rtc\_t \*rtc)
- void [free\\_rtc](#) (rtc\_t \*rtc)
- int [input\\_rtc2](#) (rtc\_t \*rtc, unsigned char data)
- int [input\\_rtc3](#) (rtc\_t \*rtc, unsigned char data)
- int [input\\_rtc2f](#) (rtc\_t \*rtc, FILE \*fp)
- int [input\\_rtc3f](#) (rtc\_t \*rtc, FILE \*fp)
- int [gen\\_rtc2](#) (rtc\_t \*rtc, int type, int sync)

### 11.212.1 Detailed Description

RTCM functions headers.

#### Authors

- 2007-2013, T. Takasu
- 2017, Javier Arribas
- 2017, Carles Fernandez

This is a derived work from RTKLIB <http://www.rtklib.com/> The original source code at <https://github.com/tomojitakasu/RTKLIB> is released under the BSD 2-clause license with an additional exclusive clause that does not apply here. This additional clause is reproduced below:

" The software package includes some companion executive binaries or shared libraries necessary to execute APs on Windows. These licenses succeed to the original ones of these software. "

Neither the executive binaries nor the shared libraries are required by, used or included in GNSS-SDR.

Copyright (C) 2007-2013, T. Takasu Copyright (C) 2017, Javier Arribas Copyright (C) 2017, Carles Fernandez All rights reserved.

SPDX-License-Identifier: BSD-2-Clause  
Definition in file [rtklib\\_rtcn.h](#).

## 11.212.2 Macro Definition Documentation

### 11.212.2.1 RTCM2PREAMB

```
#define RTCM2PREAMB 0x66 /* rtcn ver.2 frame preamble */
```

Definition at line 42 of file [rtklib\\_rtcn.h](#).

### 11.212.2.2 RTCM3PREAMB

```
#define RTCM3PREAMB 0xD3 /* rtcn ver.3 frame preamble */
```

Definition at line 43 of file [rtklib\\_rtcn.h](#).

## 11.213 rtklib\_rtcn.h

[Go to the documentation of this file.](#)

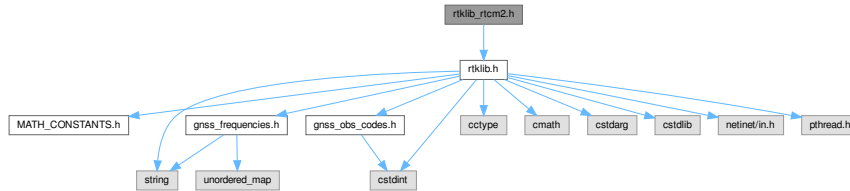
```
00001 /*!
00002  * \file rtklib_rtcn.h
00003  * \brief RTCM functions headers
00004  * \authors <ul>
00005  *         <li> 2007-2013, T. Takasu
00006  *         <li> 2017, Javier Arribas
00007  *         <li> 2017, Carles Fernandez
00008  *         </ul>
00009  *
00010  * This is a derived work from RTKLIB http://www.rtklib.com/
00011  * The original source code at https://github.com/tomojitakasu/RTKLIB is
00012  * released under the BSD 2-clause license with an additional exclusive clause
00013  * that does not apply here. This additional clause is reproduced below:
00014  *
00015  * " The software package includes some companion executive binaries or shared
00016  * libraries necessary to execute APs on Windows. These licenses succeed to the
00017  * original ones of these software. "
00018  *
00019  * Neither the executive binaries nor the shared libraries are required by, used
00020  * or included in GNSS-SDR.
00021  *
00022  * -----
00023  * Copyright (C) 2007-2013, T. Takasu
00024  * Copyright (C) 2017, Javier Arribas
00025  * Copyright (C) 2017, Carles Fernandez
00026  * All rights reserved.
00027  *
00028  * SPDX-License-Identifier: BSD-2-Clause
00029  *
00030  * -----
00031  */
00032 */
00033
00034
00035 #ifndef GNSS_SDR_RTKLIB_RTCN_H
00036 #define GNSS_SDR_RTKLIB_RTCN_H
00037
00038 #include "rtklib.h"
00039 #include "rtklib_rtcn2.h"
00040 #include "rtklib_rtcn3.h"
00041
00042 #define RTCM2PREAMB 0x66 /* rtcn ver.2 frame preamble */
00043 #define RTCM3PREAMB 0xD3 /* rtcn ver.3 frame preamble */
00044
00045
00046 int init_rtcn(rtcn_t *rtc);
00047 void free_rtcn(rtcn_t *rtc);
00048 int input_rtcn2(rtcn_t *rtc, unsigned char data);
00049 int input_rtcn3(rtcn_t *rtc, unsigned char data);
00050 int input_rtcn2f(rtcn_t *rtc, FILE *fp);
00051 int input_rtcn3f(rtcn_t *rtc, FILE *fp);
00052 int gen_rtcn2(rtcn_t *rtc, int type, int sync);
00053 // int gen_rtcn3(rtcn_t *rtc, int type, int sync);
00054
00055
00056 #endif // GNSS_SDR_RTKLIB_RTCN_H
```

## 11.214 rtklib\_rtc2.h File Reference

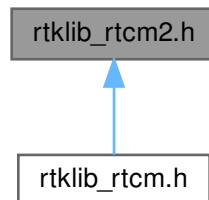
RTCM v2 functions headers.

```
#include "rtklib.h"
```

Include dependency graph for rtklib\_rtc2.h:



This graph shows which files directly or indirectly include this file:



### Functions

- void **adjhour** ([rtcm\\_t](#) \*rtcm, double zcnt)
- int **obsindex** ([obs\\_t](#) \*obs, [gtime\\_t](#) time, int sat)
- int **decode\_type1** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type3** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type14** ([rtcm\\_t](#) \*rtcm, bool pre\_2009\_file=false)
- int **decode\_type16** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type17** ([rtcm\\_t](#) \*rtcm, bool pre\_2009\_file=false)
- int **decode\_type18** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type19** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type22** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type23** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type24** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type31** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type32** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type34** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type36** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type37** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type59** ([rtcm\\_t](#) \*rtcm)
- int **decode\_rtc2** ([rtcm\\_t](#) \*rtcm)

### 11.214.1 Detailed Description

RTCM v2 functions headers.

#### Authors

- 2007-2013, T. Takasu
- 2017, Javier Arribas
- 2017, Carles Fernandez

This is a derived work from RTKLIB <http://www.rtklib.com/> The original source code at <https://github.com/tomojitakasu/RTKLIB> is released under the BSD 2-clause license with an additional exclusive clause that does not apply here. This additional clause is reproduced below:

" The software package includes some companion executive binaries or shared libraries necessary to execute APs on Windows. These licenses succeed to the original ones of these software. "

Neither the executive binaries nor the shared libraries are required by, used or included in GNSS-SDR.

Copyright (C) 2007-2013, T. Takasu Copyright (C) 2017, Javier Arribas Copyright (C) 2017, Carles Fernandez All rights reserved.

SPDX-License-Identifier: BSD-2-Clause

Definition in file [rtklib\\_rtc2.h](#).

## 11.215 rtklib\_rtc2.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file rtklib_rtc2.h
00003  * \brief RTCM v2 functions headers
00004  * \authors <ul>
00005  *         <li> 2007-2013, T. Takasu
00006  *         <li> 2017, Javier Arribas
00007  *         <li> 2017, Carles Fernandez
00008  *         </ul>
00009  *
00010  * This is a derived work from RTKLIB http://www.rtklib.com/
00011  * The original source code at https://github.com/tomojitakasu/RTKLIB is
00012  * released under the BSD 2-clause license with an additional exclusive clause
00013  * that does not apply here. This additional clause is reproduced below:
00014  *
00015  * " The software package includes some companion executive binaries or shared
00016  * libraries necessary to execute APs on Windows. These licenses succeed to the
00017  * original ones of these software. "
00018  *
00019  * Neither the executive binaries nor the shared libraries are required by, used
00020  * or included in GNSS-SDR.
00021  *
00022  * -----
00023  * Copyright (C) 2007-2013, T. Takasu
00024  * Copyright (C) 2017, Javier Arribas
00025  * Copyright (C) 2017, Carles Fernandez
00026  * All rights reserved.
00027  *
00028  * SPDX-License-Identifier: BSD-2-Clause
00029  *
00030  * -----
00031  */
00032 */
00033
00034 #ifndef GNSS_SDR_RTKLIB_RTC2_H
00035 #define GNSS_SDR_RTKLIB_RTC2_H
00036
00037 #include "rtklib.h"
00038
00039
00040 void adjhour(rtc2_t *rtc2, double zcnt);
00041 int obsindex(obs_t *obs, gtime_t time, int sat);
00042 int decode_type1(rtc2_t *rtc2);
00043 int decode_type3(rtc2_t *rtc2);
00044 int decode_type14(rtc2_t *rtc2, bool pre_2009_file = false);
00045 int decode_type16(rtc2_t *rtc2);
00046 int decode_type17(rtc2_t *rtc2, bool pre_2009_file = false);
00047 int decode_type18(rtc2_t *rtc2);
00048 int decode_type19(rtc2_t *rtc2);
00049 int decode_type22(rtc2_t *rtc2);
00050 int decode_type23(rtc2_t *rtc2);
00051 int decode_type24(rtc2_t *rtc2);
00052 int decode_type31(rtc2_t *rtc2);
00053 int decode_type32(rtc2_t *rtc2);
```

```

00054 int decode_type34(rtcm_t *rtcm);
00055 int decode_type36(rtcm_t *rtcm);
00056 int decode_type37(rtcm_t *rtcm);
00057 int decode_type59(rtcm_t *rtcm);
00058 int decode_rtcm2(rtcm_t *rtcm);
00059
00060 #endif

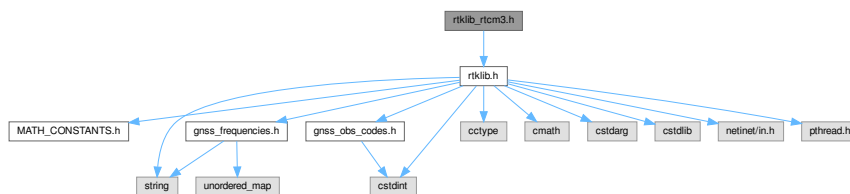
```

## 11.216 rtklib\_rtcm3.h File Reference

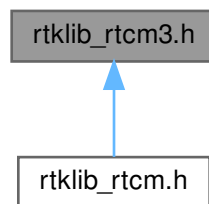
RTCM v3 functions headers.

```
#include "rtklib.h"
```

Include dependency graph for rtklib\_rtcm3.h:



This graph shows which files directly or indirectly include this file:



### Functions

- double **getbitg** (const unsigned char \*buff, int pos, int len)
- void **adjweek** (*rtcm\_t* \*rtcm, double tow)
- int **adjbdtweek** (int week)
- void **adjday\_glot** (*rtcm\_t* \*rtcm, double tod)
- double **adjcp** (*rtcm\_t* \*rtcm, int sat, int freq, double cp)
- int **lossoflock** (*rtcm\_t* \*rtcm, int sat, int freq, int lock)
- unsigned char **snratio** (double snr)
- int **obsindex3** (*obs\_t* \*obs, *gtime\_t* time, int sat)
- int **test\_staid** (*rtcm\_t* \*rtcm, int staid)
- int **decode\_head1001** (*rtcm\_t* \*rtcm, int \*sync)
- int **decode\_type1001** (*rtcm\_t* \*rtcm)
- int **decode\_type1002** (*rtcm\_t* \*rtcm)
- int **decode\_type1003** (*rtcm\_t* \*rtcm)
- int **decode\_type1004** (*rtcm\_t* \*rtcm)
- double **getbits\_38** (const unsigned char \*buff, int pos)



- int **decode\_type1005** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1006** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1007** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1008** ([rtcm\\_t](#) \*rtcm)
- int **decode\_head1009** ([rtcm\\_t](#) \*rtcm, int \*sync)
- int **decode\_type1009** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1010** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1011** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1012** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1013** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1019** ([rtcm\\_t](#) \*rtcm, bool pre\_2009\_file=false)
- int **decode\_type1020** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1021** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1022** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1023** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1024** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1025** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1026** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1027** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1029** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1030** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1031** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1032** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1033** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1034** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1035** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1037** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1038** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1039** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1044** ([rtcm\\_t](#) \*rtcm, bool pre\_2009\_file=false)
- int **decode\_type1045** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1046** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1047** ([rtcm\\_t](#) \*rtcm)
- int **decode\_type1063** ([rtcm\\_t](#) \*rtcm)
- int **decode\_ssr1\_head** ([rtcm\\_t](#) \*rtcm, int sys, int \*sync, int \*iod, double \*udint, int \*refd, int \*hsize)
- int **decode\_ssr2\_head** ([rtcm\\_t](#) \*rtcm, int sys, int \*sync, int \*iod, double \*udint, int \*hsize)
- int **decode\_ssr7\_head** ([rtcm\\_t](#) \*rtcm, int sys, int \*sync, int \*iod, double \*udint, int \*dispe, int \*mw, int \*hsize)
- int **decode\_ssr1** ([rtcm\\_t](#) \*rtcm, int sys)
- int **decode\_ssr2** ([rtcm\\_t](#) \*rtcm, int sys)
- int **decode\_ssr3** ([rtcm\\_t](#) \*rtcm, int sys)
- int **decode\_ssr4** ([rtcm\\_t](#) \*rtcm, int sys)
- int **decode\_ssr5** ([rtcm\\_t](#) \*rtcm, int sys)
- int **decode\_ssr6** ([rtcm\\_t](#) \*rtcm, int sys)
- int **decode\_ssr7** ([rtcm\\_t](#) \*rtcm, int sys)
- void **sigindex** (int sys, const unsigned char \*code, const int \*freq, int n, const char \*opt, int \*ind)
- void **save\_msm\_obs** ([rtcm\\_t](#) \*rtcm, int sys, [msm\\_h\\_t](#) \*h, const double \*r, const double \*pr, const double \*cp, const double \*rr, const double \*rrf, const double \*cnr, const int \*lock, const int \*ex, const int \*half)
- int **decode\_msm\_head** ([rtcm\\_t](#) \*rtcm, int sys, int \*sync, int \*iod, [msm\\_h\\_t](#) \*h, int \*hsize)
- int **decode\_msm0** ([rtcm\\_t](#) \*rtcm, int sys)
- int **decode\_msm4** ([rtcm\\_t](#) \*rtcm, int sys)
- int **decode\_msm5** ([rtcm\\_t](#) \*rtcm, int sys)
- int **decode\_msm6** ([rtcm\\_t](#) \*rtcm, int sys)
- int **decode\_msm7** ([rtcm\\_t](#) \*rtcm, int sys)
- int **decode\_type1230** ([rtcm\\_t](#) \*rtcm)
- int **decode\_rtc3** ([rtcm\\_t](#) \*rtcm)

## Variables

- const double `PRUNIT_GPS` = 299792.458
- const double `PRUNIT_GLO` = 599584.916
- const double `RANGE_MS` = `SPEED_OF_LIGHT_M_S` \* 0.001
- const double `SSRUDINT` [16]
- const int `CODES_GPS` []
- const int `CODES_GLO` []
- const int `CODES_GAL` []
- const int `CODES_QZS` []
- const int `CODES_BDS` []
- const int `CODES_SBS` []

### 11.216.1 Detailed Description

RTCM v3 functions headers.

#### Authors

- 2007-2013, T. Takasu
- 2017, Javier Arribas
- 2017, Carles Fernandez

This is a derived work from RTKLIB <http://www.rtklib.com/> The original source code at <https://github.com/tomojitakasu/RTKLIB> is released under the BSD 2-clause license with an additional exclusive clause that does not apply here. This additional clause is reproduced below:

" The software package includes some companion executive binaries or shared libraries necessary to execute APs on Windows. These licenses succeed to the original ones of these software. "

Neither the executive binaries nor the shared libraries are required by, used or included in GNSS-SDR.

Copyright (C) 2007-2013, T. Takasu Copyright (C) 2017, Javier Arribas Copyright (C) 2017, Carles Fernandez All rights reserved.

SPDX-License-Identifier: BSD-2-Clause

Definition in file [rtklib\\_rtc3.h](#).

### 11.216.2 Variable Documentation

#### 11.216.2.1 CODES\_BDS

```
const int CODES_BDS[]
```

##### Initial value:

```
= {
    CODE_L1I, CODE_L1Q, CODE_L1X, CODE_L7I, CODE_L7Q, CODE_L7X, CODE_L6I, CODE_L6Q,
    CODE_L6X}
```

Definition at line 73 of file [rtklib\\_rtc3.h](#).

#### 11.216.2.2 CODES\_GAL

```
const int CODES_GAL[]
```

##### Initial value:

```
= {
    CODE_L1A, CODE_L1B, CODE_L1C, CODE_L1X, CODE_L1Z, CODE_L5I, CODE_L5Q, CODE_L5X,
    CODE_L7I, CODE_L7Q, CODE_L7X, CODE_L8I, CODE_L8Q, CODE_L8X, CODE_L6A, CODE_L6B,
    CODE_L6C, CODE_L6X, CODE_L6Z}
```

Definition at line 62 of file [rtklib\\_rtc3.h](#).

#### 11.216.2.3 CODES\_GLO

```
const int CODES_GLO[]
```

##### Initial value:

```
= {
    CODE_L1C, CODE_L1P, CODE_L2C, CODE_L2P}
```

Definition at line 58 of file [rtklib\\_rtc3.h](#).

**11.216.2.4 CODES\_GPS**

```
const int CODES_GPS[]
```

**Initial value:**

```
= {
    CODE_L1C, CODE_L1P, CODE_L1W, CODE_L1Y, CODE_L1M, CODE_L2C, CODE_L2D, CODE_L2S,
    CODE_L2L, CODE_L2X, CODE_L2P, CODE_L2W, CODE_L2Y, CODE_L2M, CODE_L5I, CODE_L5Q,
    CODE_L5X}
```

Definition at line 52 of file [rtklib\\_rtc3.h](#).**11.216.2.5 CODES\_QZS**

```
const int CODES_QZS[]
```

**Initial value:**

```
= {
    CODE_L1C, CODE_L1S, CODE_L1L, CODE_L2S, CODE_L2L, CODE_L2X, CODE_L5I, CODE_L5Q,
    CODE_L5X, CODE_L6S, CODE_L6L, CODE_L6X, CODE_L1X}
```

Definition at line 68 of file [rtklib\\_rtc3.h](#).**11.216.2.6 CODES\_SBS**

```
const int CODES_SBS[]
```

**Initial value:**

```
= {
    CODE_L1C, CODE_L5I, CODE_L5Q, CODE_L5X}
```

Definition at line 78 of file [rtklib\\_rtc3.h](#).**11.216.2.7 PRUNIT\_GLO**

```
const double PRUNIT_GLO = 599584.916
```

Definition at line 42 of file [rtklib\\_rtc3.h](#).**11.216.2.8 PRUNIT\_GPS**

```
const double PRUNIT_GPS = 299792.458
```

Definition at line 41 of file [rtklib\\_rtc3.h](#).**11.216.2.9 RANGE\_MS**

```
const double RANGE_MS = SPEED_OF_LIGHT_M_S * 0.001
```

Definition at line 43 of file [rtklib\\_rtc3.h](#).**11.216.2.10 SSRUDINT**

```
const double SSRUDINT[16]
```

**Initial value:**

```
= {
    1, 2, 5, 10, 15, 30, 60, 120, 240, 300, 600, 900, 1800, 3600, 7200, 10800}
```

Definition at line 47 of file [rtklib\\_rtc3.h](#).**11.217 rtklib\_rtc3.h**[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file rtklib_rtc3.h
00003  * \brief RTCM v3 functions headers
00004  * \authors <ul>
00005  *         <li> 2007-2013, T. Takasu
00006  *         <li> 2017, Javier Arribas
00007  *         <li> 2017, Carles Fernandez
00008  *         </ul>
00009  *
00010  * This is a derived work from RTKLIB http://www.rtklib.com/
00011  * The original source code at https://github.com/tomojitakasu/RTKLIB is
00012  * released under the BSD 2-clause license with an additional exclusive clause
```

```

00013  * that does not apply here. This additional clause is reproduced below:
00014  *
00015  * " The software package includes some companion executive binaries or shared
00016  * libraries necessary to execute APs on Windows. These licenses succeed to the
00017  * original ones of these software. "
00018  *
00019  * Neither the executive binaries nor the shared libraries are required by, used
00020  * or included in GNSS-SDR.
00021  *
00022  * -----
00023  * Copyright (C) 2007-2013, T. Takasu
00024  * Copyright (C) 2017, Javier Arribas
00025  * Copyright (C) 2017, Carles Fernandez
00026  * All rights reserved.
00027  *
00028  * SPDX-License-Identifier: BSD-2-Clause
00029  *
00030  *
00031  * -----
00032  */
00033
00034 #ifndef GNSS_SDR_RTKLIB_RTCM3_H
00035 #define GNSS_SDR_RTKLIB_RTCM3_H
00036
00037 #include "rtklib.h"
00038
00039 /* constants -----*/
00040
00041 const double PRUNIT_GPS = 299792.458; /* rtcv ver.3 unit of gps pseudorange (m) */
00042 const double PRUNIT_GLO = 599584.916; /* rtcv ver.3 unit of glonass pseudorange (m) */
00043 const double RANGE_MS = SPEED_OF_LIGHT_M_S * 0.001; /* range in 1 ms */
00044
00045
00046 /* ssr update intervals -----*/
00047 const double SSRUDINT[16] = {
00048     1, 2, 5, 10, 15, 30, 60, 120, 240, 300, 600, 900, 1800, 3600, 7200, 10800;
00049
00050
00051 /* ssr 3 and 7 signal and tracking mode ids -----*/
00052 const int CODES_GPS[] = {
00053     CODE_L1C, CODE_L1P, CODE_L1W, CODE_L1Y, CODE_L1M, CODE_L2C, CODE_L2D, CODE_L2S,
00054     CODE_L2L, CODE_L2X, CODE_L2P, CODE_L2W, CODE_L2Y, CODE_L2M, CODE_L5I, CODE_L5Q,
00055     CODE_L5X};
00056
00057
00058 const int CODES_GLO[] = {
00059     CODE_L1C, CODE_L1P, CODE_L2C, CODE_L2P};
00060
00061
00062 const int CODES_GAL[] = {
00063     CODE_L1A, CODE_L1B, CODE_L1C, CODE_L1X, CODE_L1Z, CODE_L5I, CODE_L5Q, CODE_L5X,
00064     CODE_L7I, CODE_L7Q, CODE_L7X, CODE_L8I, CODE_L8Q, CODE_L8X, CODE_L6A, CODE_L6B,
00065     CODE_L6C, CODE_L6X, CODE_L6Z};
00066
00067
00068 const int CODES_QZS[] = {
00069     CODE_L1C, CODE_L1S, CODE_L1L, CODE_L2S, CODE_L2L, CODE_L2X, CODE_L5I, CODE_L5Q,
00070     CODE_L5X, CODE_L6S, CODE_L6L, CODE_L6X, CODE_L1X};
00071
00072
00073 const int CODES_BDS[] = {
00074     CODE_L1I, CODE_L1Q, CODE_L1X, CODE_L7I, CODE_L7Q, CODE_L7X, CODE_L6I, CODE_L6Q,
00075     CODE_L6X};
00076
00077
00078 const int CODES_SBS[] = {
00079     CODE_L1C, CODE_L5I, CODE_L5Q, CODE_L5X};
00080
00081
00082 double getbitg(const unsigned char *buff, int pos, int len);
00083
00084 void adjweek(rtcv_t *rtcm, double tow);
00085
00086 int adjbdtweek(int week);
00087
00088 void adjday_glot(rtcv_t *rtcm, double tod);
00089
00090 double adjcp(rtcv_t *rtcm, int sat, int freq, double cp);
00091
00092 int lossoflock(rtcv_t *rtcm, int sat, int freq, int lock);
00093
00094 unsigned char snratio(double snr);
00095
00096 int obsindex3(obs_t *obs, gtime_t time, int sat);
00097
00098 int test_staid(rtcv_t *rtcm, int staid);
00099

```

```
00100 int decode_head1001(rtc_t *rtc, int *sync);
00101
00102 int decode_type1001(rtc_t *rtc);
00103
00104 int decode_type1002(rtc_t *rtc);
00105
00106 int decode_type1003(rtc_t *rtc);
00107
00108 int decode_type1004(rtc_t *rtc);
00109
00110 double getbits_38(const unsigned char *buff, int pos);
00111
00112 int decode_type1005(rtc_t *rtc);
00113
00114 int decode_type1006(rtc_t *rtc);
00115
00116 int decode_type1007(rtc_t *rtc);
00117
00118 int decode_type1008(rtc_t *rtc);
00119
00120 int decode_head1009(rtc_t *rtc, int *sync);
00121
00122 int decode_type1009(rtc_t *rtc);
00123
00124 int decode_type1010(rtc_t *rtc);
00125
00126 int decode_type1011(rtc_t *rtc);
00127
00128 int decode_type1012(rtc_t *rtc);
00129
00130 int decode_type1013(rtc_t *rtc);
00131
00132 int decode_type1019(rtc_t *rtc, bool pre_2009_file = false);
00133
00134 int decode_type1020(rtc_t *rtc);
00135
00136 int decode_type1021(rtc_t *rtc);
00137
00138 int decode_type1022(rtc_t *rtc);
00139
00140 int decode_type1023(rtc_t *rtc);
00141
00142 int decode_type1024(rtc_t *rtc);
00143
00144 int decode_type1025(rtc_t *rtc);
00145
00146 int decode_type1026(rtc_t *rtc);
00147
00148 int decode_type1027(rtc_t *rtc);
00149
00150 int decode_type1029(rtc_t *rtc);
00151
00152 int decode_type1030(rtc_t *rtc);
00153
00154 int decode_type1031(rtc_t *rtc);
00155
00156 int decode_type1032(rtc_t *rtc);
00157
00158 int decode_type1033(rtc_t *rtc);
00159
00160 int decode_type1034(rtc_t *rtc);
00161
00162 int decode_type1035(rtc_t *rtc);
00163
00164 int decode_type1037(rtc_t *rtc);
00165
00166 int decode_type1038(rtc_t *rtc);
00167
00168 int decode_type1039(rtc_t *rtc);
00169
00170 int decode_type1044(rtc_t *rtc, bool pre_2009_file = false);
00171
00172 int decode_type1045(rtc_t *rtc);
00173
00174 int decode_type1046(rtc_t *rtc);
00175
00176 int decode_type1047(rtc_t *rtc);
00177
00178 int decode_type1063(rtc_t *rtc);
00179
00180 int decode_ssrl_head(rtc_t *rtc, int sys, int *sync, int *iod,
00181 double *udint, int *refd, int *hsize);
00182
00183 int decode_ssrl2_head(rtc_t *rtc, int sys, int *sync, int *iod,
00184 double *udint, int *hsize);
00185
00186 int decode_ssrl7_head(rtc_t *rtc, int sys, int *sync, int *iod,
```

```

00187     double *udint, int *dispe, int *mw, int *hsize);
00188
00189 int decode_ssrl(rtcmt_t *rtcmt, int sys);
00190
00191 int decode_ssrl2(rtcmt_t *rtcmt, int sys);
00192
00193 int decode_ssrl3(rtcmt_t *rtcmt, int sys);
00194
00195 int decode_ssrl4(rtcmt_t *rtcmt, int sys);
00196
00197 int decode_ssrl5(rtcmt_t *rtcmt, int sys);
00198
00199 int decode_ssrl6(rtcmt_t *rtcmt, int sys);
00200
00201 int decode_ssrl7(rtcmt_t *rtcmt, int sys);
00202
00203 void sigindex(int sys, const unsigned char *code, const int *freq, int n,
00204               const char *opt, int *ind);
00205
00206 void save_msm_obs(rtcmt_t *rtcmt, int sys, msm_h_t *h, const double *r,
00207                  const double *pr, const double *cp, const double *rr,
00208                  const double *rrf, const double *cnr, const int *lock,
00209                  const int *ex, const int *half);
00210
00211 int decode_msm_head(rtcmt_t *rtcmt, int sys, int *sync, int *iod,
00212                    msm_h_t *h, int *hsize);
00213
00214 int decode_msm0(rtcmt_t *rtcmt, int sys);
00215
00216 int decode_msm4(rtcmt_t *rtcmt, int sys);
00217
00218 int decode_msm5(rtcmt_t *rtcmt, int sys);
00219
00220 int decode_msm6(rtcmt_t *rtcmt, int sys);
00221
00222 int decode_msm7(rtcmt_t *rtcmt, int sys);
00223
00224 int decode_type1230(rtcmt_t *rtcmt);
00225
00226 int decode_rtcmt3(rtcmt_t *rtcmt);
00227
00228
00229 #endif

```

## 11.218 rtklib\_rtkcmn.h File Reference

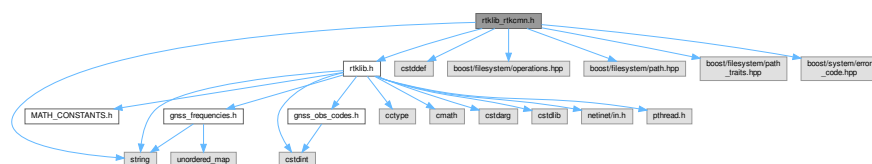
rtklib common functions

```

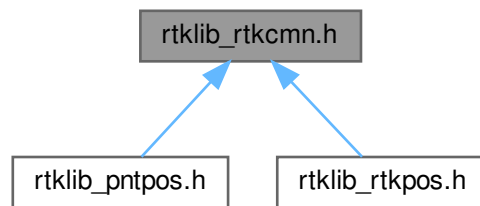
#include "rtklib.h"
#include <cstddef>
#include <string>
#include <boost/filesystem/operations.hpp>
#include <boost/filesystem/path.hpp>
#include <boost/filesystem/path_traits.hpp>
#include <boost/system/error_code.hpp>

```

Include dependency graph for rtklib\_rtkcmn.h:



This graph shows which files directly or indirectly include this file:



### Macros

- `#define Rx(t, X)`
- `#define Ry(t, X)`
- `#define Rz(t, X)`

### Functions

- `char * strncpy_no_trunc (char *out, size_t outsz, const char *in, size_t insz)`
- `void fatalerr (const char *format,...)`
- `int satno (int sys, int prn)`
- `int satsys (int sat, int *prn)`
- `int satid2no (const char *id)`
- `std::string satno2id (int sat)`
- `int satexclude (int sat, int svh, const procpt\_t *opt)`
- `int testsnr (int base, int freq, double el, double snr, const snrmask\_t *mask)`
- `unsigned char obs2code (const char *obs, int *freq)`
- `char * code2obs (unsigned char code, int *freq)`
- `void setcodepri (int sys, int freq, const char *pri)`
- `int getcodepri (int sys, unsigned char code, const char *opt)`
- `unsigned int getbitu (const unsigned char *buff, int pos, int len)`
- `int getbits (const unsigned char *buff, int pos, int len)`
- `void setbitu (unsigned char *buff, int pos, int len, unsigned int data)`
- `void setbits (unsigned char *buff, int pos, int len, int data)`
- `unsigned int rtk_crc32 (const unsigned char *buff, int len)`
- `unsigned int rtk_crc24q (const unsigned char *buff, int len)`
- `unsigned short rtk_crc16 (const unsigned char *buff, int len)`
- `int decode_word (unsigned int word, unsigned char *data)`
- `double * mat (int n, int m)`
- `int * imat (int n, int m)`
- `double * zeros (int n, int m)`
- `double * eye (int n)`
- `double dot (const double *a, const double *b, int n)`
- `double norm_rtk (const double *a, int n)`
- `void cross3 (const double *a, const double *b, double *c)`
- `int normv3 (const double *a, double *b)`
- `void matcpy (double *A, const double *B, int n, int m)`
- `void matmul (const char *tr, int n, int k, int m, double alpha, const double *A, const double *B, double beta, double *C)`

- int **matinv** (double \*A, int n)
- int **solve** (const char \*tr, const double \*A, const double \*Y, int n, int m, double \*X)
- int **lsq** (const double \*A, const double \*y, int n, int m, double \*x, double \*Q)
- int **filter\_** (const double \*x, const double \*P, const double \*H, const double \*v, const double \*R, int n, int m, double \*xp, double \*Pp)
- int **filter** (double \*x, double \*P, const double \*H, const double \*v, const double \*R, int n, int m)
- int **smoother** (const double \*xf, const double \*Qf, const double \*xb, const double \*Qb, int n, double \*xs, double \*Qs)
- void **matfprint** (const double A[], int n, int m, int p, int q, FILE \*fp)
- void **matsprint** (const double A[], int n, int m, int p, int q, std::string &buffer)
- void **matprint** (const double A[], int n, int m, int p, int q)
- double **str2num** (const char \*s, int i, int n)
- int **str2time** (const char \*s, int i, int n, [gtime\\_t](#) \*t)
- [gtime\\_t](#) **epoch2time** (const double \*ep)
- void **time2epoch** ([gtime\\_t](#) t, double \*ep)
- [gtime\\_t](#) **gpst2time** (int week, double sec)
- double **time2gpst** ([gtime\\_t](#) t, int \*week)
- [gtime\\_t](#) **gst2time** (int week, double sec)
- double **time2gst** ([gtime\\_t](#) t, int \*week)
- [gtime\\_t](#) **bdt2time** (int week, double sec)
- double **time2bdt** ([gtime\\_t](#) t, int \*week)
- [gtime\\_t](#) **timeadd** ([gtime\\_t](#) t, double sec)
- double **timediff** ([gtime\\_t](#) t1, [gtime\\_t](#) t2)
- double **timediffweekcrossover** ([gtime\\_t](#) t1, [gtime\\_t](#) t2)
- [gtime\\_t](#) **timeget** ()
- void **timeset** ([gtime\\_t](#) t)
- int **read\_leaps\_text** (FILE \*fp)
- int **read\_leaps\_usno** (FILE \*fp)
- int **read\_leaps** (const char \*file)
- [gtime\\_t](#) **gpst2utc** ([gtime\\_t](#) t)
- [gtime\\_t](#) **utc2gpst** ([gtime\\_t](#) t)
- [gtime\\_t](#) **gpst2bdt** ([gtime\\_t](#) t)
- [gtime\\_t](#) **bdt2gpst** ([gtime\\_t](#) t)
- double **time2sec** ([gtime\\_t](#) time, [gtime\\_t](#) \*day)
- double **utc2gmst** ([gtime\\_t](#) t, double ut1\_utc)
- void **time2str** ([gtime\\_t](#) t, char \*s, int n)
- char \* **time\_str** ([gtime\\_t](#) t, int n)
- double **time2doy** ([gtime\\_t](#) t)
- int **adjgpsweek** (int week, bool pre\_2009\_file=false)
- unsigned int **tickget** ()
- void **sleepms** (int ms)
- void **deg2dms** (double deg, double \*dms, int ndec)
- void **deg2dms** (double deg, double \*dms)
- double **dms2deg** (const double \*dms)
- void **ecf2pos** (const double \*r, double \*pos)
- void **pos2ecf** (const double \*pos, double \*r)
- void **xyz2enu** (const double \*pos, double \*E)
- void **ecf2enu** (const double \*pos, const double \*r, double \*e)
- void **enu2ecf** (const double \*pos, const double \*e, double \*r)
- void **covenu** (const double \*pos, const double \*P, double \*Q)
- void **covecf** (const double \*pos, const double \*Q, double \*P)
- void **ast\_args** (double t, double \*f)
- void **nut\_iau1980** (double t, const double \*f, double \*dpsi, double \*deps)
- void **eci2ecf** ([gtime\\_t](#) tutc, const double \*erpv, double \*U, double \*gmst)
- int **decodef** (char \*p, int n, double \*v)



- void **addpcv** (const [pcv\\_t](#) \*pcv, [pcvs\\_t](#) \*pcvs)
- int **readngspcv** (const char \*file, [pcvs\\_t](#) \*pcvs)
- int **readantex** (const char \*file, [pcvs\\_t](#) \*pcvs)
- int **readpcv** (const char \*file, [pcvs\\_t](#) \*pcvs)
- [pcv\\_t](#) \* **searchpcv** (int sat, const char \*type, [gtime\\_t](#) time, const [pcvs\\_t](#) \*pcvs)
- void **readpos** (const char \*file, const char \*rcv, double \*pos)
- int **readblqrecord** (FILE \*fp, double \*odisp)
- int **readblq** (const char \*file, const char \*sta, double \*odisp)
- int **readerp** (const char \*file, [erp\\_t](#) \*erp)
- int **geterp** (const [erp\\_t](#) \*erp, [gtime\\_t](#) time, double \*erpv)
- int **cmpeph** (const void \*p1, const void \*p2)
- void **uniqueph** ([nav\\_t](#) \*nav)
- int **cmpgeph** (const void \*p1, const void \*p2)
- void **uniggeph** ([nav\\_t](#) \*nav)
- int **cmpseph** (const void \*p1, const void \*p2)
- void **uniqseph** ([nav\\_t](#) \*nav)
- void **uniqnav** ([nav\\_t](#) \*nav)
- int **cmpobs** (const void \*p1, const void \*p2)
- int **sortobs** ([obs\\_t](#) \*obs)
- int **screent** ([gtime\\_t](#) time, [gtime\\_t](#) ts, [gtime\\_t](#) te, double tint)
- int **readnav** (const char \*file, [nav\\_t](#) \*nav)
- int **savenav** (const char \*file, const [nav\\_t](#) \*nav)
- void **freeobs** ([obs\\_t](#) \*obs)
- void **freenav** ([nav\\_t](#) \*nav, int opt)
- void **traceopen** (const char \*file)
- void **traceclose** ()
- void **tracelevel** (int level)
- void **traceswap** ()
- void **trace** (int level, const char \*format,...)
- void **tracet** (int level, const char \*format,...)
- void **tracemat** (int level, const double \*A, int n, int m, int p, int q)
- void **traceobs** (int level, const [obsd\\_t](#) \*obs, int n)
- int **execcmd** (const char \*cmd)
- void **createdir** (fs::path const &path)
- int **reppath** (std::string const &path, std::string &rpath, [gtime\\_t](#) time, const char \*rov, const char \*base)
- double **satwavelen** (int sat, int frq, const [nav\\_t](#) \*nav)
- double **geodist** (const double \*rs, const double \*rr, double \*e)
- double **satazel** (const double \*pos, const double \*e, double \*azel)
- void **dops** (int ns, const double \*azel, double elmin, double \*dop)
- double **ionmodel** ([gtime\\_t](#) t, const double \*ion, const double \*pos, const double \*azel)
- double **ionmapf** (const double \*pos, const double \*azel)
- double **ionppp** (const double \*pos, const double \*azel, double re, double hion, double \*posp)
- double **tropmodel** ([gtime\\_t](#) time, const double \*pos, const double \*azel, double humi)
- double **interp** (const double coef[], double lat)
- double **mapf** (double el, double a, double b, double c)
- double **nmf** ([gtime\\_t](#) time, const double pos[], const double azel[], double \*mapfw)
- double **tropmapf** ([gtime\\_t](#) time, const double pos[], const double azel[], double \*mapfw)
- double **interpvar** (double ang, const double \*var)
- void **antmodel** (const [pcv\\_t](#) \*pcv, const double \*del, const double \*azel, int opt, double \*dant)
- void **antmodel\_s** (const [pcv\\_t](#) \*pcv, double nadir, double \*dant)
- void **sunmoonpos\_eci** ([gtime\\_t](#) tut, double \*rsun, double \*rmoon)
- void **sunmoonpos** ([gtime\\_t](#) tutc, const double \*erpv, double \*rsun, double \*rmoon, double \*gmst)
- void **csmooth** ([obs\\_t](#) \*obs, int ns)
- int **rtk\_uncompress** (const char \*file, char \*uncfile)
- int **expath** (const char \*path, char \*paths[], int nmax)
- void **windupcorr** ([gtime\\_t](#) time, const double \*rs, const double \*rr, double \*phw)

## 11.218.1 Detailed Description

rtklib common functions

### Authors

- 2007-2013, T. Takasu
- 2017, Javier Arribas
- 2017, Carles Fernandez

This is a derived work from RTKLIB <http://www.rtklib.com/> The original source code at <https://github.com/tomojitakasu/RTKLIB> is released under the BSD 2-clause license with an additional exclusive clause that does not apply here. This additional clause is reproduced below:

" The software package includes some companion executive binaries or shared libraries necessary to execute APs on Windows. These licenses succeed to the original ones of these software. "

Neither the executive binaries nor the shared libraries are required by, used or included in GNSS-SDR.

Copyright (C) 2007-2013, T. Takasu Copyright (C) 2017, Javier Arribas Copyright (C) 2017, Carles Fernandez All rights reserved.

SPDX-License-Identifier: BSD-2-Clause

References : [1] IS-GPS-200M, Navstar GPS Space Segment/Navigation User Interfaces, May, 2021 [2] RTCA/DO-229C, Minimum operational performance standards for global positioning system/wide area augmentation system airborne equipment, RTCA inc, November 28, 2001 [3] M.Rothacher, R.Schmid, ANTEX: The Antenna Exchange Format Version 1.4, 15 September, 2010 [4] A.Gelb ed., Applied Optimal Estimation, The M.I.T Press, 1974 [5] A. E.Niell, Global mapping functions for the atmosphere delay at radio wavelengths, Journal of geophysical research, 1996 [6] W.Gurtner and L.Estey, RINEX The Receiver Independent Exchange Format Version 3.00, November 28, 2007 [7] J.Kouba, A Guide to using International GNSS Service (IGS) products, May 2009 [8] China Satellite Navigation Office, BeiDou navigation satellite system signal in space interface control document, open service signal B1I (version 1.0), Dec 2012 [9] J.Boehm, A.Niell, P.Tregoning and H.Shuh, Global Mapping Function (GMF): A new empirical mapping function base on numerical weather model data, Geophysical Research Letters, 33, L07304, 2006 [10] GLONASS/GPS/Galileo/Compass/SBAS NV08C receiver series B1NR interface protocol specification ver.1.3, August, 2012

Definition in file [rtklib\\_rtkcmn.h](#).

## 11.218.2 Macro Definition Documentation

### 11.218.2.1 Rx

```
#define Rx(
```

```
    t,  
    X)
```

Value:

```
do  
{  
    (X)[0] = 1.0;  
    (X)[1] = (X)[2] = (X)[3] = (X)[6] = 0.0;  
    (X)[4] = (X)[8] = cos(t);  
    (X)[7] = sin(t);  
    (X)[5] = -(X)[7];  
}  
while (0)
```

Definition at line 85 of file [rtklib\\_rtkcmn.h](#).

### 11.218.2.2 Ry

```
#define Ry(
```

```
    t,  
    X)
```

Value:

```
do  
{  
    (X)[4] = 1.0;  
    (X)[1] = (X)[3] = (X)[5] = (X)[7] = 0.0;  
    (X)[0] = (X)[8] = cos(t);  
    (X)[2] = sin(t);  
    (X)[6] = -(X)[2];  
}  

```

```
while (0)
```

Definition at line 96 of file [rtklib\\_rtkcmn.h](#).

### 11.218.2.3 Rz

```
#define Rz(
```

```
    t,  
    X)
```

Value:

```
do  
{  
    (X)[8] = 1.0;  
    (X)[2] = (X)[5] = (X)[6] = (X)[7] = 0.0;  
    (X)[0] = (X)[4] = cos(t);  
    (X)[3] = sin(t);  
    (X)[1] = -(X)[3];  
}  
while (0)
```

Definition at line 107 of file [rtklib\\_rtkcmn.h](#).

## 11.219 rtklib\_rtkcmn.h

[Go to the documentation of this file.](#)

```
00001 /*!  
00002 * \file rtklib_rtkcmn.h  
00003 * \brief rtklib common functions  
00004 * \authors <ul>  
00005 *     <li> 2007-2013, T. Takasu  
00006 *     <li> 2017, Javier Arribas  
00007 *     <li> 2017, Carles Fernandez  
00008 * </ul>  
00009 *  
00010 * This is a derived work from RTKLIB http://www.rtklib.com/  
00011 * The original source code at https://github.com/tomojitakasu/RTKLIB is  
00012 * released under the BSD 2-clause license with an additional exclusive clause  
00013 * that does not apply here. This additional clause is reproduced below:  
00014 *  
00015 * " The software package includes some companion executive binaries or shared  
00016 * libraries necessary to execute APs on Windows. These licenses succeed to the  
00017 * original ones of these software. "  
00018 *  
00019 * Neither the executive binaries nor the shared libraries are required by, used  
00020 * or included in GNSS-SDR.  
00021 *  
00022 * -----  
00023 * Copyright (C) 2007-2013, T. Takasu  
00024 * Copyright (C) 2017, Javier Arribas  
00025 * Copyright (C) 2017, Carles Fernandez  
00026 * All rights reserved.  
00027 *  
00028 * SPDX-License-Identifier: BSD-2-Clause  
00029 *  
00030 *  
00031 * References :  
00032 * [1] IS-GPS-200M, Navstar GPS Space Segment/Navigation User Interfaces,  
00033 *     May, 2021  
00034 * [2] RTCA/DO-229C, Minimum operational performance standards for global  
00035 *     positioning system/wide area augmentation system airborne equipment,  
00036 *     RTCA inc, November 28, 2001  
00037 * [3] M.Rothacher, R.Schmid, ANTEX: The Antenna Exchange Format Version 1.4,  
00038 *     15 September, 2010  
00039 * [4] A.Gelb ed., Applied Optimal Estimation, The M.I.T Press, 1974  
00040 * [5] A.E.Niell, Global mapping functions for the atmosphere delay at radio  
00041 *     wavelengths, Journal of geophysical research, 1996  
00042 * [6] W.Gurtner and L.Estey, RINEX The Receiver Independent Exchange Format  
00043 *     Version 3.00, November 28, 2007  
00044 * [7] J.Kouba, A Guide to using International GNSS Service (IGS) products,  
00045 *     May 2009  
00046 * [8] China Satellite Navigation Office, BeiDou navigation satellite system  
00047 *     signal in space interface control document, open service signal B1I  
00048 *     (version 1.0), Dec 2012  
00049 * [9] J.Boehm, A.Niell, P.Tregoning and H.Shuh, Global Mapping Function  
00050 *     (GMF): A new empirical mapping function base on numerical weather  
00051 *     model data, Geophysical Research Letters, 33, L07304, 2006  
00052 * [10] GLONASS/GPS/Galileo/Compass/SBAS NV08C receiver series BINR interface  
00053 *     protocol specification ver.1.3, August, 2012  
00054 *  
00055 * -----  
00056 */
```

```

00057
00058 #ifndef GNSS_SDR_RTKLIB_RTKCMN_H
00059 #define GNSS_SDR_RTKLIB_RTKCMN_H
00060
00061 #include "rtklib.h"
00062 #include <stddef>
00063 #include <string>
00064
00065 #if HAS_STD_FILESYSTEM
00066 #include <system_error>
00067 namespace errorlib = std;
00068 #if HAS_STD_FILESYSTEM_EXPERIMENTAL
00069 #include <experimental/filesystem>
00070 namespace fs = std::experimental::filesystem;
00071 #else
00072 #include <filesystem>
00073 namespace fs = std::filesystem;
00074 #endif
00075 #else
00076 #include <boost/filesystem/operations.hpp> // for create_directories, exists
00077 #include <boost/filesystem/path.hpp> // for path, operator<
00078 #include <boost/filesystem/path_traits.hpp> // for filesystem
00079 #include <boost/system/error_code.hpp> // for error_code
00080 namespace fs = boost::filesystem;
00081 namespace errorlib = boost::system;
00082 #endif
00083
00084 /* coordinate rotation matrix -----*/
00085 #define Rx(t, X)
00086     do
00087     {
00088         (X)[0] = 1.0;
00089         (X)[1] = (X)[2] = (X)[3] = (X)[6] = 0.0;
00090         (X)[4] = (X)[8] = cos(t);
00091         (X)[7] = sin(t);
00092         (X)[5] = -(X)[7];
00093     }
00094     while (0)
00095
00096 #define Ry(t, X)
00097     do
00098     {
00099         (X)[4] = 1.0;
00100         (X)[1] = (X)[3] = (X)[5] = (X)[7] = 0.0;
00101         (X)[0] = (X)[8] = cos(t);
00102         (X)[2] = sin(t);
00103         (X)[6] = -(X)[2];
00104     }
00105     while (0)
00106
00107 #define Rz(t, X)
00108     do
00109     {
00110         (X)[8] = 1.0;
00111         (X)[2] = (X)[5] = (X)[6] = (X)[7] = 0.0;
00112         (X)[0] = (X)[4] = cos(t);
00113         (X)[3] = sin(t);
00114         (X)[1] = -(X)[3];
00115     }
00116     while (0)
00117
00118 char *strncpy_no_trunc(char *out, size_t outsz, const char *in, size_t insz);
00119 void fatalerr(const char *format, ...);
00120 int satno(int sys, int prn);
00121 int satsys(int sat, int *prn);
00122 int satid2no(const char *id);
00123 std::string satno2id(int sat);
00124 int satexclude(int sat, int svh, const prcopt_t *opt);
00125 int testsnr(int base, int freq, double el, double snr, const snrmask_t *mask);
00126 unsigned char obs2code(const char *obs, int *freq);
00127 char *code2obs(unsigned char code, int *freq);
00128 void setcodepri(int sys, int freq, const char *pri);
00129 int getcodepri(int sys, unsigned char code, const char *opt);
00130 unsigned int getbitu(const unsigned char *buff, int pos, int len);
00131 int getbits(const unsigned char *buff, int pos, int len);
00132 void setbitu(unsigned char *buff, int pos, int len, unsigned int data);
00133 void setbits(unsigned char *buff, int pos, int len, int data);
00134 unsigned int rtk_crc32(const unsigned char *buff, int len);
00135 unsigned int rtk_crc24q(const unsigned char *buff, int len);
00136 unsigned short rtk_crc16(const unsigned char *buff, int len);
00137 int decode_word(unsigned int word, unsigned char *data);
00138 double *mat(int n, int m);
00139 int *imat(int n, int m);
00140 double *zeros(int n, int m);
00141 double *eye(int n);
00142 double dot(const double *a, const double *b, int n);
00143 double norm_rtk(const double *a, int n);

```

```
00144 void cross3(const double *a, const double *b, double *c);
00145 int normv3(const double *a, double *b);
00146 void matcpy(double *A, const double *B, int n, int m);
00147 void matmul(const char *tr, int n, int k, int m, double alpha,
00148     const double *A, const double *B, double beta, double *C);
00149 int matinv(double *A, int n);
00150 int solve(const char *tr, const double *A, const double *Y, int n,
00151     int m, double *X);
00152 int lsq(const double *A, const double *y, int n, int m, double *x,
00153     double *Q);
00154 int filter_(const double *x, const double *P, const double *H,
00155     const double *v, const double *R, int n, int m,
00156     double *xp, double *Pp);
00157 int filter(double *x, double *P, const double *H, const double *v,
00158     const double *R, int n, int m);
00159 int smoother(const double *xf, const double *Qf, const double *xb,
00160     const double *Qb, int n, double *xs, double *Qs);
00161 void matfprint(const double A[], int n, int m, int p, int q, FILE *fp);
00162 void matsprint(const double A[], int n, int m, int p, int q, std::string &buffer);
00163 void matprint(const double A[], int n, int m, int p, int q);
00164 double str2num(const char *s, int i, int n);
00165 int str2time(const char *s, int i, int n, gtime_t *t);
00166 gtime_t epoch2time(const double *ep);
00167 void time2epoch(gtime_t t, double *ep);
00168 gtime_t gpst2time(int week, double sec);
00169 double time2gpst(gtime_t t, int *week);
00170 gtime_t gst2time(int week, double sec);
00171 double time2gst(gtime_t t, int *week);
00172 gtime_t bdt2time(int week, double sec);
00173 double time2bdt(gtime_t t, int *week);
00174 gtime_t timeadd(gtime_t t, double sec);
00175 double timediff(gtime_t t1, gtime_t t2);
00176 double timediffweekcrossover(gtime_t t1, gtime_t t2);
00177 gtime_t timeget();
00178 void timeset(gtime_t t);
00179 int read_leaps_text(FILE *fp);
00180 int read_leaps_usno(FILE *fp);
00181 int read_leaps(const char *file);
00182 gtime_t gpst2utc(gtime_t t);
00183 gtime_t utc2gpst(gtime_t t);
00184 gtime_t gpst2bdt(gtime_t t);
00185 gtime_t bdt2gpst(gtime_t t);
00186 double time2sec(gtime_t time, gtime_t *day);
00187 double utc2gmst(gtime_t t, double ut1_utc);
00188 void time2str(gtime_t t, char *s, int n);
00189 char *time_str(gtime_t t, int n);
00190 double time2doy(gtime_t t);
00191 int adjgpsweek(int week, bool pre_2009_file = false);
00192 unsigned int tickget();
00193 void sleepms(int ms);
00194 void deg2dms(double deg, double *dms, int ndec);
00195 void deg2dms(double deg, double *dms);
00196 double dms2deg(const double *dms);
00197 void ecef2pos(const double *r, double *pos);
00198 void pos2ecef(const double *pos, double *r);
00199 void xyz2enu(const double *pos, double *E);
00200 void ecef2enu(const double *pos, const double *r, double *e);
00201 void enu2ecef(const double *pos, const double *e, double *r);
00202 void covenu(const double *pos, const double *P, double *Q);
00203 void covcecef(const double *pos, const double *Q, double *P);
00204 void ast_args(double t, double *f);
00205 void nut_iau1980(double t, const double *f, double *dpsi, double *deps);
00206 void eci2ecef(gtime_t tutc, const double *erpv, double *U, double *gmst);
00207 int decodef(char *p, int n, double *v);
00208 void addpcv(const pcv_t *pcv, pcv_t *pcvs);
00209 int readngspcv(const char *file, pcv_t *pcvs);
00210 int readantex(const char *file, pcv_t *pcvs);
00211 int readpcv(const char *file, pcv_t *pcvs);
00212 pcv_t *searchpcv(int sat, const char *type, gtime_t time,
00213     const pcv_t *pcvs);
00214 void readpos(const char *file, const char *rcv, double *pos);
00215 int readblqrecord(FILE *fp, double *odisp);
00216 int readblq(const char *file, const char *sta, double *odisp);
00217 int readerp(const char *file, erp_t *erp);
00218 int geterp(const erp_t *erp, gtime_t time, double *erpv);
00219 int cmppeph(const void *p1, const void *p2);
00220 void uniqeph(nav_t *nav);
00221 int cmpgeph(const void *p1, const void *p2);
00222 void uniqgeph(nav_t *nav);
00223 int cmpseph(const void *p1, const void *p2);
00224 void uniqseph(nav_t *nav);
00225 void unignav(nav_t *nav);
00226 int cmpobs(const void *p1, const void *p2);
00227 int sortobs(obs_t *obs);
00228 int screen(gtime_t time, gtime_t ts, gtime_t te, double tint);
00229 int readnav(const char *file, nav_t *nav);
00230 int savenav(const char *file, const nav_t *nav);
```

```

00231 void freeobs(obs_t *obs);
00232 void freenav(nav_t *nav, int opt);
00233
00234 void traceopen(const char *file);
00235 void traceclose();
00236 void tracelevel(int level);
00237 void traceswap();
00238 void trace(int level, const char *format, ...);
00239 void tracet(int level, const char *format, ...);
00240 void tracemat(int level, const double *A, int n, int m, int p, int q);
00241 void traceobs(int level, const obsd_t *obs, int n);
00242 // void tracenav(int level, const nav_t *nav);
00243 // void tracegnav(int level, const nav_t *nav);
00244 // void tracehnav(int level, const nav_t *nav);
00245 // void tracepeph(int level, const nav_t *nav);
00246 // void tracepclk(int level, const nav_t *nav);
00247 // void traceb (int level, const unsigned char *p, int n);
00248
00249 int execcmd(const char *cmd);
00250 void createdir(fs::path const &path);
00251 int reppath(std::string const &path, std::string &rpath, gtime_t time, const char *rov,
00252     const char *base);
00253 double satwavelen(int sat, int frq, const nav_t *nav);
00254 double geodist(const double *rs, const double *rr, double *e);
00255 double satazel(const double *pos, const double *e, double *azel);
00256
00257 void dops(int ns, const double *azel, double elmin, double *dop);
00258 double ionmodel(gtime_t t, const double *ion, const double *pos,
00259     const double *azel);
00260 double ionmapf(const double *pos, const double *azel);
00261 double ionppp(const double *pos, const double *azel, double re,
00262     double hion, double *posp);
00263 double tropmodel(gtime_t time, const double *pos, const double *azel,
00264     double humi);
00265 double interpc(const double coef[], double lat);
00266 double mapf(double el, double a, double b, double c);
00267 double nmf(gtime_t time, const double pos[], const double azel[],
00268     double *mapfw);
00269 double tropmapf(gtime_t time, const double pos[], const double azel[],
00270     double *mapfw);
00271 double interpvar(double ang, const double *var);
00272
00273 void antmodel(const pcv_t *pcv, const double *del, const double *azel,
00274     int opt, double *dant);
00275
00276 void antmodel_s(const pcv_t *pcv, double nadir, double *dant);
00277 void sunmoonpos_eci(gtime_t tut, double *rsun, double *rmoon);
00278 void sunmoonpos(gtime_t tutc, const double *erpv, double *rsun,
00279     double *rmoon, double *gmst);
00280 void csmooth(obs_t *obs, int ns);
00281 int rtk_uncompress(const char *file, char *uncfile);
00282 int expath(const char *path, char *paths[], int nmax);
00283 void windupcorr(gtime_t time, const double *rs, const double *rr, double *phw);
00284
00285 #endif // GNSS_SDR_RTKLIB_RTKCMN_H

```

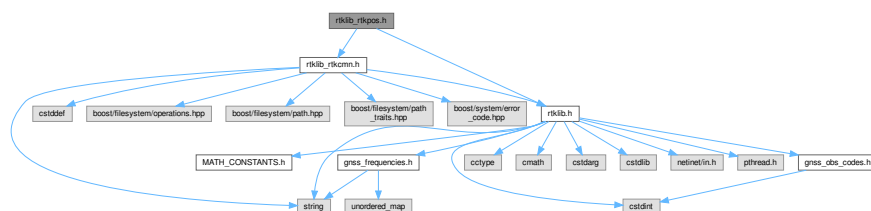
## 11.220 rtklib\_rtkpos.h File Reference

rtklib ppp-related functions

```
#include "rtklib.h"
```

```
#include "rtklib_rtkcmn.h"
```

Include dependency graph for rtklib\_rtkpos.h:



### Macros

- `#define IL_RTK(s, opt)`

- #define **IT\_RTK**(r, opt)
- #define **IL\_RTK**(f, opt)
- #define **IB\_RTK**(s, f, opt)

## Functions

- int **rtkopenstat** (const char \*file, int level)
- void **rtkclosestat** ()
- void **rtkoutstat** (rtk\_t \*rtk)
- void **swapsolstat** ()
- void **outsolstat** (rtk\_t \*rtk)
- void **errmsg** (rtk\_t \*rtk, const char \*format,...)
- double **sdobs** (const obsd\_t \*obs, int i, int j, int f)
- double **gfobs\_L1L2** (const obsd\_t \*obs, int i, int j, const double \*lam)
- double **gfobs\_L1L5** (const obsd\_t \*obs, int i, int j, const double \*lam)
- double **varerr** (int sat, int sys, double el, double bl, double dt, int f, const prcopt\_t \*opt)
- double **baseline** (const double \*ru, const double \*rb, double \*dr)
- void **initx\_rtk** (rtk\_t \*rtk, double xi, double var, int i)
- int **selsat** (const obsd\_t \*obs, const double \*azel, int nu, int nr, const prcopt\_t \*opt, int \*sat, int \*iu, int \*ir)
- void **udpos** (rtk\_t \*rtk, double tt)
- void **udion** (rtk\_t \*rtk, double tt, double bl, const int \*sat, int ns)
- void **udtrop** (rtk\_t \*rtk, double tt, double bl)
- void **udrcvbias** (rtk\_t \*rtk, double tt)
- void **detslp\_ll** (rtk\_t \*rtk, const obsd\_t \*obs, int i, int rcv)
- void **detslp\_gf\_L1L2** (rtk\_t \*rtk, const obsd\_t \*obs, int i, int j, const nav\_t \*nav)
- void **detslp\_gf\_L1L5** (rtk\_t \*rtk, const obsd\_t \*obs, int i, int j, const nav\_t \*nav)
- void **detslp\_dop** (rtk\_t \*rtk, const obsd\_t \*obs, int i, int rcv, const nav\_t \*nav)
- void **udbias** (rtk\_t \*rtk, double tt, const obsd\_t \*obs, const int \*sat, const int \*iu, const int \*ir, int ns, const nav\_t \*nav)
- void **udstate** (rtk\_t \*rtk, const obsd\_t \*obs, const int \*sat, const int \*iu, const int \*ir, int ns, const nav\_t \*nav)
- void **zdres\_sat** (int base, double r, const obsd\_t \*obs, const nav\_t \*nav, const double \*azel, const double \*dant, const prcopt\_t \*opt, double \*y)
- int **zdres** (int base, const obsd\_t \*obs, int n, const double \*rs, const double \*dts, const int \*svh, const nav\_t \*nav, const double \*rr, const prcopt\_t \*opt, int index, double \*y, double \*e, double \*azel)
- int **validobs** (int i, int j, int f, int nf, const double \*y)
- void **ddcov** (const int \*nb, int n, const double \*Ri, const double \*Rj, int nv, double \*R)
- int **constbl** (rtk\_t \*rtk, const double \*x, const double \*P, double \*v, double \*H, double \*Ri, double \*Rj, int index)
- double **prectrop** (gtime\_t time, const double \*pos, int r, const double \*azel, const prcopt\_t \*opt, const double \*x, double \*dtdx)
- double **gloicbcorr** (int sat1, int sat2, const prcopt\_t \*opt, double lam1, double lam2, int f)
- int **test\_sys** (int sys, int m)
- int **ddres** (rtk\_t \*rtk, const nav\_t \*nav, double dt, const double \*x, const double \*P, const int \*sat, double \*y, const double \*e, double \*azel, const int \*iu, const int \*ir, int ns, double \*v, double \*H, double \*R, int \*vflg)
- double **intpres** (gtime\_t time, const obsd\_t \*obs, int n, const nav\_t \*nav, rtk\_t \*rtk, double \*y)
- int **ddmat** (rtk\_t \*rtk, double \*D)
- void **restamb** (rtk\_t \*rtk, const double \*bias, int nb, double \*xa)
- void **holdamb** (rtk\_t \*rtk, const double \*xa)
- int **resamb\_LAMBDA** (rtk\_t \*rtk, double \*bias, double \*xa)
- int **valpos** (rtk\_t \*rtk, const double \*v, const double \*R, const int \*vflg, int nv, double thres)
- int **relpos** (rtk\_t \*rtk, const obsd\_t \*obs, int nu, int nr, const nav\_t \*nav)
- void **rtkinit** (rtk\_t \*rtk, const prcopt\_t \*opt)
- void **rtkfree** (rtk\_t \*rtk)
- int **rtkpos** (rtk\_t \*rtk, const obsd\_t \*obs, int n, const nav\_t \*nav)

## Variables

- const double `VAR_POS` = `std::pow(30.0, 2.0)`
- const double `VAR_VEL` = `std::pow(10.0, 2.0)`
- const double `VAR_ACC` = `std::pow(10.0, 2.0)`
- const double `VAR_HWBIAS` = `std::pow(1.0, 2.0)`
- const double `VAR_GRA` = `std::pow(0.001, 2.0)`
- const double `INIT_ZWD` = 0.15
- const double `PRN_HWBIA` = 1E-6
- const double `MAXAC` = 30.0
- const double `VAR_HOLDAMB` = 0.001
- const double `TTOL_MOVEB` = (1.0 + 2 \* `DTTOL`)

## 11.220.1 Detailed Description

rtklib ppp-related functions

### Authors

- 2007-2013, T. Takasu
- 2017, Javier Arribas
- 2017, Carles Fernandez

This is a derived work from RTKLIB <http://www.rtklib.com/> The original source code at <https://github.com/tomojitakasu/RTKLIB> is released under the BSD 2-clause license with an additional exclusive clause that does not apply here. This additional clause is reproduced below:

" The software package includes some companion executive binaries or shared libraries necessary to execute APs on Windows. These licenses succeed to the original ones of these software. "

Neither the executive binaries nor the shared libraries are required by, used or included in GNSS-SDR.

Copyright (C) 2007-2013, T. Takasu Copyright (C) 2017, Javier Arribas Copyright (C) 2017, Carles Fernandez All rights reserved.

### 11.220.1.1 SPDX-License-Identifier: BSD-2-Clause

Definition in file [rtklib\\_rtkpos.h](#).

## 11.221 rtklib\_rtkpos.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file rtklib_rtkpos.h
00003  * \brief rtklib ppp-related functions
00004  * \authors <ul>
00005  *         <li> 2007-2013, T. Takasu
00006  *         <li> 2017, Javier Arribas
00007  *         <li> 2017, Carles Fernandez
00008  *         </ul>
00009  *
00010  * This is a derived work from RTKLIB http://www.rtklib.com/
00011  * The original source code at https://github.com/tomojitakasu/RTKLIB is
00012  * released under the BSD 2-clause license with an additional exclusive clause
00013  * that does not apply here. This additional clause is reproduced below:
00014  *
00015  * " The software package includes some companion executive binaries or shared
00016  * libraries necessary to execute APs on Windows. These licenses succeed to the
00017  * original ones of these software. "
00018  *
00019  * Neither the executive binaries nor the shared libraries are required by, used
00020  * or included in GNSS-SDR.
00021  *
00022  * -----
00023  * Copyright (C) 2007-2013, T. Takasu
00024  * Copyright (C) 2017, Javier Arribas
00025  * Copyright (C) 2017, Carles Fernandez
00026  * All rights reserved.
00027  *
00028  * SPDX-License-Identifier: BSD-2-Clause
00029  * -----
```



```

00030  */
00031
00032 #ifndef GNSS_SDR_RTKLIB_RKTPOS_H
00033 #define GNSS_SDR_RTKLIB_RKTPOS_H
00034
00035 #include "rtklib.h"
00036 #include "rtklib_rtkcmn.h"
00037
00038 /** \addtogroup PVT
00039  * \{ */
00040 /** \addtogroup RTKLIB_Library
00041  * \{ */
00042
00043
00044 /* constants/macros -----*/
00045 const double VAR_POS = std::pow(30.0, 2.0); /* initial variance of receiver pos (m^2) */
00046 const double VAR_VEL = std::pow(10.0, 2.0); /* initial variance of receiver vel ((m/s)^2) */
00047 const double VAR_ACC = std::pow(10.0, 2.0); /* initial variance of receiver acc ((m/ss)^2) */
00048 const double VAR_HWBIAS = std::pow(1.0, 2.0); /* initial variance of h/w bias ((m/MHz)^2) */
00049 const double VAR_GRA = std::pow(0.001, 2.0); /* initial variance of gradient (m^2) */
00050 const double INIT_ZWD = 0.15; /* initial zwd (m) */
00051
00052 const double PRN_HWBIA = 1E-6; /* process noise of h/w bias (m/MHz/sqrt(s)) */
00053 const double MAXAC = 30.0; /* max accel for doppler slip detection (m/s^2) */
00054
00055 const double VAR_HOLDAMB = 0.001; /* constraint to hold ambiguity (cycle^2) */
00056
00057 const double TTOL_MOVEB = (1.0 + 2 * DTTOL);
00058 /* time sync tolerance for moving-baseline (s) */
00059
00060 /* number of parameters (pos,ionos,tropos,hw-bias,phase-bias,real,estimated) */
00061
00062
00063 /* state variable index */
00064 // clang-format off
00065 #define IL_RTK(s, opt) (NP_RTK(opt) + (s)-1) /* ionos (s:satellite no)
00066 */
00067 #define IT_RTK(r, opt) (NP_RTK(opt) + NI_RTK(opt) + NT_RTK(opt) / 2 * (r)) /* tropos (r:0=rov,1:ref)
00068 */
00069 #define IL_RTK(f, opt) (NP_RTK(opt) + NI_RTK(opt) + NT_RTK(opt) + (f)) /* receiver h/w bias */
00070 #define IB_RTK(s, f, opt) (NR_RTK(opt) + MAXSAT * (f) + (s)-1) /* phase bias
00071 (s:satno,f:freq) */
00072 // clang-format on
00073
00074 int rtkopenstat(const char *file, int level);
00075
00076 void rtkclosestat();
00077
00078 void rtkoutstat(rtk_t *rtk);
00079
00080 void swapsolstat();
00081
00082 void outsolstat(rtk_t *rtk);
00083
00084 void errmsg(rtk_t *rtk, const char *format, ...);
00085
00086 double sobs(const obsd_t *obs, int i, int j, int f);
00087
00088 double gfobs_L1L2(const obsd_t *obs, int i, int j, const double *lam);
00089
00090 double gfobs_L1L5(const obsd_t *obs, int i, int j, const double *lam);
00091
00092 double varerr(int sat, int sys, double el, double bl, double dt, int f,
00093 const prcopt_t *opt);
00094
00095 double baseline(const double *ru, const double *rb, double *dr);
00096
00097 void initx_rtk(rtk_t *rtk, double xi, double var, int i);
00098
00099 int selsat(const obsd_t *obs, const double *azel, int nu, int nr,
00100 const prcopt_t *opt, int *sat, int *iu, int *ir);
00101
00102 void udpos(rtk_t *rtk, double tt);
00103
00104 void udion(rtk_t *rtk, double tt, double bl, const int *sat, int ns);
00105
00106 void uttrop(rtk_t *rtk, double tt, double bl);
00107
00108 void udrcvbias(rtk_t *rtk, double tt);
00109
00110 void detslp_ll(rtk_t *rtk, const obsd_t *obs, int i, int rcv);
00111
00112 void detslp_gf_L1L2(rtk_t *rtk, const obsd_t *obs, int i, int j,
00113 const nav_t *nav);
00114
00115 void detslp_gf_L1L5(rtk_t *rtk, const obsd_t *obs, int i, int j,
00116 const nav_t *nav);

```

```

00114
00115 void detslp_dop(rtk_t *rtk, const obsd_t *obs, int i, int rcv,
00116     const nav_t *nav);
00117
00118 void udbias(rtk_t *rtk, double tt, const obsd_t *obs, const int *sat,
00119     const int *iu, const int *ir, int ns, const nav_t *nav);
00120
00121 void udsta(rtk_t *rtk, const obsd_t *obs, const int *sat,
00122     const int *iu, const int *ir, int ns, const nav_t *nav);
00123
00124 void zdres_sat(int base, double r, const obsd_t *obs, const nav_t *nav,
00125     const double *azel, const double *dant,
00126     const prcopt_t *opt, double *y);
00127
00128 int zdres(int base, const obsd_t *obs, int n, const double *rs,
00129     const double *dts, const int *svh, const nav_t *nav,
00130     const double *rr, const prcopt_t *opt, int index, double *y,
00131     double *e, double *azel);
00132
00133 int validobs(int i, int j, int f, int nf, const double *y);
00134
00135 void ddcov(const int *nb, int n, const double *Ri, const double *Rj,
00136     int nv, double *R);
00137
00138 int constbl(rtk_t *rtk, const double *x, const double *P, double *v,
00139     double *H, double *Ri, double *Rj, int index);
00140
00141 double prectrop(gtime_t time, const double *pos, int r,
00142     const double *azel, const prcopt_t *opt, const double *x,
00143     double *dtdx);
00144
00145 double gloicbcorr(int sat1, int sat2, const prcopt_t *opt, double lam1,
00146     double lam2, int f);
00147
00148 int test_sys(int sys, int m);
00149
00150 int ddres(rtk_t *rtk, const nav_t *nav, double dt, const double *x,
00151     const double *P, const int *sat, double *y, const double *e,
00152     double *azel, const int *iu, const int *ir, int ns, double *v,
00153     double *H, double *R, int *vflg);
00154
00155 double intpres(gtime_t time, const obsd_t *obs, int n, const nav_t *nav,
00156     rtk_t *rtk, double *y);
00157
00158
00159 int ddmatrix(rtk_t *rtk, double *D);
00160
00161 void restamb(rtk_t *rtk, const double *bias, int nb, double *xa);
00162
00163 void holdamb(rtk_t *rtk, const double *xa);
00164
00165 int resamb_LAMBDA(rtk_t *rtk, double *bias, double *xa);
00166
00167 int valpos(rtk_t *rtk, const double *v, const double *R, const int *vflg,
00168     int nv, double thres);
00169
00170 int relpos(rtk_t *rtk, const obsd_t *obs, int nu, int nr,
00171     const nav_t *nav);
00172
00173 void rtkinit(rtk_t *rtk, const prcopt_t *opt);
00174
00175 void rtkfree(rtk_t *rtk);
00176
00177 int rtkpos(rtk_t *rtk, const obsd_t *obs, int n, const nav_t *nav);
00178
00179
00180 /** \} */
00181 /** \} */
00182 #endif // GNSS_SDR_RTKLIB_RKTIPOS_H

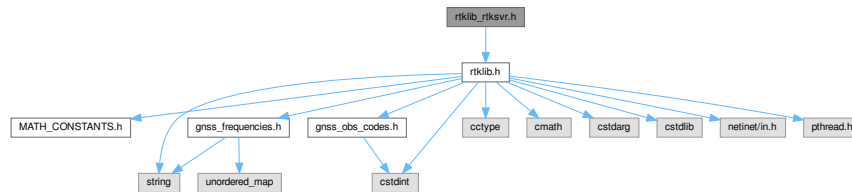
```

## 11.222 rtklib\_rtksvr.h File Reference

rtk server functions

```
#include "rtklib.h"
```

Include dependency graph for rtklib\_rtksvr.h:



## Functions

- void **writesolhead** ([stream\\_t](#) \*stream, const [solopt\\_t](#) \*solopt)
- void **saveoutbuf** ([rtksvr\\_t](#) \*svr, unsigned char \*buff, int n, int index)
- void **writesol** ([rtksvr\\_t](#) \*svr, int index)
- void **updatenav** ([nav\\_t](#) \*nav)
- void **updatefcu** ([rtksvr\\_t](#) \*svr)
- void **updatesvr** ([rtksvr\\_t](#) \*svr, int ret, [obs\\_t](#) \*obs, [nav\\_t](#) \*nav, int sat, [sbsmsg\\_t](#) \*sbsmsg, int index, int iobs)
- int **decoderaw** ([rtksvr\\_t](#) \*svr, int index)
- void **decodefile** ([rtksvr\\_t](#) \*svr, int index)
- void \* **rtksvrthread** (void \*arg)
- int **rtksvrinit** ([rtksvr\\_t](#) \*svr)
- void **rtksvrfree** ([rtksvr\\_t](#) \*svr)
- void **rtksvrlock** ([rtksvr\\_t](#) \*svr)
- void **rtksvrunlock** ([rtksvr\\_t](#) \*svr)
- int **rtksvrstart** ([rtksvr\\_t](#) \*svr, int cycle, int bufsize, int \*strs, char \*\*paths, const int \*formats, int navsel, char \*\*cmds, char \*\*rcvopts, int nmeacycle, int nmeareq, const double \*nmeapos, [prcopt\\_t](#) \*prcopt, [solopt\\_t](#) \*solopt, [stream\\_t](#) \*moni)
- void **rtksvrstop** ([rtksvr\\_t](#) \*svr, char \*\*cmds)
- int **rtksvropenstr** ([rtksvr\\_t](#) \*svr, int index, int str, const char \*path, const [solopt\\_t](#) \*solopt)
- void **rtksvrclosestr** ([rtksvr\\_t](#) \*svr, int index)
- int **rtksvrostat** ([rtksvr\\_t](#) \*svr, int rcv, [gtime\\_t](#) \*time, int \*sat, double \*az, double \*el, int \*\*snr, int \*vsat)
- void **rtksvrsstat** ([rtksvr\\_t](#) \*svr, int \*sstat, char \*msg)

## Variables

- const [solopt\\_t](#) SOLOPT\_DEFAULT
- const [prcopt\\_t](#) PRCOPT\_DEFAULT

### 11.222.1 Detailed Description

rtk server functions

#### Authors

- 2007-2013, T. Takasu
- 2017, Javier Arribas
- 2017, Carles Fernandez

This is a derived work from RTKLIB <http://www.rtklib.com/> The original source code at <https://github.com/tomojitakasu/RTKLIB> is released under the BSD 2-clause license with an additional exclusive clause that does not apply here. This additional clause is reproduced below:

" The software package includes some companion executive binaries or shared libraries necessary to execute APs on Windows. These licenses succeed to the original ones of these software. "

Neither the executive binaries nor the shared libraries are required by, used or included in GNSS-SDR.  
 Copyright (C) 2007-2013, T. Takasu Copyright (C) 2017, Javier Arribas Copyright (C) 2017, Carles Fernandez All rights reserved.

SPDX-License-Identifier: BSD-2-Clause  
 Definition in file [rtklib\\_rtksvr.h](#).

## 11.222.2 Variable Documentation

### 11.222.2.1 PROCOPT\_DEFAULT

const [prcopt\\_t](#) PROCOPT\_DEFAULT

**Initial value:**

```
= {
    PMODE_SINGLE, 0, 2, SYS_GPS,
    15.0 * D2R, {{}}, {{}}, {{}},
    0, 1, 1, 1,
    5, 0, 10, 1,
    0, 0, 0, 0,
    1, 0, 0, 0, 0,
    0, 0,
    {100.0, 100.0, 100.0},
    {100.0, 0.003, 0.003, 0.0, 1.0},
    {30.0, 0.03, 0.3},
    {1e-4, 1e-3, 1e-4, 1e-1, 1e-2, 0.0},
    5E-12,
    {3.0, 0.9999, 0.25, 0.1, 0.05, 0, 0, 0},
    0.0, 0.0, 0.05,
    30.0, 30.0, 30.0,
    {}, {}, {},
    {"", ""},
    {}, {}, {},
    0, 0, 0, {"", ""}, {}, 0, {}, {}, {}, {}, {}, {}, {}, {}, {}, 0, {}, true}
}
```

Definition at line 49 of file [rtklib\\_rtksvr.h](#).

### 11.222.2.2 SOLOPT\_DEFAULT

const [solopt\\_t](#) SOLOPT\_DEFAULT

**Initial value:**

```
= {
    SOLF_LLH, TIMES_GPST, 1, 3,
    0, 1, 0, 0, 0, 0,
    0, 0, 0,
    {0.0, 0.0},
    " ", " ", 0
}
```

Definition at line 40 of file [rtklib\\_rtksvr.h](#).

## 11.223 rtklib\_rtksvr.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file rtklib_rtksvr.h
00003  * \brief rtk server functions
00004  * \authors <ul>
00005  *         <li> 2007-2013, T. Takasu
00006  *         <li> 2017, Javier Arribas
00007  *         <li> 2017, Carles Fernandez
00008  *         </ul>
00009  *
00010  * This is a derived work from RTKLIB http://www.rtklib.com/
00011  * The original source code at https://github.com/tomojitakasu/RTKLIB is
00012  * released under the BSD 2-clause license with an additional exclusive clause
00013  * that does not apply here. This additional clause is reproduced below:
00014  *
00015  * " The software package includes some companion executive binaries or shared
00016  * libraries necessary to execute APs on Windows. These licenses succeed to the
00017  * original ones of these software. "
00018  *
00019  * Neither the executive binaries nor the shared libraries are required by, used
00020  * or included in GNSS-SDR.
00021  *
00022  * -----
00023  * Copyright (C) 2007-2013, T. Takasu
00024  * Copyright (C) 2017, Javier Arribas
```

```

00025  * Copyright (C) 2017, Carles Fernandez
00026  * All rights reserved.
00027  *
00028  * SPDX-License-Identifier: BSD-2-Clause
00029  *
00030  *
00031  * -----
00032  */
00033
00034 #ifndef GNSS_SDR_RTKLIB_RKTSVR_H
00035 #define GNSS_SDR_RTKLIB_RKTSVR_H
00036
00037 #include "rtklib.h"
00038
00039
00040 const solopt_t SOLOPT_DEFAULT = {
00041     /* defaults solution output options */
00042     SOLF_LLH, TIMES_GPST, 1, 3, /* posf, times, timef, timeu */
00043     0, 1, 0, 0, 0, 0, /* degf, outthead, outopt, datum, height, geoid */
00044     0, 0, 0, /* solstatic, sstat, trace */
00045     {0.0, 0.0}, /* nmeaintv */
00046     " ", " ", 0 /* separator/program name */
00047 };
00048
00049 const prcopt_t PRCOPT_DEFAULT = { /* defaults processing options */
00050     PMODE_SINGLE, 0, 2, SYS_GPS, /* mode, soltype, nf, navsys */
00051     15.0 * D2R, {{}}, {{}}, {{}}, /* elmin, snrmask */
00052     0, 1, 1, 1, /* sateph, modear, glomodear, bdsmodear */
00053     5, 0, 10, 1, /* maxout, minlock, minfix, armaxiter */
00054     0, 0, 0, 0, /* estion, esttrop, dynamics, tidecorr */
00055     1, 0, 0, 0, 0, /* niter, codesmooth, intpref, sbascorr, sbassatsel */
00056     0, 0, /* rovpos, refpos */
00057     {100.0, 100.0, 100.0}, /* eratio */
00058     {100.0, 0.003, 0.003, 0.0, 1.0}, /* err */
00059     {30.0, 0.03, 0.3}, /* std */
00060     {1e-4, 1e-3, 1e-4, 1e-1, 1e-2, 0.0}, /* prn */
00061     5E-12, /* sclkstab */
00062     {3.0, 0.9999, 0.25, 0.1, 0.05, 0, 0, 0}, /* thresar */
00063     0.0, 0.0, 0.05, /* elmaskar, almaskhold, thresslip */
00064     30.0, 30.0, 30.0, /* maxtdif, maxinno, maxgdop */
00065     {{}}, {{}}, {{}}, /* baseline, ru, rb */
00066     {"", ""}, /* anttype */
00067     {{}}, {{}}, {{}}, /* antdel, pcv, exsats */
00068     0, 0, 0, {"", ""}, {{}}, 0, {{}}, {{}}, {{}}, {{}}, {{}}, {{}}, {{}}, 0, {{}}, true;
00069
00070
00071 void writesolhead(stream_t *stream, const solopt_t *solopt);
00072
00073 void saveoutbuf(rtksvr_t *svr, unsigned char *buff, int n, int index);
00074
00075 void writesol(rtksvr_t *svr, int index);
00076
00077 void updatenav(nav_t *nav);
00078
00079 void updatefcfn(rtksvr_t *svr);
00080
00081 void updatesvr(rtksvr_t *svr, int ret, obs_t *obs, nav_t *nav, int sat,
00082     sbmsg_t *sbmsg, int index, int iobs);
00083
00084 int decoderaw(rtksvr_t *svr, int index);
00085
00086 void decodefile(rtksvr_t *svr, int index);
00087
00088 void *rtksvrthread(void *arg);
00089
00090 int rtksvrinit(rtksvr_t *svr);
00091
00092 void rtksvrfree(rtksvr_t *svr);
00093
00094 void rtksvrlock(rtksvr_t *svr);
00095
00096 void rtksvrunlock(rtksvr_t *svr);
00097
00098 int rtksvrstart(rtksvr_t *svr, int cycle, int buffsize, int *strs,
00099     char **paths, const int *formats, int navsel, char **cmds,
00100     char **rcvopts, int nmeacycle, int nmeareq,
00101     const double *nmeapos, prcopt_t *prcopt,
00102     solopt_t *solopt, stream_t *moni);
00103
00104 void rtksvrstop(rtksvr_t *svr, char **cmds);
00105
00106 int rtksvropenstr(rtksvr_t *svr, int index, int str, const char *path,
00107     const solopt_t *solopt);
00108
00109 void rtksvrclosestr(rtksvr_t *svr, int index);
00110
00111 int rtksvrostat(rtksvr_t *svr, int rcv, gtime_t *time, int *sat,

```

```

00112     double *az, double *el, int **snr, int *vsat);
00113
00114 void rtksvrsstat(rtksvr_t *svr, int *sstat, char *msg);
00115
00116
00117 #endif

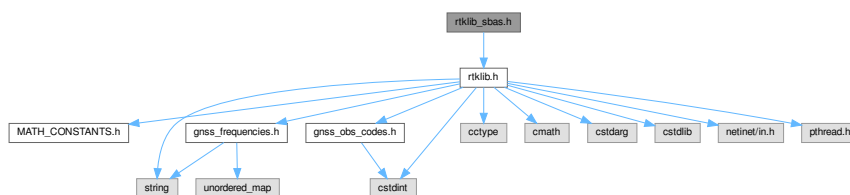
```

## 11.224 rtklib\_sbas.h File Reference

sbas functions

```
#include "rtklib.h"
```

Include dependency graph for rtklib\_sbas.h:



### Functions

- char \* **getfield** (char \*p, int pos)
- double **varfcorr** (int udre)
- double **varicorr** (int give)
- double **degfcorr** (int ai)
- int **decode\_sbstype1** (const *sbsmsg\_t* \*msg, *sbssat\_t* \*sbssat)
- int **decode\_sbstype2** (const *sbsmsg\_t* \*msg, *sbssat\_t* \*sbssat)
- int **decode\_sbstype6** (const *sbsmsg\_t* \*msg, *sbssat\_t* \*sbssat)
- int **decode\_sbstype7** (const *sbsmsg\_t* \*msg, *sbssat\_t* \*sbssat)
- int **decode\_sbstype9** (const *sbsmsg\_t* \*msg, *nav\_t* \*nav)
- int **decode\_sbstype18** (const *sbsmsg\_t* \*msg, *sbsion\_t* \*sbsion)
- int **decode\_longcorr0** (const *sbsmsg\_t* \*msg, int p, *sbssat\_t* \*sbssat)
- int **decode\_longcorr1** (const *sbsmsg\_t* \*msg, int p, *sbssat\_t* \*sbssat)
- int **decode\_longcorrh** (const *sbsmsg\_t* \*msg, int p, *sbssat\_t* \*sbssat)
- int **decode\_sbstype24** (const *sbsmsg\_t* \*msg, *sbssat\_t* \*sbssat)
- int **decode\_sbstype25** (const *sbsmsg\_t* \*msg, *sbssat\_t* \*sbssat)
- int **decode\_sbstype26** (const *sbsmsg\_t* \*msg, *sbsion\_t* \*sbsion)
- int **sbsupdatecorr** (const *sbsmsg\_t* \*msg, *nav\_t* \*nav)
- void **readmsgs** (const char \*file, int sel, *gtime\_t* ts, *gtime\_t* te, *sbs\_t* \*sbs)
- int **cmpmsgs** (const void \*p1, const void \*p2)
- int **sbsreadmsgt** (const char \*file, int sel, *gtime\_t* ts, *gtime\_t* te, *sbs\_t* \*sbs)
- int **sbsreadmsg** (const char \*file, int sel, *sbs\_t* \*sbs)
- void **sbsoutmsg** (FILE \*fp, *sbsmsg\_t* \*sbsmsg)
- void **searchigp** (*gtime\_t* time, const double \*pos, const *sbsion\_t* \*ion, const *sbsigp\_t* \*\*igp, double \*x, double \*y)
- int **sbsioncorr** (*gtime\_t* time, const *nav\_t* \*nav, const double \*pos, const double \*azel, double \*delay, double \*var)
- void **getmet** (double lat, double \*met)
- double **sbstropcorr** (*gtime\_t* time, const double \*pos, const double \*azel, double \*var)
- int **sbslongcorr** (*gtime\_t* time, int sat, const *sbssat\_t* \*sbssat, double \*drs, double \*ddts)
- int **sbsfastcorr** (*gtime\_t* time, int sat, const *sbssat\_t* \*sbssat, double \*prc, double \*var)
- int **sbssatcorr** (*gtime\_t* time, int sat, const *nav\_t* \*nav, double \*rs, double \*dts, double \*var)
- int **sbsdecodemsg** (*gtime\_t* time, int prn, const unsigned int \*words, *sbsmsg\_t* \*sbsmsg)

## Variables

- const int [WEEKOFFSET](#) = 1024
- const [sbsigband\\_t](#) [IGPBAND1](#) [9][8]
- const [sbsigband\\_t](#) [IGPBAND2](#) [2][5]

## 11.224.1 Detailed Description

sbas functions

Authors

- 2007-2013, T. Takasu
- 2017, Javier Arribas
- 2017, Carles Fernandez

This is a derived work from RTKLIB <http://www.rtklib.com/> The original source code at <https://github.com/tomojitakasu/RTKLIB> is released under the BSD 2-clause license with an additional exclusive clause that does not apply here. This additional clause is reproduced below:

" The software package includes some companion executive binaries or shared libraries necessary to execute APs on Windows. These licenses succeed to the original ones of these software. "

Neither the executive binaries nor the shared libraries are required by, used or included in GNSS-SDR.

Copyright (C) 2007-2013, T. Takasu Copyright (C) 2017, Javier Arribas Copyright (C) 2017, Carles Fernandez All rights reserved.

SPDX-License-Identifier: BSD-2-Clause

References : [1] RTCA/DO-229C, Minimum operational performance standards for global positioning system/wide area augmentation system airborne equipment, RTCA inc, November 28, 2001 [2] IS-QZSS v.1.1, Quasi-Zenith Satellite System Navigation Service Interface Specification for QZSS, Japan Aerospace Exploration Agency, July 31, 2009

Definition in file [rtklib\\_sbas.h](#).

## 11.224.2 Variable Documentation

### 11.224.2.1 IGPBAND1

```
const sbsigband_t IGPBAND1[9][8]
```

**Initial value:**

```
= {
    {{-180, X1, 1, 28}, {-175, X2, 29, 51}, {-170, X3, 52, 78}, {-165, X2, 79, 101},
     {-160, X3, 102, 128}, {-155, X2, 129, 151}, {-150, X3, 152, 178}, {-145, X2, 179, 201}},
    {{-140, X4, 1, 28}, {-135, X2, 29, 51}, {-130, X3, 52, 78}, {-125, X2, 79, 101},
     {-120, X3, 102, 128}, {-115, X2, 129, 151}, {-110, X3, 152, 178}, {-105, X2, 179, 201}},
    {{-100, X3, 1, 27}, {-95, X2, 28, 50}, {-90, X1, 51, 78}, {-85, X2, 79, 101},
     {-80, X3, 102, 128}, {-75, X2, 129, 151}, {-70, X3, 152, 178}, {-65, X2, 179, 201}},
    {{-60, X3, 1, 27}, {-55, X2, 28, 50}, {-50, X4, 51, 78}, {-45, X2, 79, 101},
     {-40, X3, 102, 128}, {-35, X2, 129, 151}, {-30, X3, 152, 178}, {-25, X2, 179, 201}},
    {{-20, X3, 1, 27}, {-15, X2, 28, 50}, {-10, X3, 51, 77}, {-5, X2, 78, 100},
     {0, X1, 101, 128}, {5, X2, 129, 151}, {10, X3, 152, 178}, {15, X2, 179, 201}},
    {{20, X3, 1, 27}, {25, X2, 28, 50}, {30, X3, 51, 77}, {35, X2, 78, 100},
     {40, X4, 101, 128}, {45, X2, 129, 151}, {50, X3, 152, 178}, {55, X2, 179, 201}},
    {{60, X3, 1, 27}, {65, X2, 28, 50}, {70, X3, 51, 77}, {75, X2, 78, 100},
     {80, X3, 101, 127}, {85, X2, 128, 150}, {90, X1, 151, 178}, {95, X2, 179, 201}},
    {{100, X3, 1, 27}, {105, X2, 28, 50}, {110, X3, 51, 77}, {115, X2, 78, 100},
     {120, X3, 101, 127}, {125, X2, 128, 150}, {130, X4, 151, 178}, {135, X2, 179, 201}},
    {{140, X3, 1, 27}, {145, X2, 28, 50}, {150, X3, 51, 77}, {155, X2, 78, 100},
     {160, X3, 101, 127}, {165, X2, 128, 150}, {170, X3, 151, 177}, {175, X2, 178, 200}}}
```

Definition at line 73 of file [rtklib\\_sbas.h](#).

### 11.224.2.2 IGPBAND2

```
const sbsigband_t IGPBAND2[2][5]
```

**Initial value:**

```
= {
    {{60, X5, 1, 72}, {65, X6, 73, 108}, {70, X6, 109, 144}, {75, X6, 145, 180},
     {85, X7, 181, 192}},
    {{-60, X5, 1, 72}, {-65, X6, 73, 108}, {-70, X6, 109, 144}, {-75, X6, 145, 180},
     {-85, X8, 181, 192}}}
```

Definition at line 92 of file [rtklib\\_sbas.h](#).

### 11.224.2.3 WEEKOFFSET

const int WEEKOFFSET = 1024

Definition at line 49 of file [rtklib\\_sbass.h](#).

## 11.225 rtklib\_sbass.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file rtklib_sbass.h
00003  * \brief sbass functions
00004  * \authors <ul>
00005  *         <li> 2007-2013, T. Takasu
00006  *         <li> 2017, Javier Arribas
00007  *         <li> 2017, Carles Fernandez
00008  *         </ul>
00009  *
00010  * This is a derived work from RTKLIB http://www.rtklib.com/
00011  * The original source code at https://github.com/tomokitakasu/RTKLIB is
00012  * released under the BSD 2-clause license with an additional exclusive clause
00013  * that does not apply here. This additional clause is reproduced below:
00014  *
00015  * " The software package includes some companion executive binaries or shared
00016  * libraries necessary to execute APs on Windows. These licenses succeed to the
00017  * original ones of these software. "
00018  *
00019  * Neither the executive binaries nor the shared libraries are required by, used
00020  * or included in GNSS-SDR.
00021  *
00022  * -----
00023  * Copyright (C) 2007-2013, T. Takasu
00024  * Copyright (C) 2017, Javier Arribas
00025  * Copyright (C) 2017, Carles Fernandez
00026  * All rights reserved.
00027  *
00028  * SPDX-License-Identifier: BSD-2-Clause
00029  *
00030  *
00031  * References :
00032  * [1] RTCA/DO-229C, Minimum operational performance standards for global
00033  * positioning system/wide area augmentation system airborne equipment,
00034  * RTCA inc, November 28, 2001
00035  * [2] IS-QZSS v.1.1, Quasi-Zenith Satellite System Navigation Service
00036  * Interface Specification for QZSS, Japan Aerospace Exploration Agency,
00037  * July 31, 2009
00038  *
00039  * -----
00040  */
00041
00042 #ifndef GNSS_SDR_RTKLIB_SBASS_H
00043 #define GNSS_SDR_RTKLIB_SBASS_H
00044
00045 #include "rtklib.h"
00046
00047 /* constants -----*/
00048
00049 const int WEEKOFFSET = 1024; /* gps week offset for NovAtel OEM-3 */
00050
00051 /* sbass igp definition -----*/
00052 static const short
00053 X1[] = {-75, -65, -55, -50, -45, -40, -35, -30, -25, -20, -15, -10, -5, 0, 5, 10, 15, 20,
00054         25, 30, 35, 40, 45, 50, 55, 65, 75, 85},
00055 X2[] = {-55, -50, -45, -40, -35, -30, -25, -20, -15, -10, -5, 0, 5, 10, 15, 20, 25, 30,
00056         35, 40, 45, 50, 55},
00057 X3[] = {-75, -65, -55, -50, -45, -40, -35, -30, -25, -20, -15, -10, -5, 0, 5, 10, 15, 20,
00058         25, 30, 35, 40, 45, 50, 55, 65, 75},
00059 X4[] = {-85, -75, -65, -55, -50, -45, -40, -35, -30, -25, -20, -15, -10, -5, 0, 5, 10, 15,
00060         20, 25, 30, 35, 40, 45, 50, 55, 65, 75},
00061 X5[] = {-180, -175, -170, -165, -160, -155, -150, -145, -140, -135, -130, -125, -120, -115,
00062         -110, -105, -100, -95, -90, -85, -80, -75, -70, -65, -60, -55, -50, -45,
00063         -40, -35, -30, -25, -20, -15, -10, -5, 0, 5, 10, 15, 20, 25,
00064         30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95,
00065         100, 105, 110, 115, 120, 125, 130, 135, 140, 145, 150, 155, 160, 165,
00066         170, 175},
00067 X6[] = {-180, -170, -160, -150, -140, -130, -120, -110, -100, -90, -80, -70, -60, -50,
00068         -40, -30, -20, -10, 0, 10, 20, 30, 40, 50, 60, 70, 80, 90,
00069         100, 110, 120, 130, 140, 150, 160, 170},
00070 X7[] = {-180, -150, -120, -90, -60, -30, 0, 30, 60, 90, 120, 150},
00071 X8[] = {-170, -140, -110, -80, -50, -20, 10, 40, 70, 100, 130, 160};
00072
00073 const sbasigpband_t IGPBAND1[9][8] = { /* band 0-8 */
00074     {-180, X1, 1, 28}, {-175, X2, 29, 51}, {-170, X3, 52, 78}, {-165, X2, 79, 101},
00075     {-160, X3, 102, 128}, {-155, X2, 129, 151}, {-150, X3, 152, 178}, {-145, X2, 179, 201}},
```



```

00076     {-140, X4, 1, 28}, {-135, X2, 29, 51}, {-130, X3, 52, 78}, {-125, X2, 79, 101},
00077     {-120, X3, 102, 128}, {-115, X2, 129, 151}, {-110, X3, 152, 178}, {-105, X2, 179, 201}},
00078     {-100, X3, 1, 27}, {-95, X2, 28, 50}, {-90, X1, 51, 78}, {-85, X2, 79, 101},
00079     {-80, X3, 102, 128}, {-75, X2, 129, 151}, {-70, X3, 152, 178}, {-65, X2, 179, 201}},
00080     {-60, X3, 1, 27}, {-55, X2, 28, 50}, {-50, X4, 51, 78}, {-45, X2, 79, 101},
00081     {-40, X3, 102, 128}, {-35, X2, 129, 151}, {-30, X3, 152, 178}, {-25, X2, 179, 201}},
00082     {-20, X3, 1, 27}, {-15, X2, 28, 50}, {-10, X3, 51, 77}, {-5, X2, 78, 100},
00083     {0, X1, 101, 128}, {5, X2, 129, 151}, {10, X3, 152, 178}, {15, X2, 179, 201}},
00084     {20, X3, 1, 27}, {25, X2, 28, 50}, {30, X3, 51, 77}, {35, X2, 78, 100},
00085     {40, X4, 101, 128}, {45, X2, 129, 151}, {50, X3, 152, 178}, {55, X2, 179, 201}},
00086     {60, X3, 1, 27}, {65, X2, 28, 50}, {70, X3, 51, 77}, {75, X2, 78, 100},
00087     {80, X3, 101, 127}, {85, X2, 128, 150}, {90, X1, 151, 178}, {95, X2, 179, 201}},
00088     {100, X3, 1, 27}, {105, X2, 28, 50}, {110, X3, 51, 77}, {115, X2, 78, 100},
00089     {120, X3, 101, 127}, {125, X2, 128, 150}, {130, X4, 151, 178}, {135, X2, 179, 201}},
00090     {140, X3, 1, 27}, {145, X2, 28, 50}, {150, X3, 51, 77}, {155, X2, 78, 100},
00091     {160, X3, 101, 127}, {165, X2, 128, 150}, {170, X3, 151, 177}, {175, X2, 178, 200}}};
00092 const sbssigpbandt IGPBAND2[2][5] = { /* band 9-10 */
00093     {60, X5, 1, 72}, {65, X6, 73, 108}, {70, X6, 109, 144}, {75, X6, 145, 180},
00094     {85, X7, 181, 192}},
00095     {{-60, X5, 1, 72}, {-65, X6, 73, 108}, {-70, X6, 109, 144}, {-75, X6, 145, 180},
00096     {-85, X8, 181, 192}}};
00097
00098
00099 char *getfield(char *p, int pos);
00100 double varfcorr(int udre);
00101 double varicorr(int give);
00102 double degfcorr(int ai);
00103
00104 int decode_sbstype1(const sbmsg_t *msg, sbssat_t *sbssat);
00105 int decode_sbstype2(const sbmsg_t *msg, sbssat_t *sbssat);
00106 int decode_sbstype6(const sbmsg_t *msg, sbssat_t *sbssat);
00107 int decode_sbstype7(const sbmsg_t *msg, sbssat_t *sbssat);
00108 int decode_sbstype9(const sbmsg_t *msg, nav_t *nav);
00109 int decode_sbstype18(const sbmsg_t *msg, sbssion_t *sbssion);
00110 int decode_longcorr0(const sbmsg_t *msg, int p, sbssat_t *sbssat);
00111 int decode_longcorr1(const sbmsg_t *msg, int p, sbssat_t *sbssat);
00112 int decode_longcorrh(const sbmsg_t *msg, int p, sbssat_t *sbssat);
00113 int decode_sbstype24(const sbmsg_t *msg, sbssat_t *sbssat);
00114 int decode_sbstype25(const sbmsg_t *msg, sbssat_t *sbssat);
00115 int decode_sbstype26(const sbmsg_t *msg, sbssion_t *sbssion);
00116
00117 int sbssupdatercorr(const sbmsg_t *msg, nav_t *nav);
00118 void readmsgs(const char *file, int sel, gtime_t ts, gtime_t te, sbs_t *sbs);
00119 int cmpmsgs(const void *p1, const void *p2);
00120 int sbssreadmsg(const char *file, int sel, gtime_t ts, gtime_t te,
00121     sbs_t *sbs);
00122 int sbssreadmsg(const char *file, int sel, sbs_t *sbs);
00123 void sbssoutmsg(FILE *fp, sbmsg_t *sbmsg);
00124 void searchigp(gtime_t time, const double *pos, const sbssion_t *ion,
00125     const sbssigp_t **igp, double *x, double *y);
00126 int sbssioncorr(gtime_t time, const nav_t *nav, const double *pos,
00127     const double *azel, double *delay, double *var);
00128
00129 void getmet(double lat, double *met);
00130 double sbstropcorr(gtime_t time, const double *pos, const double *azel,
00131     double *var);
00132 int sbsslongcorr(gtime_t time, int sat, const sbssat_t *sbssat,
00133     double *drs, double *ddts);
00134 int sbssfastcorr(gtime_t time, int sat, const sbssat_t *sbssat,
00135     double *prc, double *var);
00136
00137 int sbssatcorr(gtime_t time, int sat, const nav_t *nav, double *rs,
00138     double *dts, double *var);
00139 int sbssdecodemsg(gtime_t time, int prn, const unsigned int *words,
00140     sbmsg_t *sbmsg);
00141
00142
00143 #endif // GNSS_SDR_RTKLIB_SBAS_H

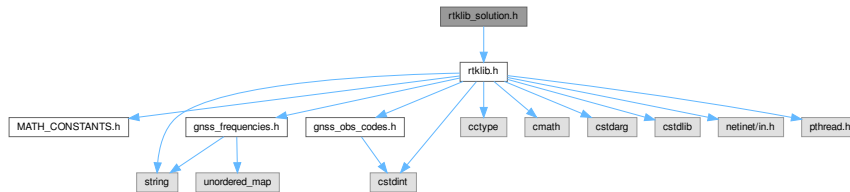
```

## 11.226 rtklib\_solution.h File Reference

solution functions headers

```
#include "rtklib.h"
```

Include dependency graph for rtklib\_solution.h:



## Macros

- #define **COMMENTH** "%" /\* comment line indicator for solution \*/
- #define **MSG\_DISCONN** "\$\_DISCONNECT\r\n" /\* disconnect message \*/

## Functions

- const char \* **opt2sep** (const [solopt\\_t](#) \*opt)
- int **tonum** (char \*buff, const char \*sep, double \*v)
- double **sqvar** (double covar)
- double **dmm2deg** (double dmm)
- void **septime** (double t, double \*t1, double \*t2, double \*t3)
- void **soltocov** (const [sol\\_t](#) \*sol, double \*P)
- void **covtosol** (const double \*P, [sol\\_t](#) \*sol)
- int **decode\_nmearmc** (char \*\*val, int n, [sol\\_t](#) \*sol)
- int **decode\_nmeagga** (char \*\*val, int n, [sol\\_t](#) \*sol)
- int **decode\_nmea** (char \*buff, [sol\\_t](#) \*sol)
- char \* **decode\_soltime** (char \*buff, const [solopt\\_t](#) \*opt, [gtime\\_t](#) \*time)
- int **decode\_solxyz** (char \*buff, const [solopt\\_t](#) \*opt, [sol\\_t](#) \*sol)
- int **decode\_solllh** (char \*buff, const [solopt\\_t](#) \*opt, [sol\\_t](#) \*sol)
- int **decode\_solenu** (char \*buff, const [solopt\\_t](#) \*opt, [sol\\_t](#) \*sol)
- int **decode\_solgsi** (char \*buff, const [solopt\\_t](#) \*opt, [sol\\_t](#) \*sol)
- int **decode\_solpos** (char \*buff, const [solopt\\_t](#) \*opt, [sol\\_t](#) \*sol)
- void **decode\_refpos** (char \*buff, const [solopt\\_t](#) \*opt, double \*rb)
- int **decode\_sol** (char \*buff, const [solopt\\_t](#) \*opt, [sol\\_t](#) \*sol, double \*rb)
- void **decode\_solopt** (char \*buff, [solopt\\_t](#) \*opt)
- void **readsolopt** (FILE \*fp, [solopt\\_t](#) \*opt)
- int **inputsol** (unsigned char data, [gtime\\_t](#) ts, [gtime\\_t](#) te, double tint, int qflag, const [solopt\\_t](#) \*opt, [solbuf\\_t](#) \*solbuf)
- int **readsoldata** (FILE \*fp, [gtime\\_t](#) ts, [gtime\\_t](#) te, double tint, int qflag, const [solopt\\_t](#) \*opt, [solbuf\\_t](#) \*solbuf)
- int **cmpsol** (const void \*p1, const void \*p2)
- int **sort\_solbuf** ([solbuf\\_t](#) \*solbuf)
- int **readsolt** (char \*files[], int nfile, [gtime\\_t](#) ts, [gtime\\_t](#) te, double tint, int qflag, [solbuf\\_t](#) \*solbuf)
- int **readsol** (char \*files[], int nfile, [solbuf\\_t](#) \*sol)
- int **addsol** ([solbuf\\_t](#) \*solbuf, const [sol\\_t](#) \*sol)
- [sol\\_t](#) \* **getsol** ([solbuf\\_t](#) \*solbuf, int index)
- void **initsolbuf** ([solbuf\\_t](#) \*solbuf, int cyclic, int nmax)
- void **freesolbuf** ([solbuf\\_t](#) \*solbuf)
- void **freesolstatbuf** ([solstatbuf\\_t](#) \*solstatbuf)
- int **cmpsolstat** (const void \*p1, const void \*p2)
- int **sort\_solstat** ([solstatbuf\\_t](#) \*statbuf)
- int **decode\_solstat** (char \*buff, [solstat\\_t](#) \*stat)
- void **addsolstat** ([solstatbuf\\_t](#) \*statbuf, const [solstat\\_t](#) \*stat)

- int **readsolstatdata** (FILE \*fp, [gtime\\_t](#) ts, [gtime\\_t](#) te, double tint, [solstatbuf\\_t](#) \*statbuf)
- int **readsolstatt** (char \*files[], int nfile, [gtime\\_t](#) ts, [gtime\\_t](#) te, double tint, [solstatbuf\\_t](#) \*statbuf)
- int **readsolstat** (char \*files[], int nfile, [solstatbuf\\_t](#) \*statbuf)
- int **outecef** (unsigned char \*buff, const char \*s, const [sol\\_t](#) \*sol, const [solopt\\_t](#) \*opt)
- int **outpos** (unsigned char \*buff, const char \*s, const [sol\\_t](#) \*sol, const [solopt\\_t](#) \*opt)
- int **outenu** (unsigned char \*buff, const char \*s, const [sol\\_t](#) \*sol, const double \*rb, const [solopt\\_t](#) \*opt)
- int **outnmea\_rmc** (unsigned char \*buff, const [sol\\_t](#) \*sol)
- int **outnmea\_gga** (unsigned char \*buff, const [sol\\_t](#) \*sol)
- int **outnmea\_gsa** (unsigned char \*buff, const [sol\\_t](#) \*sol, const [ssat\\_t](#) \*ssat)
- int **outnmea\_gsv** (unsigned char \*buff, const [sol\\_t](#) \*sol, const [ssat\\_t](#) \*ssat)
- int **outprcopts** (unsigned char \*buff, const [prcopt\\_t](#) \*opt)
- int **outsolheads** (unsigned char \*buff, const [solopt\\_t](#) \*opt)
- int **outsols** (unsigned char \*buff, const [sol\\_t](#) \*sol, const double \*rb, const [solopt\\_t](#) \*opt)
- int **outsollexs** (unsigned char \*buff, const [sol\\_t](#) \*sol, const [ssat\\_t](#) \*ssat, const [solopt\\_t](#) \*opt)
- void **outprcopt** (FILE \*fp, const [prcopt\\_t](#) \*opt)
- void **outsolhead** (FILE \*fp, const [solopt\\_t](#) \*opt)
- void **outsol** (FILE \*fp, const [sol\\_t](#) \*sol, const double \*rb, const [solopt\\_t](#) \*opt)
- void **outsollex** (FILE \*fp, const [sol\\_t](#) \*sol, const [ssat\\_t](#) \*ssat, const [solopt\\_t](#) \*opt)

### 11.226.1 Detailed Description

solution functions headers

#### Authors

- 2007-2013, T. Takasu
- 2017, Javier Arribas
- 2017, Carles Fernandez

This is a derived work from RTKLIB <http://www.rtklib.com/> The original source code at <https://github.com/tomojitakasu/RTKLIB> is released under the BSD 2-clause license with an additional exclusive clause that does not apply here. This additional clause is reproduced below:

" The software package includes some companion executive binaries or shared libraries necessary to execute APs on Windows. These licenses succeed to the original ones of these software. "

Neither the executive binaries nor the shared libraries are required by, used or included in GNSS-SDR.

Copyright (C) 2007-2013, T. Takasu Copyright (C) 2017, Javier Arribas Copyright (C) 2017, Carles Fernandez All rights reserved.

#### 11.226.1.1 SPDX-License-Identifier: BSD-2-Clause

Definition in file [rtklib\\_solution.h](#).

## 11.227 rtklib\_solution.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file rtklib_solution.h
00003  * \brief solution functions headers
00004  * \authors <ul>
00005  *         <li> 2007-2013, T. Takasu
00006  *         <li> 2017, Javier Arribas
00007  *         <li> 2017, Carles Fernandez
00008  *         </ul>
00009  *
00010  * This is a derived work from RTKLIB http://www.rtklib.com/
00011  * The original source code at https://github.com/tomojitakasu/RTKLIB is
00012  * released under the BSD 2-clause license with an additional exclusive clause
00013  * that does not apply here. This additional clause is reproduced below:
00014  *
00015  * " The software package includes some companion executive binaries or shared
00016  * libraries necessary to execute APs on Windows. These licenses succeed to the
00017  * original ones of these software. "
00018  *
```

```

00019  * Neither the executive binaries nor the shared libraries are required by, used
00020  * or included in GNSS-SDR.
00021  *
00022  * -----
00023  * Copyright (C) 2007-2013, T. Takasu
00024  * Copyright (C) 2017, Javier Arribas
00025  * Copyright (C) 2017, Carles Fernandez
00026  * All rights reserved.
00027  *
00028  * SPDX-License-Identifier: BSD-2-Clause
00029  *-----
00030  */
00031
00032 #ifndef GNSS_SDR_RTKLIB_SOLUTION_H
00033 #define GNSS_SDR_RTKLIB_SOLUTION_H
00034
00035 #include "rtklib.h"
00036
00037 /** \addtogroup PVT
00038  * \{ */
00039 /** \addtogroup RTKLIB_Library
00040  * \{ */
00041
00042
00043 #define COMMENTH "%" /* comment line indicator for solution */
00044 #define MSG_DISCONNECT "$_DISCONNECT\r\n" /* disconnect message */
00045
00046 const char *opt2sep(const solopt_t *opt);
00047
00048 int tonum(char *buff, const char *sep, double *v);
00049
00050 double sqvar(double covar);
00051
00052 double dmm2deg(double dmm);
00053
00054 void septime(double t, double *t1, double *t2, double *t3);
00055
00056 void soltocov(const sol_t *sol, double *P);
00057
00058 void covtosol(const double *P, sol_t *sol);
00059
00060 int decode_nmearmc(char **val, int n, sol_t *sol);
00061
00062 int decode_nmeagga(char **val, int n, sol_t *sol);
00063
00064 int decode_nmea(char *buff, sol_t *sol);
00065
00066 char *decode_soltime(char *buff, const solopt_t *opt, gtime_t *time);
00067
00068 int decode_solxyz(char *buff, const solopt_t *opt, sol_t *sol);
00069
00070 int decode_solllh(char *buff, const solopt_t *opt, sol_t *sol);
00071
00072 int decode_solenu(char *buff, const solopt_t *opt, sol_t *sol);
00073
00074 int decode_solgsi(char *buff, const solopt_t *opt, sol_t *sol);
00075
00076 int decode_solpos(char *buff, const solopt_t *opt, sol_t *sol);
00077
00078 void decode_refpos(char *buff, const solopt_t *opt, double *rb);
00079
00080 int decode_sol(char *buff, const solopt_t *opt, sol_t *sol, double *rb);
00081
00082 void decode_solopt(char *buff, solopt_t *opt);
00083
00084 void readsolopt(FILE *fp, solopt_t *opt);
00085
00086 int inputsol(unsigned char data, gtime_t ts, gtime_t te, double tint,
00087             int qflag, const solopt_t *opt, solbuf_t *solbuf);
00088
00089 int readsoldata(FILE *fp, gtime_t ts, gtime_t te, double tint, int qflag,
00090               const solopt_t *opt, solbuf_t *solbuf);
00091
00092 int cmpsol(const void *p1, const void *p2);
00093
00094 int sort_solbuf(solbuf_t *solbuf);
00095
00096 int readsolt(char *files[], int nfile, gtime_t ts, gtime_t te,
00097             double tint, int qflag, solbuf_t *solbuf);
00098
00099 int readsol(char *files[], int nfile, solbuf_t *sol);
00100
00101 int addsol(solbuf_t *solbuf, const sol_t *sol);
00102
00103 sol_t *getsol(solbuf_t *solbuf, int index);
00104
00105 void initsolbuf(solbuf_t *solbuf, int cyclic, int nmax);

```

```

00106
00107 void freesolbuf(solbuf_t *solbuf);
00108
00109 void freesolstatbuf(solstatbuf_t *solstatbuf);
00110
00111 int cmpsolstat(const void *p1, const void *p2);
00112
00113 int sort_solstat(solstatbuf_t *statbuf);
00114
00115 int decode_solstat(char *buff, solstat_t *stat);
00116
00117 void addsolstat(solstatbuf_t *statbuf, const solstat_t *stat);
00118
00119 int readsolstatdata(FILE *fp, gtime_t ts, gtime_t te, double tint,
00120     solstatbuf_t *statbuf);
00121
00122 int readsolstatt(char *files[], int nfile, gtime_t ts, gtime_t te,
00123     double tint, solstatbuf_t *statbuf);
00124
00125 int readsolstat(char *files[], int nfile, solstatbuf_t *statbuf);
00126
00127 int outecef(unsigned char *buff, const char *s, const sol_t *sol,
00128     const solopt_t *opt);
00129
00130 int outpos(unsigned char *buff, const char *s, const sol_t *sol, const solopt_t *opt);
00131
00132 int outenu(unsigned char *buff, const char *s, const sol_t *sol,
00133     const double *rb, const solopt_t *opt);
00134
00135 int outnmea_rmc(unsigned char *buff, const sol_t *sol);
00136
00137 int outnmea_gga(unsigned char *buff, const sol_t *sol);
00138
00139 int outnmea_gsa(unsigned char *buff, const sol_t *sol,
00140     const ssat_t *ssat);
00141
00142 int outnmea_gsv(unsigned char *buff, const sol_t *sol,
00143     const ssat_t *ssat);
00144
00145 int outprcopts(unsigned char *buff, const prcopt_t *opt);
00146
00147 int outsolheads(unsigned char *buff, const solopt_t *opt);
00148
00149 int outsols(unsigned char *buff, const sol_t *sol, const double *rb,
00150     const solopt_t *opt);
00151
00152 int outsolexs(unsigned char *buff, const sol_t *sol, const ssat_t *ssat,
00153     const solopt_t *opt);
00154
00155 void outprcopt(FILE *fp, const prcopt_t *opt);
00156
00157 void outsolhead(FILE *fp, const solopt_t *opt);
00158
00159 void outsol(FILE *fp, const sol_t *sol, const double *rb,
00160     const solopt_t *opt);
00161
00162 void outsolex(FILE *fp, const sol_t *sol, const ssat_t *ssat,
00163     const solopt_t *opt);
00164
00165
00166 /** \} */
00167 /** \} */
00168 #endif // GNSS_SDR_RTKLIB_SOLUTION_H

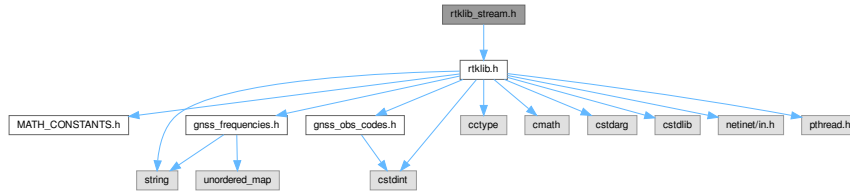
```

## 11.228 rtklib\_stream.h File Reference

streaming functions

```
#include "rtklib.h"
```

Include dependency graph for rtklib\_stream.h:



## Macros

- `#define TINTACT 200` /\* period for stream active (ms) \*/
- `#define SERIBUFFSIZE 4096` /\* serial buffer size (bytes) \*/
- `#define TIMETAGH_LEN 64` /\* time tag file header length \*/
- `#define MAXCLI 32` /\* max client connection for tcp svr \*/
- `#define MAXSTATMSG 32` /\* max length of status message \*/
- `#define VER_RTKLIB "2.4.2"`
- `#define NTRIP_AGENT "RTKLIB/" VER_RTKLIB`
- `#define NTRIP_CLI_PORT 2101` /\* default ntrip-client connection port \*/
- `#define NTRIP_SVR_PORT 80` /\* default ntrip-server connection port \*/
- `#define NTRIP_MAXRSP 32768` /\* max size of ntrip response \*/
- `#define NTRIP_MAXSTR 256` /\* max length of mountpoint string \*/
- `#define NTRIP_RSP_OK_CLI "ICY 200 OK\r\n"` /\* ntrip response: client \*/
- `#define NTRIP_RSP_OK_SVR "OK\r\n"` /\* ntrip response: server \*/
- `#define NTRIP_RSP_SRCTBL "SOURCETABLE 200 OK\r\n"` /\* ntrip response: source table \*/
- `#define NTRIP_RSP_TBLEND "ENDSOURCETABLE"`
- `#define NTRIP_RSP_HTTP "HTTP/"` /\* ntrip response: http \*/
- `#define NTRIP_RSP_ERROR "ERROR"` /\* ntrip response: error \*/
- `#define FTP_CMD "wget"` /\* ftp/http command \*/
- `#define FTP_TIMEOUT 30` /\* ftp/http timeout (s) \*/

## Functions

- `serial_t * openserial` (const char \*path, int mode, char \*msg)
- void `closeserial` (serial\_t \*serial)
- int `readserial` (serial\_t \*serial, unsigned char \*buff, int n, char \*msg)
- int `writeserial` (serial\_t \*serial, unsigned char \*buff, int n, char \*msg)
- int `stateserial` (serial\_t \*serial)
- int `openfile_` (file\_t \*file, gtime\_t time, char \*msg)
- void `closefile_` (file\_t \*file)
- `file_t * openfile` (const char \*path, int mode, char \*msg)
- void `closefile` (file\_t \*file)
- void `swapfile` (file\_t \*file, gtime\_t time, char \*msg)
- void `swapclose` (file\_t \*file)
- int `statefile` (file\_t \*file)
- int `readfile` (file\_t \*file, unsigned char \*buff, int nmax, char \*msg)
- int `writefile` (file\_t \*file, unsigned char \*buff, int n, char \*msg)
- void `syncfile` (file\_t \*file1, file\_t \*file2)
- void `decodetcppath` (const char \*path, char \*addr, char \*port, char \*user, char \*passwd, char \*mntpnt, char \*str)
- int `errsock` ()
- int `setsock` (socket\_t sock, char \*msg)

- `socket_t accept_nb` (`socket_t sock`, `struct sockaddr *addr`, `socklen_t *len`)
- `int connect_nb` (`socket_t sock`, `struct sockaddr *addr`, `socklen_t len`)
- `int recv_nb` (`socket_t sock`, `unsigned char *buff`, `int n`)
- `int send_nb` (`socket_t sock`, `unsigned char *buff`, `int n`)
- `int gentcp` (`tcp_t *tcp`, `int type`, `char *msg`)
- `void discontcp` (`tcp_t *tcp`, `int tcon`)
- `tcpsvr_t * opentcpsvr` (`const char *path`, `char *msg`)
- `void closetcpsvr` (`tcpsvr_t *tcpsvr`)
- `void updatetcpsvr` (`tcpsvr_t *tcpsvr`, `char *msg`)
- `int accsock` (`tcpsvr_t *tcpsvr`, `char *msg`)
- `int waittcpsvr` (`tcpsvr_t *tcpsvr`, `char *msg`)
- `int readtcpsvr` (`tcpsvr_t *tcpsvr`, `unsigned char *buff`, `int n`, `char *msg`)
- `int writetcpsvr` (`tcpsvr_t *tcpsvr`, `unsigned char *buff`, `int n`, `char *msg`)
- `int statetcpsvr` (`tcpsvr_t *tcpsvr`)
- `int consock` (`tcpcli_t *tcpcli`, `char *msg`)
- `tcpcli_t * opentcpcli` (`const char *path`, `char *msg`)
- `void closetcpcli` (`tcpcli_t *tcpcli`)
- `int waittcpcli` (`tcpcli_t *tcpcli`, `char *msg`)
- `int readtcpcli` (`tcpcli_t *tcpcli`, `unsigned char *buff`, `int n`, `char *msg`)
- `int writetcpcli` (`tcpcli_t *tcpcli`, `unsigned char *buff`, `int n`, `char *msg`)
- `int statetcpcli` (`tcpcli_t *tcpcli`)
- `int encbase64` (`char *str`, `const unsigned char *byte`, `int n`)
- `int reqntrip_s` (`ntrip_t *ntrip`, `char *msg`)
- `int reqntrip_c` (`ntrip_t *ntrip`, `char *msg`)
- `int rspntrip_s` (`ntrip_t *ntrip`, `char *msg`)
- `int rspntrip_c` (`ntrip_t *ntrip`, `char *msg`)
- `int waitntrip` (`ntrip_t *ntrip`, `char *msg`)
- `ntrip_t * openntrip` (`const char *path`, `int type`, `char *msg`)
- `void closentrip` (`ntrip_t *ntrip`)
- `int readntrip` (`ntrip_t *ntrip`, `unsigned char *buff`, `int n`, `char *msg`)
- `int writentrip` (`ntrip_t *ntrip`, `unsigned char *buff`, `int n`, `char *msg`)
- `int statentrip` (`ntrip_t *ntrip`)
- `void decodeftppath` (`const char *path`, `char *addr`, `char *file`, `char *user`, `char *passwd`, `int *topts`)
- `gtime_t nextdlttime` (`const int *topts`, `int stat`)
- `void * ftpthread` (`void *arg`)
- `ftp_t * openftp` (`const char *path`, `int type`, `char *msg`)
- `void closeftp` (`ftp_t *ftp`)
- `int readftp` (`ftp_t *ftp`, `unsigned char *buff`, `int n`, `char *msg`)
- `int stateftp` (`ftp_t *ftp`)
- `void strinitcom` ()
- `void strinit` (`stream_t *stream`)
- `int stropen` (`stream_t *stream`, `int type`, `int mode`, `const char *path`)
- `void strclose` (`stream_t *stream`)
- `void strsync` (`stream_t *stream1`, `stream_t *stream2`)
- `void strlock` (`stream_t *stream`)
- `void strunlock` (`stream_t *stream`)
- `int strread` (`stream_t *stream`, `unsigned char *buff`, `int n`)
- `int strwrite` (`stream_t *stream`, `unsigned char *buff`, `int n`)
- `int strstat` (`stream_t *stream`, `char *msg`)
- `void strsum` (`stream_t *stream`, `int *inb`, `int *inr`, `int *outb`, `int *outr`)
- `void strsetopt` (`const int *opt`)
- `void strsettimeout` (`stream_t *stream`, `int inactive_timeout`, `int tirecon`)
- `void strsetdir` (`const char *dir`)
- `void strsetproxy` (`const char *addr`)
- `gtime_t strgettime` (`stream_t *stream`)
- `void strsendnmea` (`stream_t *stream`, `const double *pos`)
- `int gen_hex` (`const char *msg`, `unsigned char *buff`)
- `void strsendcmd` (`stream_t *str`, `const char *cmd`)

### 11.228.1 Detailed Description

streaming functions

#### Authors

- 2007-2013, T. Takasu
- 2017, Javier Arribas
- 2017, Carles Fernandez

This is a derived work from RTKLIB <http://www.rtklib.com/> The original source code at <https://github.com/tomojitakasu/RTKLIB> is released under the BSD 2-clause license with an additional exclusive clause that does not apply here. This additional clause is reproduced below:

" The software package includes some companion executive binaries or shared libraries necessary to execute APs on Windows. These licenses succeed to the original ones of these software. "

Neither the executive binaries nor the shared libraries are required by, used or included in GNSS-SDR.

Copyright (C) 2007-2013, T. Takasu Copyright (C) 2017, Javier Arribas Copyright (C) 2017, Carles Fernandez All rights reserved.

SPDX-License-Identifier: BSD-2-Clause

Definition in file [rtklib\\_stream.h](#).

### 11.228.2 Macro Definition Documentation

#### 11.228.2.1 FTP\_CMD

```
#define FTP_CMD "wget" /* ftp/http command */
```

Definition at line 58 of file [rtklib\\_stream.h](#).

#### 11.228.2.2 FTP\_TIMEOUT

```
#define FTP_TIMEOUT 30 /* ftp/http timeout (s) */
```

Definition at line 59 of file [rtklib\\_stream.h](#).

#### 11.228.2.3 MAXCLI

```
#define MAXCLI 32 /* max client connection for tcp svr */
```

Definition at line 42 of file [rtklib\\_stream.h](#).

#### 11.228.2.4 MAXSTATMSG

```
#define MAXSTATMSG 32 /* max length of status message */
```

Definition at line 43 of file [rtklib\\_stream.h](#).

#### 11.228.2.5 NTRIP\_AGENT

```
#define NTRIP_AGENT "RTKLIB/" VER_RTKLIB
```

Definition at line 46 of file [rtklib\\_stream.h](#).

#### 11.228.2.6 NTRIP\_CLI\_PORT

```
#define NTRIP_CLI_PORT 2101 /* default ntrip-client connection port */
```

Definition at line 47 of file [rtklib\\_stream.h](#).

#### 11.228.2.7 NTRIP\_MAXRSP

```
#define NTRIP_MAXRSP 32768 /* max size of ntrip response */
```

Definition at line 49 of file [rtklib\\_stream.h](#).

#### 11.228.2.8 NTRIP\_MAXSTR

```
#define NTRIP_MAXSTR 256 /* max length of mountpoint string */
```

Definition at line 50 of file [rtklib\\_stream.h](#).



### 11.228.2.9 NTRIP\_RSP\_ERROR

```
#define NTRIP_RSP_ERROR "ERROR" /* ntrip response: error */
```

Definition at line 56 of file [rtklib\\_stream.h](#).

### 11.228.2.10 NTRIP\_RSP\_HTTP

```
#define NTRIP_RSP_HTTP "HTTP/" /* ntrip response: http */
```

Definition at line 55 of file [rtklib\\_stream.h](#).

### 11.228.2.11 NTRIP\_RSP\_OK\_CLI

```
#define NTRIP_RSP_OK_CLI "ICY 200 OK\r\n" /* ntrip response: client */
```

Definition at line 51 of file [rtklib\\_stream.h](#).

### 11.228.2.12 NTRIP\_RSP\_OK\_SVR

```
#define NTRIP_RSP_OK_SVR "OK\r\n" /* ntrip response: server */
```

Definition at line 52 of file [rtklib\\_stream.h](#).

### 11.228.2.13 NTRIP\_RSP\_SRCTBL

```
#define NTRIP_RSP_SRCTBL "SOURCETABLE 200 OK\r\n" /* ntrip response: source table */
```

Definition at line 53 of file [rtklib\\_stream.h](#).

### 11.228.2.14 NTRIP\_RSP\_TBLEND

```
#define NTRIP_RSP_TBLEND "ENDSOURCETABLE"
```

Definition at line 54 of file [rtklib\\_stream.h](#).

### 11.228.2.15 NTRIP\_SVR\_PORT

```
#define NTRIP_SVR_PORT 80 /* default ntrip-server connection port */
```

Definition at line 48 of file [rtklib\\_stream.h](#).

### 11.228.2.16 SERIBUFFSIZE

```
#define SERIBUFFSIZE 4096 /* serial buffer size (bytes) */
```

Definition at line 40 of file [rtklib\\_stream.h](#).

### 11.228.2.17 TIMETAGH\_LEN

```
#define TIMETAGH_LEN 64 /* time tag file header length */
```

Definition at line 41 of file [rtklib\\_stream.h](#).

### 11.228.2.18 TINTACT

```
#define TINTACT 200 /* period for stream active (ms) */
```

Definition at line 39 of file [rtklib\\_stream.h](#).

### 11.228.2.19 VER\_RTKLIB

```
#define VER_RTKLIB "2.4.2"
```

Definition at line 45 of file [rtklib\\_stream.h](#).

## 11.229 rtklib\_stream.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file rtklib_stream.h
00003  * \brief streaming functions
00004  * \authors <ul>
00005  *         <li> 2007-2013, T. Takasu
00006  *         <li> 2017, Javier Arribas
00007  *         <li> 2017, Carles Fernandez
00008  *         </ul>
00009  *
00010  * This is a derived work from RTKLIB http://www.rtklib.com/
00011  * The original source code at https://github.com/tomojitakasu/RTKLIB is
00012  * released under the BSD 2-clause license with an additional exclusive clause
00013  * that does not apply here. This additional clause is reproduced below:
00014  *
00015  * " The software package includes some companion executive binaries or shared
00016  * libraries necessary to execute APs on Windows. These licenses succeed to the
00017  * original ones of these software. "
00018  *
00019  * Neither the executive binaries nor the shared libraries are required by, used
00020  * or included in GNSS-SDR.
00021  *
00022  * -----
00023  * Copyright (C) 2007-2013, T. Takasu
00024  * Copyright (C) 2017, Javier Arribas
00025  * Copyright (C) 2017, Carles Fernandez
00026  * All rights reserved.
00027  *
00028  * SPDX-License-Identifier: BSD-2-Clause
00029  *
00030  *-----*/
00031
00032 #ifndef GNSS_SDR_RTKLIB_STREAM_H
00033 #define GNSS_SDR_RTKLIB_STREAM_H
00034
00035 #include "rtklib.h"
00036
00037 /* constants -----*/
00038
00039 #define TINTACT 200          /* period for stream active (ms) */
00040 #define SERIBUFFSIZE 4096   /* serial buffer size (bytes) */
00041 #define TIMETAGH_LEN 64     /* time tag file header length */
00042 #define MAXCLI 32           /* max client connection for tcp svr */
00043 #define MAXSTATMSG 32      /* max length of status message */
00044
00045 #define VER_RTKLIB "2.4.2"
00046 #define NTRIP_AGENT "RTKLIB/" VER_RTKLIB
00047 #define NTRIP_CLI_PORT 2101 /* default ntrip-client connection port */
00048 #define NTRIP_SVR_PORT 80   /* default ntrip-server connection port */
00049 #define NTRIP_MAXRSP 32768  /* max size of ntrip response */
00050 #define NTRIP_MAXSTR 256    /* max length of mountpoint string */
00051 #define NTRIP_RSP_OK_CLI "ICY 200 OK\r\n" /* ntrip response: client */
00052 #define NTRIP_RSP_OK_SVR "OK\r\n"        /* ntrip response: server */
00053 #define NTRIP_RSP_SRCTBL "SOURCETABLE 200 OK\r\n" /* ntrip response: source table */
00054 #define NTRIP_RSP_TBLEND "ENDSOURCETABLE"
00055 #define NTRIP_RSP_HTTP "HTTP/" /* ntrip response: http */
00056 #define NTRIP_RSP_ERROR "ERROR" /* ntrip response: error */
00057
00058 #define FTP_CMD "wget" /* ftp/http command */
00059 #define FTP_TIMEOUT 30 /* ftp/http timeout (s) */
00060
00061 serial_t *openserial(const char *path, int mode, char *msg);
00062
00063 void closeserial(serial_t *serial);
00064
00065 int readserial(serial_t *serial, unsigned char *buff, int n, char *msg);
00066
00067 int writeserial(serial_t *serial, unsigned char *buff, int n, char *msg);
00068
00069 int stateserial(serial_t *serial);
00070
00071 int openfile_(file_t *file, gtime_t time, char *msg);
00072
00073 void closefile_(file_t *file);
00074
00075 file_t *openfile(const char *path, int mode, char *msg);
00076
00077 void closefile(file_t *file);
00078
00079 void swapfile(file_t *file, gtime_t time, char *msg);
00080
00081 void swapclose(file_t *file);
00082
00083

```

```
00084 int statefile(file_t *file);
00085
00086 int readfile(file_t *file, unsigned char *buff, int nmax, char *msg);
00087
00088 int writefile(file_t *file, unsigned char *buff, int n, char *msg);
00089
00090 void syncfile(file_t *file1, file_t *file2);
00091
00092 void decodetcpvpath(const char *path, char *addr, char *port, char *user,
00093     char *passwd, char *mntpnt, char *str);
00094
00095 int errsock();
00096
00097 int setsock(socket_t sock, char *msg);
00098
00099 socket_t accept_nb(socket_t sock, struct sockaddr *addr, socklen_t *len);
00100
00101 int connect_nb(socket_t sock, struct sockaddr *addr, socklen_t len);
00102
00103 int recv_nb(socket_t sock, unsigned char *buff, int n);
00104
00105 int send_nb(socket_t sock, unsigned char *buff, int n);
00106
00107 int gentcp(tcp_t *tcp, int type, char *msg);
00108
00109 void discontcp(tcp_t *tcp, int tcon);
00110
00111 tcpsvr_t *opentcpsvr(const char *path, char *msg);
00112
00113 void closetcpsvr(tcpsvr_t *tcpsvr);
00114
00115 void updatetcpsvr(tcpsvr_t *tcpsvr, char *msg);
00116
00117 int accsock(tcpsvr_t *tcpsvr, char *msg);
00118
00119 int waittcpsvr(tcpsvr_t *tcpsvr, char *msg);
00120
00121 int readtcpsvr(tcpsvr_t *tcpsvr, unsigned char *buff, int n, char *msg);
00122
00123 int writetcpsvr(tcpsvr_t *tcpsvr, unsigned char *buff, int n, char *msg);
00124
00125 int statetcpsvr(tcpsvr_t *tcpsvr);
00126
00127 int consock(tcpcli_t *tcpcli, char *msg);
00128
00129 tcpcli_t *opentcpcli(const char *path, char *msg);
00130
00131 void closetcpcli(tcpcli_t *tcpcli);
00132
00133 int waittcpcli(tcpcli_t *tcpcli, char *msg);
00134
00135 int readtcpcli(tcpcli_t *tcpcli, unsigned char *buff, int n, char *msg);
00136
00137 int writetcpcli(tcpcli_t *tcpcli, unsigned char *buff, int n, char *msg);
00138
00139 int statetcpcli(tcpcli_t *tcpcli);
00140
00141 int encbase64(char *str, const unsigned char *byte, int n);
00142
00143 int reqntrip_s(ntrip_t *ntrip, char *msg);
00144
00145 int reqntrip_c(ntrip_t *ntrip, char *msg);
00146
00147 int rspntrip_s(ntrip_t *ntrip, char *msg);
00148
00149 int rspntrip_c(ntrip_t *ntrip, char *msg);
00150
00151 int waitntrip(ntrip_t *ntrip, char *msg);
00152
00153 ntrip_t *openntrip(const char *path, int type, char *msg);
00154
00155 void closentrip(ntrip_t *ntrip);
00156
00157 int readntrip(ntrip_t *ntrip, unsigned char *buff, int n, char *msg);
00158
00159 int writentrip(ntrip_t *ntrip, unsigned char *buff, int n, char *msg);
00160
00161 int statentrip(ntrip_t *ntrip);
00162
00163 void decodeftppath(const char *path, char *addr, char *file, char *user,
00164     char *passwd, int *topts);
00165
00166 gtime_t nextdltime(const int *topts, int stat);
00167
00168 void *ftpthread(void *arg);
00169
00170 ftp_t *openftp(const char *path, int type, char *msg);
```

```

00171
00172 void closeftp(ftp_t *ftp);
00173
00174 int readftp(ftp_t *ftp, unsigned char *buff, int n, char *msg);
00175
00176 int stateftp(ftp_t *ftp);
00177
00178 void strinitcom();
00179
00180 void strinit(stream_t *stream);
00181
00182 int stropen(stream_t *stream, int type, int mode, const char *path);
00183
00184 void strclose(stream_t *stream);
00185
00186 void strsync(stream_t *stream1, stream_t *stream2);
00187
00188 void strlock(stream_t *stream);
00189
00190 void strunlock(stream_t *stream);
00191
00192 int strread(stream_t *stream, unsigned char *buff, int n);
00193
00194 int strwrite(stream_t *stream, unsigned char *buff, int n);
00195
00196 int strstat(stream_t *stream, char *msg);
00197
00198 void strsum(stream_t *stream, int *inb, int *inr, int *outb, int *outr);
00199
00200 void strsetopt(const int *opt);
00201
00202 void strsettimeout(stream_t *stream, int inactive_timeout, int tirecon);
00203
00204 void strsetdir(const char *dir);
00205
00206 void strsetproxy(const char *addr);
00207
00208 gtime_t strgettime(stream_t *stream);
00209
00210 void strsendnmea(stream_t *stream, const double *pos);
00211
00212 int gen_hex(const char *msg, unsigned char *buff);
00213
00214 void strsendcmd(stream_t *str, const char *cmd);
00215
00216
00217 #endif

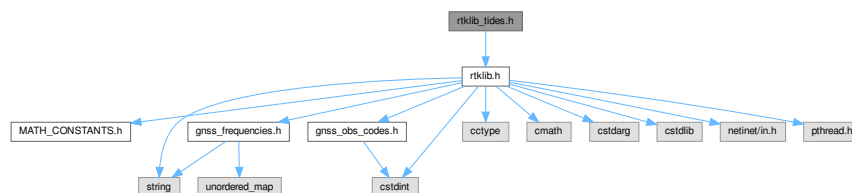
```

## 11.230 rtklib\_tides.h File Reference

Tidal displacement corrections.

```
#include "rtklib.h"
```

Include dependency graph for rtklib\_tides.h:



### Functions

- void **tide\_pl** (const double \*eu, const double \*rp, double GMp, const double \*pos, double \*dr)
- void **tide\_solid** (const double \*rsun, const double \*rmoon, const double \*pos, const double \*E, double gmst, int opt, double \*dr)
- void **tide\_oload** (gtime\_t tut, const double \*odisp, double \*denu)
- void **iers\_mean\_pole** (gtime\_t tut, double \*xp\_bar, double \*yp\_bar)
- void **tide\_pole** (gtime\_t tut, const double \*pos, const double \*erpv, double \*denu)
- void **tidedisp** (gtime\_t tutc, const double \*rr, int opt, const erp\_t \*erp, const double \*odisp, double \*dr)

**Variables**

- const double **GME** = 3.986004415E+14
- const double **GMS** = 1.327124E+20
- const double **GMM** = 4.902801E+12

**11.230.1 Detailed Description**

Tidal displacement corrections.

**Authors**

- 2015, T. Takasu
- 2017, Javier Arribas
- 2017, Carles Fernandez

This is a derived work from RTKLIB <http://www.rtklib.com/> The original source code at <https://github.com/tomojitakasu/RTKLIB> is released under the BSD 2-clause license with an additional exclusive clause that does not apply here. This additional clause is reproduced below:

" The software package includes some companion executive binaries or shared libraries necessary to execute APs on Windows. These licenses succeed to the original ones of these software. "

Neither the executive binaries nor the shared libraries are required by, used or included in GNSS-SDR.

Copyright (C) 2015, T. Takasu Copyright (C) 2017, Javier Arribas Copyright (C) 2017, Carles Fernandez All rights reserved.

SPDX-License-Identifier: BSD-2-Clause

References: [1] D.D.McCarthy, IERS Technical Note 21, IERS Conventions 1996, July 1996 [2] D.D.McCarthy and G.Petit, IERS Technical Note 32, IERS Conventions 2003, November 2003 [3] D.A.Vallado, Fundamentals of Astrodynamics and Applications 2nd ed, Space Technology Library, 2004 [4] J.Kouba, A Guide to using International GNSS Service (IGS) products, May 2009 [5] G.Petit and B.Luzum (eds), IERS Technical Note No. 36, IERS

**11.230.1.1 Conventions (2010), 2010**

Definition in file [rtklib\\_tides.h](#).

**11.230.2 Variable Documentation****11.230.2.1 GME**

```
const double GME = 3.986004415E+14
```

Definition at line 51 of file [rtklib\\_tides.h](#).

**11.230.2.2 GMM**

```
const double GMM = 4.902801E+12
```

Definition at line 53 of file [rtklib\\_tides.h](#).

**11.230.2.3 GMS**

```
const double GMS = 1.327124E+20
```

Definition at line 52 of file [rtklib\\_tides.h](#).

**11.231 rtklib\_tides.h**

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file rtklib_tides.h
00003  * \brief Tidal displacement corrections
00004  * \authors <ul>
00005  *         <li> 2015, T. Takasu
00006  *         <li> 2017, Javier Arribas
00007  *         <li> 2017, Carles Fernandez
00008  *         </ul>
```

```

00009  *
00010  * This is a derived work from RTKLIB http://www.rtklib.com/
00011  * The original source code at https://github.com/tomojitakasu/RTKLIB is
00012  * released under the BSD 2-clause license with an additional exclusive clause
00013  * that does not apply here. This additional clause is reproduced below:
00014  *
00015  * " The software package includes some companion executive binaries or shared
00016  * libraries necessary to execute APs on Windows. These licenses succeed to the
00017  * original ones of these software. "
00018  *
00019  * Neither the executive binaries nor the shared libraries are required by, used
00020  * or included in GNSS-SDR.
00021  *
00022  * -----
00023  * Copyright (C) 2015, T. Takasu
00024  * Copyright (C) 2017, Javier Arribas
00025  * Copyright (C) 2017, Carles Fernandez
00026  * All rights reserved.
00027  *
00028  * SPDX-License-Identifier: BSD-2-Clause
00029  *
00030  * References:
00031  * [1] D.D.McCarthy, IERS Technical Note 21, IERS Conventions 1996, July 1996
00032  * [2] D.D.McCarthy and G.Petit, IERS Technical Note 32, IERS Conventions
00033  *      2003, November 2003
00034  * [3] D.A.Vallado, Fundamentals of Astrodynamics and Applications 2nd ed,
00035  *      Space Technology Library, 2004
00036  * [4] J.Kouba, A Guide to using International GNSS Service (IGS) products,
00037  *      May 2009
00038  * [5] G.Petit and B.Luzum (eds), IERS Technical Note No. 36, IERS
00039  *      Conventions (2010), 2010
00040  * -----
00041  */
00042
00043
00044 #ifndef GNSS_SDR_RTKLIB_TIDES_H
00045 #define GNSS_SDR_RTKLIB_TIDES_H
00046
00047
00048 #include "rtklib.h"
00049
00050
00051 const double GME = 3.986004415E+14; /* earth gravitational constant */
00052 const double GMS = 1.327124E+20; /* sun gravitational constant */
00053 const double GMM = 4.902801E+12; /* moon gravitational constant */
00054
00055 void tide_pl(const double *eu, const double *rp, double GMp,
00056             const double *pos, double *dr);
00057
00058 void tide_solid(const double *rsun, const double *rmoon,
00059               const double *pos, const double *E, double gmst, int opt,
00060               double *dr);
00061
00062 void tide_oload(gtime_t tut, const double *odisp, double *denu);
00063
00064 void iers_mean_pole(gtime_t tut, double *xp_bar, double *yp_bar);
00065
00066
00067 void tide_pole(gtime_t tut, const double *pos, const double *erpv,
00068               double *denu);
00069
00070 void tidedisp(gtime_t tutc, const double *rr, int opt, const erp_t *erp,
00071               const double *odisp, double *dr);
00072 #endif

```

## 11.232 short\_x2\_to\_cshort.h File Reference

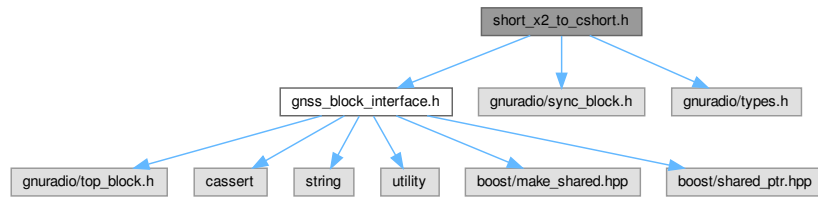
Adapts two short streams into a `std::complex<short>` stream.

```

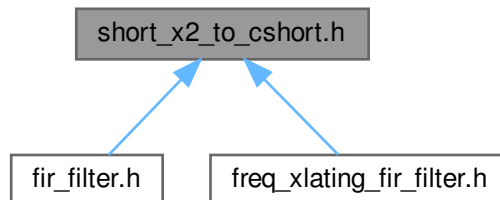
#include "gnss_block_interface.h"
#include <gnuradio/sync_block.h>
#include <gnuradio/types.h>

```

Include dependency graph for short\_x2\_to\_cshort.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [short\\_x2\\_to\\_cshort](#)

*This class adapts two short streams into a `std::complex<short>` stream.*

## Typedefs

- using [short\\_x2\\_to\\_cshort\\_sptr](#) = `gnss_shared_ptr<short\_x2\_to\_cshort>`

## Functions

- `short_x2_to_cshort_sptr make_short_x2_to_cshort ()`

### 11.232.1 Detailed Description

Adapts two short streams into a `std::complex<short>` stream.

#### Author

Carles Fernandez Prades, [cfernandez\(at\)cttc.es](mailto:cfernandez(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [short\\_x2\\_to\\_cshort.h](#).

---

## 11.233 short\_x2\_to\_cshort.h

[Go to the documentation of this file.](#)

00001 /\*!

```

00002  * \file short_x2_to_cshort.h
00003  * \brief Adapts two short streams into a std::complex<short> stream
00004  * \author Carles Fernandez Prades, cfernandez(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_SHORT_X2_TO_CSHORT_H
00018 #define GNSS_SDR_SHORT_X2_TO_CSHORT_H
00019
00020
00021 #include "gnss_block_interface.h"
00022 #include <gnuradio/sync_block.h>
00023 #include <gnuradio/types.h> // for gr_vector_const_void_star
00024
00025 /** \addtogroup Algorithms_Library
00026  * \{ */
00027 /** \addtogroup Algorithm_libs algorithms_libs
00028  * \{ */
00029
00030
00031 class short_x2_to_cshort;
00032
00033 using short_x2_to_cshort_sptr = gnss_shared_ptr<short_x2_to_cshort>;
00034
00035 short_x2_to_cshort_sptr make_short_x2_to_cshort();
00036
00037 /*!
00038  * \brief This class adapts two short streams into a std::complex<short> stream
00039  */
00040 class short_x2_to_cshort : public gr::sync_block
00041 {
00042 public:
00043     int work(int noutput_items,
00044             gr_vector_const_void_star &input_items,
00045             gr_vector_void_star &output_items);
00046
00047 private:
00048     friend short_x2_to_cshort_sptr make_short_x2_to_cshort();
00049     short_x2_to_cshort();
00050 };
00051
00052
00053 /** \} */
00054 /** \} */
00055 #endif // GNSS_SDR_SHORT_X2_TO_CSHORT_H

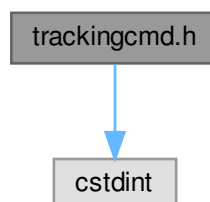
```

## 11.234 trackingcmd.h File Reference

Class that stores information to update the GNSS signal tracking estimations.

```
#include <cstdint>
```

Include dependency graph for trackingcmd.h:





**Classes**

- class [TrackingCmd](#)

**11.234.1 Detailed Description**

Class that stores information to update the GNSS signal tracking estimations.

**Author**

Javier Arribas, 2021. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [trackingcmd.h](#).

---

**11.235 trackingcmd.h**

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file trackingcmd.h
00003  * \brief Class that stores information to update the GNSS signal tracking estimations
00004  * \author Javier Arribas, 2021. jarribas(at)cttc.es
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_TRACKINGCMD_H_
00019 #define GNSS_SDR_TRACKINGCMD_H_
00020
00021 #include <stdint>
00022
00023 /** \addtogroup Algorithms_Library
00024  * \{ */
00025 /** \addtogroup Algorithm_libs algorithms_libs
00026  * \{ */
00027
00028 class TrackingCmd
00029 {
00030 public:
00031     TrackingCmd();
00032
00033     bool enable_carrier_nco_cmd = false;
00034     bool enable_code_nco_cmd = false;
00035     double code_freq_chips = 0.0;
00036     double carrier_freq_hz = 0.0;
00037     double carrier_freq_rate_hz_s = 0.0;
00038     uint64_t sample_counter = 0UL;
00039 };
00040
00041 /** \} */
00042 /** \} */
00043 #endif // GNSS_SDR_TRACKINGCMD_H_

```

**11.236 hybrid\_observables.h File Reference**

Implementation of an adapter of an observables block accepting all kind of signals to a [ObservablesInterface](#).

```

#include "gnss_synchro.h"
#include "hybrid_observables_gs.h"
#include "observables_interface.h"
#include <gnuradio/gr_complex.h>
#include <gnuradio/runtime_types.h>
#include <cstdint>

```



```

00043 class HybridObservables : public ObservablesInterface
00044 {
00045 public:
00046     HybridObservables(const ConfigurationInterface* configuration,
00047         const std::string& role,
00048         unsigned int in_streams,
00049         unsigned int out_streams);
00050
00051     ~HybridObservables() = default;
00052
00053     inline std::string role() override
00054     {
00055         return role_;
00056     }
00057
00058     /// Returns "Hybrid_Observables"
00059     inline std::string implementation() override
00060     {
00061         return "Hybrid_Observables";
00062     }
00063
00064     void connect(gr::top_block_sptr top_block) override;
00065     void disconnect(gr::top_block_sptr top_block) override;
00066     gr::basic_block_sptr get_left_block() override;
00067     gr::basic_block_sptr get_right_block() override;
00068
00069     inline void reset() override
00070     {
00071         return;
00072     }
00073
00074     /// All blocks must have an item_size() function implementation
00075     inline size_t item_size() override
00076     {
00077         return sizeof(Gnss_Synchro);
00078     }
00079
00080 private:
00081     hybrid_observables_gs_sptr observables_;
00082     std::string dump_filename_;
00083     std::string role_;
00084     unsigned int in_streams_;
00085     unsigned int out_streams_;
00086     bool dump_;
00087     bool dump_mat_;
00088 };
00089
00090 /** @} */
00091 /** @} */
00092 #endif

```

## 11.238 hybrid\_observables\_gs.h File Reference

Interface of the observables computation block.

```

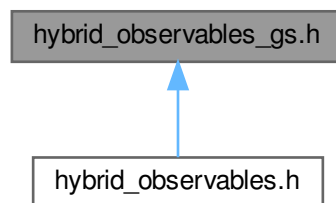
#include "gnss_block_interface.h"
#include "gnss_time.h"
#include "obs_conf.h"
#include <boost/circular_buffer.hpp>
#include <gnuradio/block.h>
#include <gnuradio/types.h>
#include <cstdint>
#include <cstdint>
#include <fstream>
#include <memory>
#include <queue>
#include <string>
#include <typeinfo>
#include <vector>

```

Include dependency graph for `hybrid_observables_gs.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [hybrid\\_observables\\_gs](#)  
*This class implements a block that computes observables.*

## Typedefs

- using [hybrid\\_observables\\_gs\\_sptr](#) = `gnss_shared_ptr<hybrid\_observables\_gs>`

## Functions

- `hybrid_observables_gs_sptr hybrid_observables_gs_make (const Obs\_Conf &conf_)`

### 11.238.1 Detailed Description

Interface of the observables computation block.

#### Author

Mara Branzanti 2013. [mara.branzanti\(at\)gmail.com](mailto:mara.branzanti(at)gmail.com)

Javier Arribas 2013. [jarribas\(at\)cttc.es](mailto:jarribas(at)cttc.es)

Antonio Ramos 2018. [antonio.ramos\(at\)cttc.es](mailto:antonio.ramos(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [hybrid\\_observables\\_gs.h](#).

---

## 11.239 hybrid\_observables\_gs.h

[Go to the documentation of this file.](#)

```
00001  /*!
00002  * \file hybrid_observables_gs.h
00003  * \brief Interface of the observables computation block
00004  * \author Mara Branzanti 2013. mara.branzanti\(at\)gmail.com
00005  * \author Javier Arribas 2013. jarribas\(at\)cttc.es
```

```

00006  * \author Antonio Ramos 2018. antonio.ramos(at)cttc.es
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019
00020 #ifndef GNSS_SDR_HYBRID_OBSERVABLES_GS_H
00021 #define GNSS_SDR_HYBRID_OBSERVABLES_GS_H
00022
00023 #include "gnss_block_interface.h"
00024 #include "gnss_time.h" // for timetags produced by Tracking
00025 #include "obs_conf.h"
00026 #include <boost/circular_buffer.hpp> // for boost::circular_buffer
00027 #include <gnuradio/block.h> // for block
00028 #include <gnuradio/types.h> // for gr_vector_int
00029 #include <cstdint> // for size_t
00030 #include <cstdint> // for int32_t
00031 #include <fstream> // for std::ofstream
00032 #include <memory> // for std::shared, std::unique_ptr
00033 #include <queue> // for std::queue
00034 #include <string> // for std::string
00035 #include <typeinfo> // for typeid
00036 #include <vector> // for std::vector
00037
00038 /** \addtogroup Observables
00039  * \{ */
00040 /** \addtogroup Observables_gnuradio_blocks obs_gr_blocks
00041  * GNU Radio blocks for the computation of GNSS observables
00042  * \{ */
00043
00044
00045 class Gnss_Synchro;
00046 class hybrid_observables_gs;
00047
00048 template <class T>
00049 class Gnss_circular_deque;
00050
00051 using hybrid_observables_gs_sptr = gnss_shared_ptr<hybrid_observables_gs>;
00052
00053 hybrid_observables_gs_sptr hybrid_observables_gs_make(const Obs_Conf& conf_);
00054
00055 /*!
00056  * \brief This class implements a block that computes observables
00057  */
00058 class hybrid_observables_gs : public gr::block
00059 {
00060 public:
00061     ~hybrid_observables_gs();
00062     void forecast(int noutput_items, gr_vector_int& ninput_items_required);
00063     int general_work(int noutput_items, gr_vector_int& ninput_items,
00064                     gr_vector_const_void_star& input_items, gr_vector_void_star& output_items);
00065 private:
00066     friend hybrid_observables_gs_sptr hybrid_observables_gs_make(const Obs_Conf& conf_);
00067
00068     explicit hybrid_observables_gs(const Obs_Conf& conf_);
00069
00070     const size_t d_double_type_hash_code = typeid(double).hash_code();
00071     const size_t d_int_type_hash_code = typeid(int).hash_code();
00072
00073     void msg_handler_pvt_to_observables(const pmt::pmt_t& msg);
00074     double compute_T_rx_s(const Gnss_Synchro& a) const;
00075     bool interp_trk_obs(Gnss_Synchro& interpolated_obs, uint32_t ch, uint64_t rx_clock) const;
00076     void update_TOW(const std::vector<Gnss_Synchro>& data);
00077     void compute_pranges(std::vector<Gnss_Synchro>& data) const;
00078     void smooth_pseudoranges(std::vector<Gnss_Synchro>& data);
00079
00080     void set_tag_timestamp_in_sdr_timeframe(const std::vector<Gnss_Synchro>& data, uint64_t rx_clock);
00081     int32_t save_matfile() const;
00082
00083     Obs_Conf d_conf;
00084
00085     std::unique_ptr<Gnss_circular_deque<Gnss_Synchro> d_gnss_synchro_history; // Tracking observable
00086     history
00087
00088     boost::circular_buffer<uint64_t> d_Rx_clock_buffer; // time history
00089
00090     std::vector<std::queue<GnssTime> d_SourceTagTimestamps;
00091     std::queue<GnssTime> d_TimeChannelTagTimestamps;

```

```

00092
00093     std::vector<bool> d_channel_last_pll_lock;
00094     std::vector<double> d_channel_last_pseudorange_smooth;
00095     std::vector<double> d_channel_last_carrier_phase_rads;
00096
00097     std::string d_dump_filename;
00098
00099     std::ofstream d_dump_file;
00100
00101     double d_smooth_filter_M;
00102     double d_T_rx_step_s;
00103     double d_last_rx_clock_round20ms_error;
00104
00105     uint32_t d_T_rx_TOW_ms;
00106     uint32_t d_T_rx_step_ms;
00107     uint32_t d_T_status_report_timer_ms;
00108     uint32_t d_nchannels_in;
00109     uint32_t d_nchannels_out;
00110
00111     bool d_T_rx_TOW_set; // rx time follow GPST
00112     bool d_always_output_gs;
00113     bool d_dump;
00114     bool d_dump_mat;
00115 };
00116
00117 /** \} */
00118 /** \} */
00119 #endif // GNSS_SDR_HYBRID_OBSERVABLES_GS_H

```

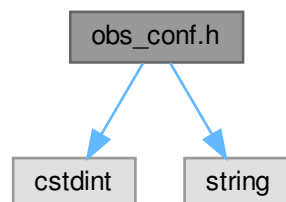
## 11.240 obs\_conf.h File Reference

Class that contains all the configuration parameters for generic observables block.

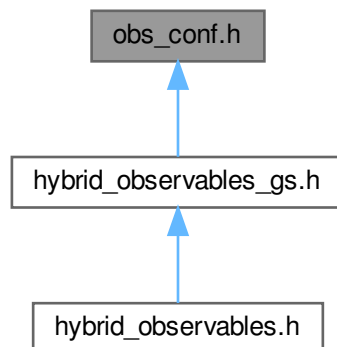
```
#include <cstdint>
```

```
#include <string>
```

Include dependency graph for obs\_conf.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Obs\\_Conf](#)

### 11.240.1 Detailed Description

Class that contains all the configuration parameters for generic observables block.

#### Author

Javier Arribas, 2020. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [obs\\_conf.h](#).

---

## 11.241 obs\_conf.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file obs_conf.h
00003  * \brief Class that contains all the configuration parameters for generic
00004  * observables block
00005  * \author Javier Arribas, 2020. jarribas(at)cttc.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_OBS_CONF_H
00019 #define GNSS_SDR_OBS_CONF_H
00020
00021 #include <stdint>
00022 #include <string>
00023
00024 /** \addtogroup Observables
00025  *  \{ */
00026 /** \addtogroup Observables_libs observables_libs
00027  * Utilities for GNSS observables configuration.
00028  *  \{ */
  
```

```

00029
00030 class Obs_Conf
00031 {
00032 public:
00033     Obs_Conf();
00034
00035     std::string dump_filename{"obs_dump.dat"};
00036     int32_t smoothing_factor{0};
00037     uint32_t nchannels_in{0U};
00038     uint32_t nchannels_out{0U};
00039     uint32_t observable_interval_ms{20U};
00040     bool enable_carrier_smoothing{false};
00041     bool always_output_gs{false};
00042     bool dump{false};
00043     bool dump_mat{false};
00044     bool enable_E6{false};
00045 };
00046
00047 /** \} */
00048 /** \} */
00049 #endif // GNSS_SDR_OBS_CONF_H

```

## 11.242 rtklib\_pvt.h File Reference

Interface of a Position Velocity and Time computation block.

```

#include "gnss_synchro.h"
#include "pvt_interface.h"
#include "rtklib.h"
#include "rtklib_pvt_gs.h"
#include <gnuradio/gr_complex.h>
#include <gnuradio/runtime_types.h>
#include <cstdio>
#include <ctime>
#include <map>
#include <string>

```

Include dependency graph for rtklib\_pvt.h:



### Classes

- class [Rtklib\\_Pvt](#)

*This class implements a [PvtInterface](#) for the RTKLIB PVT block.*

### 11.242.1 Detailed Description

Interface of a Position Velocity and Time computation block.

#### Author

Javier Arribas, 2017. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [rtklib\\_pvt.h](#).

---

## 11.243 rtklib\_pvt.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file rtklib_pvt.h
00003  * \brief Interface of a Position Velocity and Time computation block

```



```

00004  * \author Javier Arribas, 2017. jarribas(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_RTKLIB_PVT_H
00019 #define GNSS_SDR_RTKLIB_PVT_H
00020
00021 #include "gnss_synchro.h"
00022 #include "pvt_interface.h" // for PvtInterface
00023 #include "rtklib.h" // for rtk_t
00024 #include "rtklib_pvt_gs.h" // for rtklib_pvt_gs_sptr
00025 #include <gnuradio/gr_complex.h> // for gr_complex
00026 #include <gnuradio/runtime_types.h> // for basic_block_sptr, top_block_sptr
00027 #include <cstdint> // for size_t
00028 #include <ctime> // for time_t
00029 #include <map> // for map
00030 #include <string> // for string
00031
00032 /** \addtogroup PVT
00033  * Computation of Position, Velocity and Time from GNSS observables.
00034  * \{ */
00035 /** \addtogroup PVT_adapters pvt_adapters
00036  * Wrap GNU Radio PVT solvers with a PvtInterface
00037  * \{ */
00038
00039 class ConfigurationInterface;
00040 class Galileo_Almanac;
00041 class Galileo_Ephemeris;
00042 class Gps_Almanac;
00043 class Gps_Ephemeris;
00044
00045 /*!
00046  * \brief This class implements a PvtInterface for the RTKLIB PVT block
00047  *
00048  * Global configuration options used:
00049  *
00050  * GNSS-SDR.pre_2009_file - flag indicating a file older than 2009 rollover should be processed
00051  * GNSS-SDR.observable_interval_ms - (20)
00052  *
00053  * It supports the following configuration options:
00054  *
00055  * .dump - (false)
00056  * .dump_filename - (".pvt.dat")
00057  * .dump_mat - (true)
00058  * .rtk_trace_level - debug level for the RTKLIB methods (0)
00059  *
00060  * .output_rate_ms - (500)
00061  * Note that the actual rate is the least common multiple of this value and
00062  * GNSS-SDR.observable_interval_ms
00063  *
00064  * .display_rate_ms - (500)
00065  *
00066  * .flag_nmea_tty_port - (false)
00067  * .nmea_dump_filename - (".nmea_pvt.nmea")
00068  * .nmea_dump_devname - ("/dev/tty1")
00069  *
00070  * .rinex_version - (3) overridden by -RINEX_version=n.nn command line argument
00071  * .rinexobs_rate_ms - rate at which RINEX observations are written (1000). Note that
00072  * the actual rate is the least common multiple of this value and
00073  * .output_rate_ms
00074  * .rinex_name - (-RINEX_name command-line argument)
00075  *
00076  * .flag_rtcmm_tty_port - (false)
00077  * .rtcmm_dump_devname - ("/dev/pts/1")
00078  * .flag_rtcmm_server - (false)
00079  * .rtcmm_tcp_port - (2101)
00080  * .rtcmm_station_id - (1234)
00081  * Output rates ... all values are LCM with the computed output rate (above)
00082  * .rtcmm_MT1019_rate_ms - (5000)
00083  * .rtcmm_MT1020_rate_ms - (5000)
00084  * .rtcmm_MT1045_rate_ms - (5000)
00085  * .rtcmm_MSM_rate_ms - (1000)
00086  * .rtcmm_MT1077_rate_ms - (.rtcmm_MSM_rate_ms)
00087  * .rtcmm_MT1087_rate_ms - (.rtcmm_MSM_rate_ms)
00088  * .rtcmm_MT1097_rate_ms - (.rtcmm_MSM_rate_ms)
00089  *
00090  * .kml_rate_ms - (1000)

```

```

00089 * .gpx_rate_ms - (1000)
00090 * .geojson_rate_ms - (1000)
00091 * .nmea_rate_ms - (1000)
00092 *
00093 * .positioning_mode - The RTKLIB positioning mode. ("Single") Supported values are "Single",
00094 * "Static", "Kinematic", "PPP_Static" and "PPP_Kinematic". Unsupported modes
00095 * include DGPS/DGNSS, Moving Baseline, Fixed, and PPP-fixed
00096 * .num_bands - number of frequencies to use, between 1 and 3. Default is based on the channels
configured
00097 * .elevation_mask - (15.0). Value must be in the range [0,90.0]
00098 * .dynamics_model - (0) 0:none, 1:velocity, 2:acceleration
00099
00100 * .iono_model - ("OFF"). Supported values are "OFF", "Broadcast", "SBAS", "Iono-Free-LC",
00101 * "Estimate_STEC", "IONEX". Unsupported values include QZSS broadcast, QZSS
00102 * LEX, and SLANT TEC.
00103 * .trop_model - ("OFF"). Supported values are "OFF", "Saastamoinen", "SBAS", "Estimate_ZTD", and
00104 * "Estimate_ZTD_Grad". Unsupported values include ZTD correction and ZTD+grad
00105 * correction
00106 * .phwindup - phase windup correction for PPP modes (0)
00107 * .reject_GPS_IIA - whether the GPS Block IIA satellites in eclipse are excluded (0). Only applies
in PPP-* modes
00108 * .raim_fde - whether RAIM (receiver autonomous integrity monitoring) FDE (fault detection and
exclusion) is enabled (0)
00109 * .earth_tide - (0)
00110 * .navigation_system - mask of navigation systems to use. Default based on configured channels
00111 * 0x01:GPS, 0x02:SBAS, 0x04:GLONASS, 0x08:Galileo, 0x10:QZSS, 0x20:BeiDou,
00112 * 0x40:IRNS, 0x80:LEO
00113 *
00114 * .AR_GPS - Ambiguity Resolution mode for GPS ("Continuous"). Supported values are "OFF",
00115 * "Continuous", "Instantaneous", "Fix-and-Hold", "PPP-AR". Unsupported values
00116 * include PPP-AR ILS, WLNL, and TCAR.
00117 * .AR_GLO - Ambiguity Resolution mode for GLONASS (1). Value must be in the range [0,3].
(0:off,1:on,2:auto cal,3:ext cal)
00118 * .AR_DBS - Ambiguity Resolution Mode for BeiDou (1). Value must be in the range [0,1]. (0:off,1:on)
00119 * .min_ratio_to_fix_ambiguity - (3.0)
00120 * .min_lock_to_fix_ambiguity - (0)
00121 * .min_elevation_to_fix_ambiguity - minimum elevation (deg) to fix integer ambiguity (0.0)
00122 * .outage_reset_ambiguity - (5)
00123 * .slip_threshold - (0.05)
00124 * .threshold_reject_gdop - if GDOP is over this value, the observable is excluded (30.0)
00125 * .threshold_reject_innovation - if innovation is over this value, the observable is excluded (30.0)
00126 * .number_filter_iter - number of iterations for the estimation filter (1)
00127 * .bias_0 - (30.0)
00128 * .iono_0 - (0.03)
00129 * .trop_0 - (0.3)
00130 * .sigma_bias - process noise stddev of carrier-phase bias(ambiguity) (cycle/sqrt(s)) (1e-4)
00131 * .sigma_iono - process noise stddev of vertical ionospheric delay per 10km baseline (m/sqrt(s))
(1e-3)
00132 * .sigma_trop - process noise stddev of zenith tropospheric delay (m/sqrt(s)) (1e-4)
00133 * .sigma_acch - process noise stddev of the receiver acceleration horizontal component
(m/s2/sqrt(s)) (1e-1)
00134 * .sigma_accv - process noise stddev of the receiver acceleration vertical component (m/s2/sqrt(s))
(1e-2)
00135 * .sigma_pos - (0.0)
00136 * .code_phase_error_ratio_l1 - (100.0)
00137 * .code_phase_error_ratio_l2 - (100.0)
00138 * .code_phase_error_ratio_l5 - (100.0)
00139 * .carrier_phase_error_factor_a - (0.003)
00140 * .carrier_phase_error_factor_b - (0.003)
00141 *
00142 * .output_enabled - (true)
00143 * .rinex_output_enabled - (.output_enabled)
00144 * .gpx_output_enabled - (.output_enabled)
00145 * .geojson_output_enabled - (.output_enabled)
00146 * .kml_output_enabled - (.output_enabled)
00147 * .xml_output_enabled - (.output_enabled)
00148 * .nmea_output_enabled - (.output_enabled)
00149 * .rtcm_output_enabled - (false)
00150
00151 * .output_path - directory to which output files are written (".")
00152 * .rinex_output_path - (.output_path)
00153 * .gpx_output_path - (.output_path)
00154 * .geojson_output_path - (.output_path)
00155 * .kml_output_path - (.output_path)
00156 * .xml_output_path - (.output_path)
00157 * .nmea_output_path - (.output_path)
00158 * .rtcm_output_path - (.output_path)
00159 *
00160 * .enable_monitor - enable the PVT monitor (false)
00161 * .monitor_client_addresses - ("127.0.0.1")
00162 * .monitor_udp_port - DO NOT USE THE DEFAULT (1234)
00163 * .enable_protobuf - serialize using protocol buffers (true). Monitor.enable_protobuf if true, sets
this to true
00164 *
00165 * .enable_monitor_ephemeris - enable the ephemeris monitor (false)
00166 * .monitor_ephemeris_client_addresses - ("127.0.0.1")
00167 * .monitor_ephemeris_udp_port - DO NOT USE THE DEFAULT (1234)

```

```

00168 *
00169 * .show_local_time_zone - (false)
00170 * .enable_rx_clock_correction - (false)
00171 * .max_clock_offset_ms - (40)
00172 */
00173 class Rtklib_Pvt : public PvtInterface
00174 {
00175 public:
00176     Rtklib_Pvt(const ConfigurationInterface* configuration,
00177               const std::string& role,
00178               unsigned int in_streams,
00179               unsigned int out_streams);
00180
00181     virtual ~Rtklib_Pvt();
00182
00183     inline std::string role() override
00184     {
00185         return role_;
00186     }
00187
00188     //! Returns "RTKLIB_PVT"
00189     inline std::string implementation() override
00190     {
00191         return "RTKLIB_PVT";
00192     }
00193
00194     void clear_ephemeris() override;
00195     std::map<int, Gps_Ephemeris> get_gps_ephemeris() const override;
00196     std::map<int, Galileo_Ephemeris> get_galileo_ephemeris() const override;
00197     std::map<int, Gps_Almanac> get_gps_almanac() const override;
00198     std::map<int, Galileo_Almanac> get_galileo_almanac() const override;
00199
00200     void connect(gr::top_block_sptr top_block) override;
00201     void disconnect(gr::top_block_sptr top_block) override;
00202     gr::basic_block_sptr get_left_block() override;
00203     gr::basic_block_sptr get_right_block() override;
00204
00205     inline void reset() override
00206     {
00207         return;
00208     }
00209
00210     //! All blocks must have an item_size() function implementation
00211     inline size_t item_size() override
00212     {
00213         return sizeof(Gnss_Synchro);
00214     }
00215
00216     bool get_latest_PVT(double* longitude_deg,
00217                       double* latitude_deg,
00218                       double* height_m,
00219                       double* ground_speed_kmh,
00220                       double* course_over_ground_deg,
00221                       time_t* UTC_time) override;
00222
00223 private:
00224     rtklib_pvt_gs_sptr pvt_;
00225     rtk_t rtk{};
00226     std::string role_;
00227     unsigned int in_streams_;
00228     unsigned int out_streams_;
00229 };
00230
00231 /** \} */
00232 /** \} */
00233 #endif // GNSS_SDR_RTKLIB_PVT_H

```

## 11.244 rtklib\_pvt\_gs.h File Reference

Interface of a Position Velocity and Time computation block.

```

#include "gnss_block_interface.h"
#include "gnss_sdr_make_unique.h"
#include "gnss_synchro.h"
#include "gnss_time.h"
#include "osnma_data.h"
#include "rtklib.h"
#include <boost/date_time/gregorian/gregorian.hpp>
#include <boost/date_time/posix_time/posix_time.hpp>
#include <boost/interprocess/ipc/message_queue.hpp>

```

```

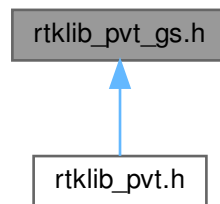
#include <gnuradio/sync_block.h>
#include <gnuradio/types.h>
#include <pmt/pmt.h>
#include <chrono>
#include <cstdint>
#include <stdint>
#include <ctime>
#include <fstream>
#include <map>
#include <memory>
#include <queue>
#include <set>
#include <string>
#include <vector>

```

Include dependency graph for rtklib\_pvt\_gs.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [rtklib\\_pvt\\_gs](#)

*This class implements a block that computes the PVT solution using the RTKLIB integrated library.*

## Typedefs

- using [rtklib\\_pvt\\_gs\\_sptr](#) = `gnss_shared_ptr<rtklib\_pvt\_gs>`

## Functions

- `rtklib_pvt_gs_sptr rtklib\_make\_pvt\_gs (uint32_t nchannels, const Pvt\_Conf &conf_, const rtk\_t &rtk)`

### 11.244.1 Detailed Description

Interface of a Position Velocity and Time computation block.

## Author

Javier Arribas, 2017. jarribas(at)cttc.es

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [rtklib\\_pvt\\_gs.h](#).

## 11.245 rtklib\_pvt\_gs.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file rtklib_pvt_gs.h
00003  * \brief Interface of a Position Velocity and Time computation block
00004  * \author Javier Arribas, 2017. jarribas(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_RTKLIB_PVT_GS_H
00018 #define GNSS_SDR_RTKLIB_PVT_GS_H
00019
00020 #include "gnss_block_interface.h"
00021 #include "gnss_sdr_make_unique.h"
00022 #include "gnss_synchro.h"
00023 #include "gnss_time.h"
00024 #include "osnma_data.h"
00025 #include "rtklib.h"
00026 #include <boost/date_time/gregorian/gregorian.hpp>
00027 #include <boost/date_time/posix_time/posix_time.hpp>
00028 #include <boost/interprocess/ipc/message_queue.hpp>
00029 #include <gnuradio/sync_block.h> // for sync_block
00030 #include <gnuradio/types.h> // for gr_vector_const_void_star
00031 #include <pmt/pmt.h> // for pmt_t
00032 #include <chrono> // for system_clock
00033 #include <cstdint> // for size_t
00034 #include <stdint> // for int32_t
00035 #include <ctime> // for time_t
00036 #include <fstream> // for std::fstream
00037 #include <map> // for map
00038 #include <memory> // for shared_ptr, unique_ptr
00039 #include <queue> // for std::queue
00040 #include <set> // for std::set
00041 #include <string> // for string
00042 #include <vector> // for vector
00043
00044 /** \addtogroup PVT
00045  * \{ */
00046 /** \addtogroup PVT_gnuradio_blocks pvt_gr_blocks
00047  * GNU Radio blocks for the computation of PVT solutions.
00048  * \{ */
00049
00050
00051 class Beidou_Dnav_Almanac;
00052 class Beidou_Dnav_Ephemeris;
00053 class Galileo_Almanac;
00054 class Galileo_Ephemeris;
00055 class Galileo_HAS_data;
00056 class Geohash;
00057 class GeoJSON_Printer;
00058 class Gps_Almanac;
00059 class Gps_Ephemeris;
00060 class Gpx_Printer;
00061 class Kml_Printer;
00062 class Monitor_Pvt_Udp_Sink;
00063 class Monitor_Ephemeris_Udp_Sink;
00064 class Nmea_Printer;
00065 class Pvt_Conf;
00066 class Rinex_Printer;
00067 class Rtcn_Printer;
00068 class An_Packet_Printer;
00069 class Has_Simple_Printer;
00070 class Rtklib_Solver;
00071 class rtklib_pvt_gs;
00072

```

```

00073 using rtklib_pvt_gs_sptr = gnss_shared_ptr<rtklib_pvt_gs>;
00074
00075 rtklib_pvt_gs_sptr rtklib_make_pvt_gs(uint32_t nchannels,
00076     const Pvt_Conf& conf_,
00077     const rtk_t& rtk);
00078
00079 /*!
00080  * \brief This class implements a block that computes the PVT solution using the RTKLIB integrated
00081  * library
00082  */
00082 class rtklib_pvt_gs : public gr::sync_block
00083 {
00084 public:
00085     ~rtklib_pvt_gs(); //!< Default destructor
00086
00087     /*!
00088      * \brief Get latest set of GPS ephemeris from PVT block
00089      */
00090     std::map<int, Gps_Ephemeris> get_gps_ephemeris_map() const;
00091
00092     /*!
00093      * \brief Get latest set of GPS almanac from PVT block
00094      */
00095     std::map<int, Gps_Almanac> get_gps_almanac_map() const;
00096
00097     /*!
00098      * \brief Get latest set of Galileo ephemeris from PVT block
00099      */
00100     std::map<int, Galileo_Ephemeris> get_galileo_ephemeris_map() const;
00101
00102     /*!
00103      * \brief Get latest set of Galileo almanac from PVT block
00104      */
00105     std::map<int, Galileo_Almanac> get_galileo_almanac_map() const;
00106
00107     /*!
00108      * \brief Get latest set of BeiDou DNAV ephemeris from PVT block
00109      */
00110     std::map<int, Beidou_Dnav_Ephemeris> get_beidou_dnav_ephemeris_map() const;
00111
00112     /*!
00113      * \brief Get latest set of BeiDou DNAV almanac from PVT block
00114      */
00115     std::map<int, Beidou_Dnav_Almanac> get_beidou_dnav_almanac_map() const;
00116
00117     /*!
00118      * \brief Clear all ephemeris information and the almanacs for GPS and Galileo
00119      */
00120     void clear_ephemeris();
00121
00122     /*!
00123      * \brief Get the latest Position WGS84 [deg], Ground Velocity, Course over Ground, and UTC Time,
00124      * if available
00125      */
00125     bool get_latest_PVT(double* longitude_deg,
00126         double* latitude_deg,
00127         double* height_m,
00128         double* ground_speed_kmh,
00129         double* course_over_ground_deg,
00130         time_t* UTC_time) const;
00131
00132     int work(int noutput_items, gr_vector_const_void_star& input_items,
00133         gr_vector_void_star& output_items); //!< PVT Signal Processing
00134
00135 private:
00136     friend rtklib_pvt_gs_sptr rtklib_make_pvt_gs(uint32_t nchannels,
00137         const Pvt_Conf& conf_,
00138         const rtk_t& rtk);
00139
00140     rtklib_pvt_gs(uint32_t nchannels,
00141         const Pvt_Conf& conf_,
00142         const rtk_t& rtk);
00143
00144     void log_source_timetag_info(double RX_time_ns, double TAG_time_ns);
00145
00146     void msg_handler_telemetry(const pmt::pmt_t& msg);
00147
00148     void msg_handler_has_data(const pmt::pmt_t& msg);
00149
00150     void msg_handler_osnma(const pmt::pmt_t& msg);
00151
00152     void initialize_and_apply_carrier_phase_offset();
00153
00154     void apply_rx_clock_offset(std::map<int, Gnss_Synchro>& observables_map,
00155         double rx_clock_offset_s);
00156
00157     void update_HAS_corrections();

```

```

00158
00159     std::map<int, Gnss_Synchro> interpolate_observables(const std::map<int, Gnss_Synchro>&
observables_map_t0,
00160         const std::map<int, Gnss_Synchro>& observables_map_t1,
00161         double rx_time_s);
00162
00163     inline std::time_t convert_to_time_t(const boost::posix_time::ptime pt) const
00164     {
00165         return (pt - boost::posix_time::ptime(boost::gregorian::date(1970, 1, 1))).total_seconds();
00166     }
00167
00168     std::vector<std::string> split_string(const std::string& s, char delim) const;
00169
00170     bool send_ttfmsg(double ttf) const;
00171     bool save_gnss_synchro_map_xml(const std::string& file_name); // debug helper function
00172     bool load_gnss_synchro_map_xml(const std::string& file_name); // debug helper function
00173
00174     std::fstream d_log_timetag_file;
00175
00176     std::shared_ptr<Rtklib_Solver> d_internal_pvt_solver;
00177     std::shared_ptr<Rtklib_Solver> d_user_pvt_solver;
00178
00179     std::unique_ptr<boost::interprocess::message_queue> d_mq;
00180
00181     std::unique_ptr<Rinex_Printer> d_rp;
00182     std::unique_ptr<Kml_Printer> d_kml_dump;
00183     std::unique_ptr<Gpx_Printer> d_gpx_dump;
00184     std::unique_ptr<Nmea_Printer> d_nmea_printer;
00185     std::unique_ptr<GeoJSON_Printer> d_geojson_printer;
00186     std::unique_ptr<Rtcm_Printer> d_rtcm_printer;
00187     std::unique_ptr<Monitor_Pvt_Udp_Sink> d_udp_sink_ptr;
00188     std::unique_ptr<Monitor_Ephemeris_Udp_Sink> d_eph_udp_sink_ptr;
00189     std::unique_ptr<Has_Simple_Printer> d_has_simple_printer;
00190     std::unique_ptr<An_Packet_Printer> d_an_printer;
00191
00192     std::chrono::time_point<std::chrono::system_clock> d_start;
00193     std::chrono::time_point<std::chrono::system_clock> d_end;
00194
00195     std::string d_queue_name;
00196     std::string d_dump_filename;
00197     std::string d_xml_base_path;
00198     std::string d_local_time_str;
00199
00200     std::vector<bool> d_channel_initialized;
00201     std::vector<double> d_initial_carrier_phase_offset_estimation_rads;
00202
00203     std::map<int, Gnss_Synchro> d_gnss_observables_map;
00204     std::map<int, Gnss_Synchro> d_gnss_observables_map_t0;
00205     std::map<int, Gnss_Synchro> d_gnss_observables_map_t1;
00206     std::map<uint32_t, std::set<uint32_t>> d_auth_nav_data_map;
00207
00208     std::queue<GnssTime> d_TimeChannelTagTimestamps;
00209
00210     boost::posix_time::time_duration d_utc_diff_time;
00211     std::unique_ptr<Geohash> d_geohash;
00212
00213     size_t d_gps_ephemeris_spstr_type_hash_code;
00214     size_t d_gps_iono_spstr_type_hash_code;
00215     size_t d_gps_utc_model_spstr_type_hash_code;
00216     size_t d_gps_cnav_ephemeris_spstr_type_hash_code;
00217     size_t d_gps_cnav_iono_spstr_type_hash_code;
00218     size_t d_gps_cnav_utc_model_spstr_type_hash_code;
00219     size_t d_gps_almanac_spstr_type_hash_code;
00220     size_t d_galileo_ephemeris_spstr_type_hash_code;
00221     size_t d_galileo_iono_spstr_type_hash_code;
00222     size_t d_galileo_utc_model_spstr_type_hash_code;
00223     size_t d_galileo_almanac_helper_spstr_type_hash_code;
00224     size_t d_galileo_almanac_spstr_type_hash_code;
00225     size_t d_glonass_gnav_ephemeris_spstr_type_hash_code;
00226     size_t d_glonass_gnav_utc_model_spstr_type_hash_code;
00227     size_t d_glonass_gnav_almanac_spstr_type_hash_code;
00228     size_t d_beidou_dnav_ephemeris_spstr_type_hash_code;
00229     size_t d_beidou_dnav_iono_spstr_type_hash_code;
00230     size_t d_beidou_dnav_utc_model_spstr_type_hash_code;
00231     size_t d_beidou_dnav_almanac_spstr_type_hash_code;
00232     size_t d_galileo_has_data_spstr_type_hash_code;
00233
00234     double d_rinex_version;
00235     double d_rx_time;
00236     uint64_t d_local_counter_ms;
00237     uint64_t d_timestamp_rx_clock_offset_correction_msg_ms;
00238
00239     int32_t d_rinexobs_rate_ms;
00240     int32_t d_rtcm_MT1045_rate_ms; // Galileo Broadcast Ephemeris
00241     int32_t d_rtcm_MT1019_rate_ms; // GPS Broadcast Ephemeris (orbits)
00242     int32_t d_rtcm_MT1020_rate_ms; // GLONASS Broadcast Ephemeris (orbits)
00243     int32_t d_rtcm_MT1077_rate_ms; // The type 7 Multiple Signal Message format for the USA's GPS

```

```

    system, popular
00244     int32_t d_rtcmt_M1087_rate_ms; // GLONASS MSM7. The type 7 Multiple Signal Message format for the
Russian GLONASS system
00245     int32_t d_rtcmt_M1097_rate_ms; // Galileo MSM7. The type 7 Multiple Signal Message format for
Europe's Galileo system
00246     int32_t d_rtcmt_MSM_rate_ms;
00247     int32_t d_kml_rate_ms;
00248     int32_t d_gpx_rate_ms;
00249     int32_t d_geojson_rate_ms;
00250     int32_t d_nmea_rate_ms;
00251     int32_t d_an_rate_ms;
00252     int32_t d_output_rate_ms;
00253     int32_t d_display_rate_ms;
00254     int32_t d_report_rate_ms;
00255     int32_t d_max_obs_block_rx_clock_offset_ms;
00256
00257     uint32_t d_nchannels;
00258     uint32_t d_type_of_rx;
00259     uint32_t d_observable_interval_ms;
00260     uint32_t d_pvt_errors_counter;
00261
00262     bool d_dump;
00263     bool d_dump_mat;
00264     bool d_rinex_output_enabled;
00265     bool d_geojson_output_enabled;
00266     bool d_gpx_output_enabled;
00267     bool d_kml_output_enabled;
00268     bool d_nmea_output_file_enabled;
00269     bool d_rtcmt_enabled;
00270     bool d_first_fix;
00271     bool d_xml_storage;
00272     bool d_flag_monitor_pvt_enabled;
00273     bool d_flag_monitor_ephemeris_enabled;
00274     bool d_show_local_time_zone;
00275     bool d_enable_rx_clock_correction;
00276     bool d_enable_has_messages;
00277     bool d_an_printer_enabled;
00278     bool d_log_timetag;
00279     bool d_use_has_corrections;
00280     bool d_use_unhealthy_sats;
00281     bool d_osnma_strict;
00282 };
00283
00284
00285 /** \} */
00286 /** \} */
00287 #endif // GNSS_SDR_RTKLIB_PVT_GS_H

```

## 11.246 an\_packet\_printer.h File Reference

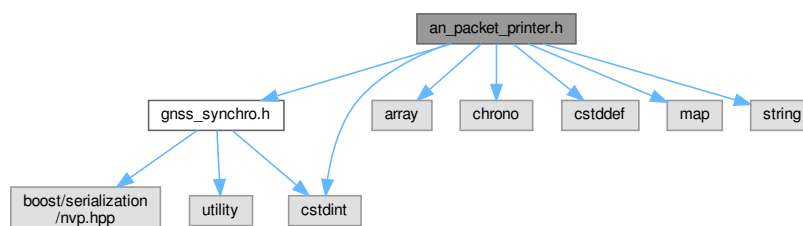
Interface of a class that prints PVT solutions in a serial device following a custom version of the Advanced Navigation Packet Protocol.

```

#include "gnss_synchro.h"
#include <array>
#include <chrono>
#include <cstdint>
#include <cstdint>
#include <map>
#include <string>

```

Include dependency graph for an\_packet\_printer.h:





**Classes**

- struct [sdr\\_gnss\\_packet\\_t](#)
- struct [an\\_packet\\_t](#)
- class [An\\_Packet\\_Printer](#)

*class that prints PVT solutions in a serial device following a custom version of the Advanced Navigation Packet Protocol.*

**11.246.1 Detailed Description**

Interface of a class that prints PVT solutions in a serial device following a custom version of the Advanced Navigation Packet Protocol.

**Author**

Carles Fernandez-Prades, 2021. [cfernandez\(at\)cttc.es](mailto:cfernandez@cttc.es)

Miguel Angel Gomez Lopez, 2021. [gomezlma\(at\)inta.es](mailto:gomezlma@inta.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [an\\_packet\\_printer.h](#).

---

**11.247 an\_packet\_printer.h**

[Go to the documentation of this file.](#)

```

00001  /*!
00002   * \file an_packet_printer.h
00003   * \brief Interface of a class that prints PVT solutions in a serial device
00004   * \brief following a custom version of the Advanced Navigation Packet Protocol
00005   * \author Carles Fernandez-Prades, 2021. cfernandez(at)cttc.es
00006   * \author Miguel Angel Gomez Lopez, 2021. gomezlma(at)inta.es
00007   *
00008   *
00009   * -----
00010   *
00011   * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00012   * This file is part of GNSS-SDR.
00013   *
00014   * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)
00015   * SPDX-License-Identifier: GPL-3.0-or-later
00016   *
00017   * -----
00018   */
00019
00020
00021 #ifndef GNSS_SDR_AN_PACKET_PRINTER_H
00022 #define GNSS_SDR_AN_PACKET_PRINTER_H
00023
00024 #include "gnss_synchro.h"
00025 #include <array>
00026 #include <chrono>
00027 #include <cstring>
00028 #include <stdint>
00029 #include <map>
00030 #include <string>
00031
00032 /** \addtogroup PVT
00033  * \{ */
00034 /** \addtogroup PVT_libs
00035  * \{ */
00036
00037 class Rtklib_Solver;
00038
00039 struct sdr_gnss_packet_t
00040 {
00041     uint8_t nsvfix;           // number of sats used in PVT fix
00042     uint8_t gps_satellites;   // number of tracked GPS satellites
00043     uint8_t galileo_satellites; // number of tracked Galileo satellites
00044     uint32_t microseconds;    // from start of receiver operation
00045     double latitude;           // in [rad]
00046     double longitude;         // in [rad]
00047     double height;            // in [m]
00048     float velocity[3];        // North, East, Down, in [m/s]
00049
00050     struct
00051     {

```

```

00052         uint8_t prn;          // PRN ID. Galileo sats expressed as PRN + 100
00053         uint8_t snr;          // in [dB-Hz]
00054         int16_t doppler;      // in [Hz], saturates at +32767 / -32768 Hz
00055     } sats[6];
00056
00057     uint32_t reserved;
00058     uint16_t status;
00059 };
00060
00061
00062 struct an_packet_t
00063 {
00064     uint8_t header[4];
00065     uint8_t data[73]; // SDR_GNSS_PACKET_LENGTH
00066 };
00067
00068
00069 /*!
00070  * \brief class that prints PVT solutions in a serial device following a custom
00071  * version of the Advanced Navigation Packet Protocol.
00072  */
00073 class An_Packet_Printer
00074 {
00075 public:
00076     /*!
00077      * \brief Default constructor.
00078      */
00079     explicit An_Packet_Printer(const std::string& an_dump_devname);
00080
00081     /*!
00082      * \brief Default destructor.
00083      */
00084     ~An_Packet_Printer();
00085
00086     /*!
00087      * \brief Print AN packet to the initialized device.
00088      */
00089     bool print_packet(const Rtklib_Solver* const pvt_data, const std::map<int, Gnss_Synchro>&
gnss_observables_map);
00090
00091     /*!
00092      * \brief Close serial port. Also done in the destructor, this is only
00093      * for testing.
00094      */
00095     void close_serial() const;
00096
00097 private:
00098     const std::array<uint16_t, 256> d_crc16_table = {
00099         {0x0000, 0x1021, 0x2042, 0x3063, 0x4084, 0x50a5, 0x60c6, 0x70e7, 0x8108, 0x9129, 0xa14a,
00100         0xb16b, 0xc18c, 0xd1ad, 0xe1ce, 0xf1ef, 0x1231, 0x0210, 0x3273,
00101         0x2252, 0x52b5, 0x4294, 0x72f7, 0x62d6, 0x9339, 0x8318, 0xb37b, 0xa35a, 0xd3bd, 0xc39c,
00102         0xf3ff, 0xe3de, 0xd462, 0xc443, 0xb420, 0xa401, 0x94e6, 0x84c7, 0x74a4, 0x6485, 0x546a, 0x444b,
00103         0x3428,
00104         0x9509, 0xe5ee, 0xf5cf, 0xc5ac, 0xd58d, 0x6553, 0x5572, 0x4511, 0x3530, 0x25d7, 0x15f6,
00105         0x0595, 0xf5b4, 0xe57b, 0xd57a, 0xc519, 0xb538, 0xa55f, 0x957e, 0x859d, 0x75bc, 0x658c, 0x55e5,
00106         0x4586,
00107         0x78a7, 0x0840, 0x1861, 0x2802, 0x3823, 0xc9cc, 0xd9ed, 0xe98e, 0xf9af, 0x8948, 0x9969,
00108         0xa90a, 0xb92b, 0x5af5, 0x4ad4, 0x7ab7, 0x6a96, 0x1a71, 0x0a50, 0x3a33, 0x2a12, 0xdbfd, 0xcbbdc,
00109         0xfbbf,
00110         0xeb9e, 0x9b79, 0x8b58, 0xbb3b, 0xab1a, 0x6ca6, 0x7c87, 0x4ce4, 0x5cc5, 0x2c22, 0x3c03,
00111         0x0c60, 0x1c41, 0xedae, 0xfd8f, 0xcdec, 0xddcd, 0xad2a, 0xbd0b, 0x8d68, 0x9d49, 0xe997, 0x6eb6,
00112         0x5ed5,
00113         0x4ef4, 0x3e13, 0x2e32, 0x1e51, 0x0e70, 0xff9f, 0xefbe, 0xdfdd, 0xcffc, 0xbfb1, 0xaf3a,
00114         0x9f59, 0x8f78, 0x9188, 0x81a9, 0xb1ca, 0xaleb, 0xd10c, 0xc12d, 0xf14e, 0xe16f, 0xd180, 0xc0a1,
00115         0x30c2,
00116         0x20e3, 0x5004, 0x4025, 0x7046, 0x6067, 0x83b9, 0x9398, 0xa3fb, 0xb3da, 0xc33d, 0xd31c,
00117         0xe37f, 0xf35e, 0x02b1, 0x1290, 0x22f3, 0x32d2, 0x4235, 0x5214, 0x6277, 0x7256, 0x823a, 0x921b,
00118         0x029a,
00119         0x8589, 0xf56e, 0xe54f, 0xd52c, 0xc50d, 0x34e2, 0x24c3, 0x14a0, 0x0481, 0x7466, 0x6447,
00120         0x5424, 0x4405, 0xa7db, 0xb7fa, 0x8799, 0x97b8, 0xe75f, 0xf77e, 0xc71d, 0xd73c, 0xa6d3, 0xb6f2,
00121         0x0691,
00122         0x16b0, 0x0657, 0x7676, 0x4615, 0x5634, 0xd94c, 0xc96d, 0xf90e, 0xe92f, 0x99c8, 0x89e9,
00123         0xb98a, 0xa9ab, 0x5844, 0x4865, 0x7806, 0x6827, 0x18c0, 0x08e1, 0x3882, 0x28a3, 0xcb7d, 0xdb5c,
00124         0xeb3f,
00125         0xfble, 0x8bf9, 0x9bd8, 0xabbb, 0xbb9a, 0x4a75, 0x5a54, 0x6a37, 0x7a16, 0x0af1, 0x1ad0,
00126         0x2ab3, 0x3a92, 0xfd2e, 0xed0f, 0xdd6c, 0xcd4d, 0xbdaa, 0xad8b, 0x9de8, 0x8dc9, 0x7c26, 0x6c07,
00127         0x5c64,
00128         0x4c45, 0x3ca2, 0x2c83, 0x1ce0, 0x0cc1, 0xef1f, 0xff3e, 0xcf5d, 0xdf7c, 0xaf9b, 0xbfba,
00129         0x8fd9, 0x9ff8, 0x6e17, 0x7e36, 0x4e55, 0x5e74, 0x2e93, 0x3eb2, 0x0ed1, 0x1ef0}};
00130
00131     const size_t SDR_GNSS_PACKET_LENGTH = 73;
00132     const uint8_t SDR_GNSS_PACKET_ID = 201;
00133
00134     int init_serial(const std::string& serial_device);
00135     void update_sdr_gnss_packet(sdr_gnss_packet_t* _packet, const Rtklib_Solver* const pvt, const
std::map<int, Gnss_Synchro>& gnss_observables_map) const;
00136     void encode_gnss_cttc_packet(sdr_gnss_packet_t* sdr_gnss_packet, an_packet_t* _packet) const;

```

```

00117     uint16_t calculate_crc16(const void* data, uint16_t length) const;
00118     uint8_t calculate_header_lrc(const uint8_t* data) const;
00119     void an_packet_encode(an_packet_t* an_packet) const;
00120     void encode_sdr_gnss_packet(sdr_gnss_packet_t* sdr_gnss_packet, an_packet_t* _packet) const;
00121     void LSB_bytes_to_array(void* _in, int offset, uint8_t* _out, uint8_t var_size) const;
00122
00123     std::chrono::time_point<std::chrono::system_clock> d_start;
00124     std::string d_an_devname;
00125     int d_an_dev_descriptor; // serial device descriptor (i.e. COM port)
00126 };
00127
00128 /** \} */
00129 /** \} */
00130 #endif // GNSS_SDR_AN_PACKET_PRINTER_H

```

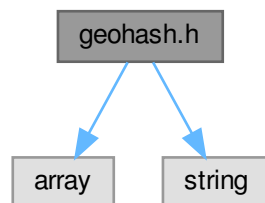
## 11.248 geohash.h File Reference

Interface of a class that encodes / decodes geohashes.

```
#include <array>
```

```
#include <string>
```

Include dependency graph for geohash.h:



### Classes

- class [Geohash](#)

Class for geohash encoding / decoding See <https://en.wikipedia.org/wiki/Geohash>.

### 11.248.1 Detailed Description

Interface of a class that encodes / decodes geohashes.

#### Author

Carles Fernandez-Prades, 2023. cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2023 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [geohash.h](#).

---

## 11.249 geohash.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file geohash.h
00003  * \brief Interface of a class that encodes / decodes geohashes
00004  * \author Carles Fernandez-Prades, 2023. cfernandez(at)cttc.es
00005  *
00006  *
00007  * -----
00008  *

```

```

00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2023 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018
00019 #ifndef GNSS_SDR_GEOHASH_H
00020 #define GNSS_SDR_GEOHASH_H
00021
00022 #include <array>
00023 #include <string>
00024
00025 /** \addtogroup PVT
00026  * \{ */
00027 /** \addtogroup PVT_libs
00028  * \{ */
00029
00030 /*!
00031  * \brief Class for geohash encoding / decoding
00032  * See https://en.wikipedia.org/wiki/Geohash
00033  */
00034 class Geohash
00035 {
00036 public:
00037     Geohash() = default;
00038
00039     /**
00040      * Encodes latitude/longitude to geohash, either to specified precision or
00041      * to automatically evaluated precision.
00042      *
00043      * @param {double} lat - Latitude in degrees.
00044      * @param {double} lon - Longitude in degrees.
00045      * @param {int} [precision] - Number of characters in resulting geohash.
00046      * @returns {string} Geohash of supplied latitude/longitude.
00047      * @throws Invalid geohash.
00048      */
00049
00050     std::string encode(double lat, double lon, int precision = -1) const;
00051
00052     /**
00053      * Decode geohash to latitude/longitude (location is approximate centre of
00054      * geohash cell, to reasonable precision).
00055      *
00056      * @param {string} geohash - Geohash string to be converted to
00057      * latitude/longitude.
00058      * @returns {lat, lon} (Center of) geohashed location.
00059      * @throws Invalid geohash.
00060      */
00061
00062     std::array<double, 2> decode(std::string geohash) const;
00063
00064 private:
00065     /**
00066      * Returns SW/NE latitude/longitude bounds of specified geohash.
00067      */
00068     std::array<double, 4> bounds(std::string geohash) const;
00069     std::string base32{"0123456789bcdefghjkmnpqrstuvxyz"};
00070 };
00071
00072 /** \} */
00073 /** \} */
00074 #endif // GNSS_SDR_GEOHASH_H

```

## 11.250 geojson\_printer.h File Reference

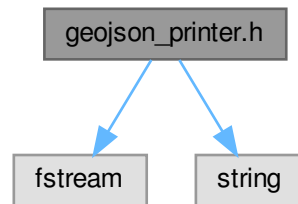
Interface of a class that prints PVT solutions in GeoJSON format.

```

#include <fstream>
#include <string>

```

Include dependency graph for geojson\_printer.h:



## Classes

- class [GeoJSON\\_Printer](#)  
*Prints PVT solutions in GeoJSON format file.*

### 11.250.1 Detailed Description

Interface of a class that prints PVT solutions in GeoJSON format.

#### Author

Carles Fernandez-Prades, 2015. cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [geojson\\_printer.h](#).

---

## 11.251 geojson\_printer.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file geojson_printer.h
00003  * \brief Interface of a class that prints PVT solutions in GeoJSON format
00004  * \author Carles Fernandez-Prades, 2015. cfernandez(at)cttc.es
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018
00019 #ifndef GNSS_SDR_GEOJSON_PRINTER_H
00020 #define GNSS_SDR_GEOJSON_PRINTER_H
00021
00022
00023 #include <fstream>
00024 #include <string>
00025
00026 /** \addtogroup PVT
00027  * \{ */
00028 /** \addtogroup PVT_libs
00029  * \{ */
00030
00031
00032 class Pvt_Solution;
00033

```

```

00034  /*!
00035   * \brief Prints PVT solutions in GeoJSON format file
00036   *
00037   * See https://tools.ietf.org/html/rfc7946
00038   */
00039  class GeoJSON_Printer
00040  {
00041  public:
00042      explicit GeoJSON_Printer(const std::string& base_path = ".");
00043      ~GeoJSON_Printer();
00044      bool set_headers(const std::string& filename, bool time_tag_name = true);
00045      bool print_position(const Pvt_Solution* const position);
00046      bool close_file();
00047  private:
00048      std::ofstream geojson_file;
00049      std::string filename_;
00050      std::string geojson_base_path;
00051      bool first_pos;
00052  };
00053  };
00054
00055
00056  /** \} */
00057  /** \} */
00058  #endif // GNSS_SDR_GEOJSON_PRINTER_H

```

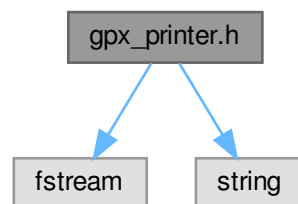
## 11.252 gpx\_printer.h File Reference

Interface of a class that prints PVT information to a gpx file.

```
#include <fstream>
```

```
#include <string>
```

Include dependency graph for gpx\_printer.h:



### Classes

- class [Gpx\\_Printer](#)

*Prints PVT information to GPX format file.*

### 11.252.1 Detailed Description

Interface of a class that prints PVT information to a gpx file.

#### Author

Álvaro Cebrián Juan, 2018. [acebrianjuan\(at\)gmail.com](mailto:acebrianjuan(at)gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [gpx\\_printer.h](#).

---

## 11.253 gpx\_printer.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file gpx_printer.h
00003  * \brief Interface of a class that prints PVT information to a gpx file
00004  * \author Álvaro Cebrián Juan, 2018. acebrianjuan(at)gmail.com
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018
00019 #ifndef GNSS_SDR_GPX_PRINTER_H
00020 #define GNSS_SDR_GPX_PRINTER_H
00021
00022
00023 #include <fstream>
00024 #include <string>
00025
00026 /** \addtogroup PVT
00027  * \{ */
00028 /** \addtogroup PVT_libs
00029  * \{ */
00030
00031
00032 class Pvt_Solution;
00033
00034 /*!
00035  * \brief Prints PVT information to GPX format file
00036  *
00037  * See https://www.topografix.com/gpx.asp
00038  */
00039 class Gpx_Printer
00040 {
00041 public:
00042     explicit Gpx_Printer(const std::string& base_path = ".");
00043     ~Gpx_Printer();
00044     bool set_headers(const std::string& filename, bool time_tag_name = true);
00045     bool print_position(const Pvt_Solution* const position);
00046     bool close_file();
00047
00048 private:
00049     std::ofstream gpx_file;
00050     std::string gpx_filename;
00051     std::string indent;
00052     std::string gpx_base_path;
00053     bool positions_printed;
00054 };
00055
00056
00057 /** \} */
00058 /** \} */
00059 #endif // GNSS_SDR_GPX_PRINTER_H

```

## 11.254 has\_simple\_printer.h File Reference

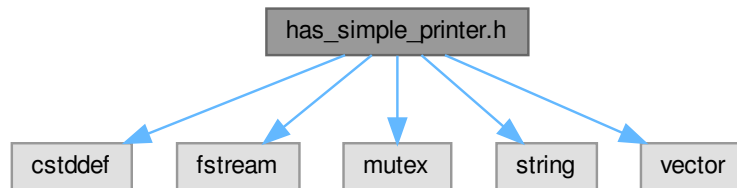
Interface of a class that prints HAS messages content in a txt file.

```

#include <cstdint>
#include <fstream>
#include <mutex>
#include <string>
#include <vector>

```

Include dependency graph for `has_simple_printer.h`:



## Classes

- class [Has\\_Simple\\_Printer](#)

*Prints HAS messages content in a txt file. See HAS-SIS-ICD for a message description.*

## 11.254.1 Detailed Description

Interface of a class that prints HAS messages content in a txt file.

### Author

Carles Fernandez-Prades, 2021. [cfernandez\(at\)cttc.es](mailto:cfernandez@cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [has\\_simple\\_printer.h](#).

---

## 11.255 has\_simple\_printer.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file has_simple_printer.h
00003  * \brief Interface of a class that prints HAS messages content in a txt file.
00004  * \author Carles Fernandez-Prades, 2021. cfernandez(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_HAS_SIMPLE_PRINTER_H
00018 #define GNSS_SDR_HAS_SIMPLE_PRINTER_H
00019
00020 #include <cstdint> // for size_t
00021 #include <fstream> // for std::ofstream
00022 #include <mutex> // for std::mutex
00023 #include <string> // for std::string
00024 #include <vector> // for std::vector
00025
00026 /** \addtogroup PVT
00027  * \{ */
00028 /** \addtogroup PVT_libs
00029  * \{ */
00030
00031 class Galileo_HAS_data;
00032
00033 /*!
00034  * \brief Prints HAS messages content in a txt file. See HAS-SIS-ICD for a
00035  * message description.
  
```



```

00036  */
00037  class Has_Simple_Printer
00038  {
00039  public:
00040      explicit Has_Simple_Printer(const std::string& base_path = std::string("."), const std::string&
          filename = std::string("HAS_Messages"), bool time_tag_name = true);
00041      ~Has_Simple_Printer();
00042      bool print_message(const Galileo_HAS_data* const has_data);
00043
00044  private:
00045      template <class T>
00046          std::string print_vector(const std::vector<T>& vec, float scale_factor = 1) const;
00047
00048      template <class T>
00049          std::string print_vector_binary(const std::vector<T>& vec, size_t bit_length) const;
00050
00051      template <class T>
00052          std::string print_matrix(const std::vector<std::vector<T>>& mat, const std::string& filler, float
          scale_factor = 1) const;
00053
00054          std::string print_vector_string(const std::vector<std::string>& vec) const;
00055
00056          bool close_file();
00057
00058          std::mutex d_mutex;
00059          std::ofstream d_has_file;
00060          std::string d_has_filename;
00061          std::string d_has_base_path;
00062          bool d_data_printed;
00063  };
00064
00065  /** \} */
00066  /** \} */
00067  #endif // GNSS_SDR_HAS_SIMPLE_PRINTER_H

```

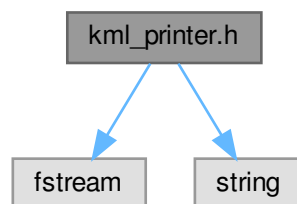
## 11.256 kml\_printer.h File Reference

Interface of a class that prints PVT information to a kml file.

```
#include <fstream>
```

```
#include <string>
```

Include dependency graph for kml\_printer.h:



### Classes

- class [Kml\\_Printer](#)

*Prints PVT information to OGC KML format file (can be viewed with Google Earth)*

### 11.256.1 Detailed Description

Interface of a class that prints PVT information to a kml file.

**Author**

Javier Arribas, 2011. jarribas(at)cttc.es Álvaro Cebrián Juan, 2018. acebrianjuan(at)gmail.com

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [kml\\_printer.h](#).

**11.257 kml\_printer.h**

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file kml_printer.h
00003  * \brief Interface of a class that prints PVT information to a kml file
00004  * \author Javier Arribas, 2011. jarribas(at)cttc.es
00005  *         Álvaro Cebrián Juan, 2018. acebrianjuan(at)gmail.com
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018
00019 #ifndef GNSS_SDR_KML_PRINTER_H
00020 #define GNSS_SDR_KML_PRINTER_H
00021
00022 #include <fstream> // for ofstream
00023 #include <string>
00024
00025 /** \addtogroup PVT
00026  * \{ */
00027 /** \addtogroup PVT_libs
00028  * \{ */
00029
00030
00031 class Pvt_Solution;
00032
00033 /*!
00034 * \brief Prints PVT information to OGC KML format file (can be viewed with Google Earth)
00035 *
00036 * See https://www.opengeospatial.org/standards/kml
00037 */
00038 class Kml_Printer
00039 {
00040 public:
00041     explicit Kml_Printer(const std::string& base_path = std::string("."));
00042     ~Kml_Printer();
00043     bool set_headers(const std::string& filename, bool time_tag_name = true);
00044     bool print_position(const Pvt_Solution* const position);
00045     bool close_file();
00046
00047 private:
00048     std::ofstream kml_file;
00049     std::ofstream tmp_file;
00050     std::string kml_filename;
00051     std::string kml_base_path;
00052     std::string tmp_file_str;
00053     std::string indent;
00054     unsigned int point_id;
00055     bool positions_printed;
00056 };
00057
00058
00059 /** \} */
00060 /** \} */
00061 #endif // GNSS_SDR_KML_PRINTER_H

```

**11.258 monitor\_ephemeris\_udp\_sink.h File Reference**

Interface of a class that sends serialized [Gps\\_Ephemeris](#) and [Galileo\\_Ephemeris](#) objects over udp to one or multiple endpoints.

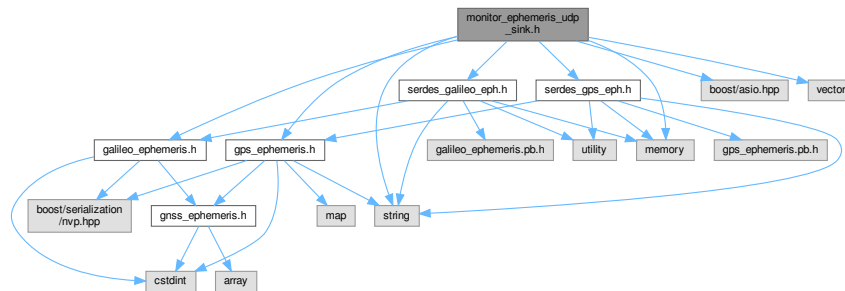
```

#include "galileo_ephemeris.h"
#include "gps_ephemeris.h"

```

```
#include "serdes_galileo_eph.h"
#include "serdes_gps_eph.h"
#include <boost/asio.hpp>
#include <memory>
#include <string>
#include <vector>
```

Include dependency graph for monitor\_ephemeris\_udp\_sink.h:



## Classes

- class [Monitor\\_Ephemeris\\_Udp\\_Sink](#)

## Typedefs

- using [b\\_io\\_context](#) = boost::asio::io\_service

### 11.258.1 Detailed Description

Interface of a class that sends serialized [Gps\\_Ephemeris](#) and [Galileo\\_Ephemeris](#) objects over udp to one or multiple endpoints.

#### Author

Javier Arribas, 2021. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [monitor\\_ephemeris\\_udp\\_sink.h](#).

---

## 11.259 monitor\_ephemeris\_udp\_sink.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file monitor_ephemeris_udp_sink.h
00003  * \brief Interface of a class that sends serialized Gps_Ephemeris and
00004  * Galileo_Ephemeris objects over udp to one or multiple endpoints.
00005  * \author Javier Arribas, 2021. jarribas(at)cttc.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_MONITOR_EPHEMERIS_UDP_SINK_H
00019 #define GNSS_SDR_MONITOR_EPHEMERIS_UDP_SINK_H
00020
```

```

00021 #include "galileo_ephemeris.h"
00022 #include "gps_ephemeris.h"
00023 #include "serdes_galileo_eph.h"
00024 #include "serdes_gps_eph.h"
00025 #include <boost/asio.hpp>
00026 #include <memory>
00027 #include <string>
00028 #include <vector>
00029
00030 /** \addtogroup PVT
00031  * \{ */
00032 /** \addtogroup PVT_libs
00033  * \{ */
00034
00035
00036 #if USE_BOOST_ASIO_IO_CONTEXT
00037 using b_io_context = boost::asio::io_context;
00038 #else
00039 using b_io_context = boost::asio::io_service;
00040 #endif
00041
00042 class Monitor_Ephemeris_Udp_Sink
00043 {
00044 public:
00045     Monitor_Ephemeris_Udp_Sink(const std::vector<std::string>& addresses, const uint16_t& port, bool
    protobuf_enabled);
00046     bool write_gps_ephemeris(const std::shared_ptr<Gps_Ephemeris>& monitor_gps_eph);
00047     bool write_galileo_ephemeris(const std::shared_ptr<Galileo_Ephemeris>& monitor_gal_eph);
00048 private:
00049     Serdes_Galileo_Eph serdes_gal;
00050     Serdes_Gps_Eph serdes_gps;
00051     b_io_context io_context;
00052     boost::asio::ip::udp::socket socket;
00053     std::vector<boost::asio::ip::udp::endpoint> endpoints;
00054     boost::system::error_code error;
00055     bool use_protobuf;
00056 };
00057
00058
00059
00060 /** \} */
00061 /** \} */
00062 #endif // GNSS_SDR_MONITOR_EPHEMERIS_UDP_SINK_H

```

## 11.260 monitor\_pvt.h File Reference

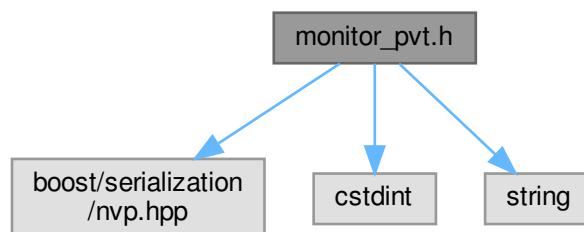
Interface of the [Monitor\\_Pvt](#) class.

```
#include <boost/serialization/nvp.hpp>
```

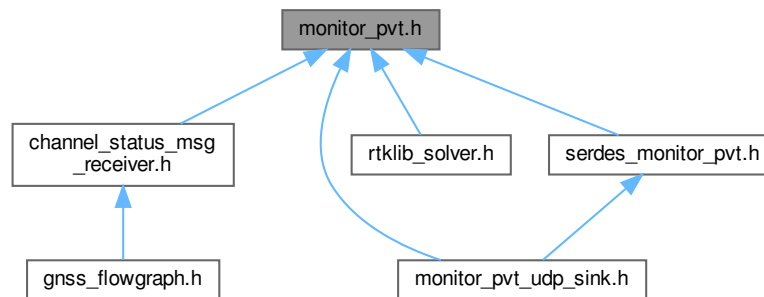
```
#include <cstdint>
```

```
#include <string>
```

Include dependency graph for monitor\_pvt.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Monitor\\_Pvt](#)

*This class contains parameters and outputs of the PVT block.*

### 11.260.1 Detailed Description

Interface of the [Monitor\\_Pvt](#) class.

Author

#### 11.260.1.1 Álvaro Cebrián Juan, 2019. [acebrianjuan\(at\)gmail.com](mailto:acebrianjuan(at)gmail.com)

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [monitor\\_pvt.h](#).

## 11.261 monitor\_pvt.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file monitor_pvt.h
00003  * \brief Interface of the Monitor_Pvt class
00004  * \author
00005  * Álvaro Cebrián Juan, 2019. acebrianjuan\(at\)gmail.com
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_MONITOR_PVT_H
00018 #define GNSS_SDR_MONITOR_PVT_H
00019
00020 #include <boost/serialization/nvp.hpp>
00021 #include <stdint>
00022 #include <string>
00023
00024 /** \addtogroup PVT
00025  * \{ */
00026 /** \addtogroup PVT_libs
00027  * \{ */
00028

```

```

00029
00030 /*!
00031 * \brief This class contains parameters and outputs of the PVT block
00032 */
00033 class Monitor_Pvt
00034 {
00035 public:
00036     // TOW
00037     uint32_t TOW_at_current_symbol_ms;
00038     // WEEK
00039     uint32_t week;
00040     // PVT GPS time
00041     double RX_time;
00042     // User clock offset [s]
00043     double user_clk_offset;
00044
00045     // ECEF POS X,Y,X [m] + ECEF VEL X,Y,X [m/s] (6 x double)
00046     double pos_x;
00047     double pos_y;
00048     double pos_z;
00049     double vel_x;
00050     double vel_y;
00051     double vel_z;
00052
00053     // position variance/covariance (m^2) {c_xx,c_yy,c_zz,c_xy,c_yz,c_zx} (6 x double)
00054     double cov_xx;
00055     double cov_yy;
00056     double cov_zz;
00057     double cov_xy;
00058     double cov_yz;
00059     double cov_zx;
00060
00061     // GEO user position Latitude [deg]
00062     double latitude;
00063     // GEO user position Longitude [deg]
00064     double longitude;
00065     // GEO user position Height [m]
00066     double height;
00067     // East, Nord, Up (ENU) Velocity [m/s]
00068     double vel_e;
00069     double vel_n;
00070     double vel_u;
00071
00072     // Course Over Ground (COG) [deg]
00073     double cog;
00074
00075     // Galileo HAS status: 1- HAS messages decoded and applied, 0 - HAS not available
00076     uint32_t galhas_status;
00077
00078     // NUMBER OF VALID SATS
00079     uint8_t valid_sats;
00080     // RTKLIB solution status
00081     uint8_t solution_status;
00082     // RTKLIB solution type (0:xyz-ecef,1:enu-baseline)
00083     uint8_t solution_type;
00084     // AR ratio factor for validation
00085     float AR_ratio_factor;
00086     // AR ratio threshold for validation
00087     float AR_ratio_threshold;
00088
00089     // GDOP / PDOP/ HDOP/ VDOP
00090     double gdop;
00091     double pdop;
00092     double hdop;
00093     double vdop;
00094
00095     // User clock drift [ppm]
00096     double user_clk_drift_ppm;
00097
00098     // PVT UTC Time (rfc 3339 datetime string)
00099     std::string utc_time;
00100
00101     std::string geohash; // See https://en.wikipedia.org/wiki/Geohash
00102
00103     /*!
00104     * \brief This member function serializes and restores
00105     * Monitor_Pvt objects from a byte stream.
00106     */
00107     template <class Archive>
00108
00109     void serialize(Archive& ar, const unsigned int version)
00110     {
00111         if (version)
00112         {
00113             };
00114
00115         ar & BOOST_SERIALIZATION_NVP(TOW_at_current_symbol_ms);

```

```

00116         ar& BOOST_SERIALIZATION_NVP (week);
00117         ar& BOOST_SERIALIZATION_NVP (RX_time);
00118         ar& BOOST_SERIALIZATION_NVP (user_clk_offset);
00119
00120         ar& BOOST_SERIALIZATION_NVP (pos_x);
00121         ar& BOOST_SERIALIZATION_NVP (pos_y);
00122         ar& BOOST_SERIALIZATION_NVP (pos_z);
00123         ar& BOOST_SERIALIZATION_NVP (vel_x);
00124         ar& BOOST_SERIALIZATION_NVP (vel_y);
00125         ar& BOOST_SERIALIZATION_NVP (vel_z);
00126
00127         ar& BOOST_SERIALIZATION_NVP (cov_xx);
00128         ar& BOOST_SERIALIZATION_NVP (cov_yy);
00129         ar& BOOST_SERIALIZATION_NVP (cov_zz);
00130         ar& BOOST_SERIALIZATION_NVP (cov_xy);
00131         ar& BOOST_SERIALIZATION_NVP (cov_yz);
00132         ar& BOOST_SERIALIZATION_NVP (cov_zx);
00133
00134         ar& BOOST_SERIALIZATION_NVP (latitude);
00135         ar& BOOST_SERIALIZATION_NVP (longitude);
00136         ar& BOOST_SERIALIZATION_NVP (height);
00137
00138         ar& BOOST_SERIALIZATION_NVP (valid_sats);
00139         ar& BOOST_SERIALIZATION_NVP (solution_status);
00140         ar& BOOST_SERIALIZATION_NVP (solution_type);
00141         ar& BOOST_SERIALIZATION_NVP (AR_ratio_factor);
00142         ar& BOOST_SERIALIZATION_NVP (AR_ratio_threshold);
00143
00144         ar& BOOST_SERIALIZATION_NVP (gdop);
00145         ar& BOOST_SERIALIZATION_NVP (pdop);
00146         ar& BOOST_SERIALIZATION_NVP (hdop);
00147         ar& BOOST_SERIALIZATION_NVP (vdop);
00148
00149         ar& BOOST_SERIALIZATION_NVP (user_clk_drift_ppm);
00150         ar& BOOST_SERIALIZATION_NVP (utc_time);
00151
00152         ar& BOOST_SERIALIZATION_NVP (vel_e);
00153         ar& BOOST_SERIALIZATION_NVP (vel_n);
00154         ar& BOOST_SERIALIZATION_NVP (vel_u);
00155
00156         ar& BOOST_SERIALIZATION_NVP (cog);
00157         ar& BOOST_SERIALIZATION_NVP (geohash);
00158     }
00159 };
00160
00161
00162 /** \} */
00163 /** \} */
00164 #endif // GNSS_SDR_MONITOR_PVT_H

```

## 11.262 monitor\_pvt\_udp\_sink.h File Reference

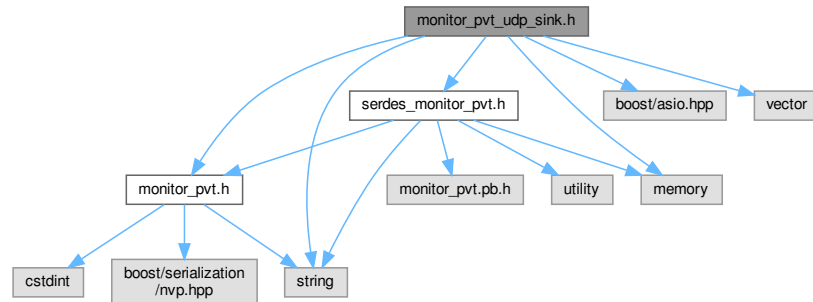
Interface of a class that sends serialized [Monitor\\_Pvt](#) objects over udp to one or multiple endpoints.

```

#include "monitor_pvt.h"
#include "serdes_monitor_pvt.h"
#include <boost/asio.hpp>
#include <memory>
#include <string>
#include <vector>

```

Include dependency graph for `monitor_pvt_udp_sink.h`:



## Classes

- class [Monitor\\_Pvt\\_Udp\\_Sink](#)

## Typedefs

- using [b\\_io\\_context](#) = boost::asio::io\_service

## 11.262.1 Detailed Description

Interface of a class that sends serialized [Monitor\\_Pvt](#) objects over udp to one or multiple endpoints.

### Author

Álvaro Cebrián Juan, 2019. [acebrianjuan\(at\)gmail.com](mailto:acebrianjuan(at)gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [monitor\\_pvt\\_udp\\_sink.h](#).

---

## 11.263 monitor\_pvt\_udp\_sink.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file monitor_pvt_udp_sink.h
00003  * \brief Interface of a class that sends serialized Monitor_Pvt objects
00004  * over udp to one or multiple endpoints
00005  * \author Álvaro Cebrián Juan, 2019. acebrianjuan(at)gmail.com
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_MONITOR_PVT_UDP_SINK_H
00019 #define GNSS_SDR_MONITOR_PVT_UDP_SINK_H
00020
00021 #include "monitor_pvt.h"
00022 #include "serdes_monitor_pvt.h"
00023 #include <boost/asio.hpp>
00024 #include <memory>
00025 #include <string>
00026 #include <vector>
00027
00028 /** \addtogroup PVT
00029  * \{
  
```



```

00030 /** \addtogroup PVT_libs
00031  * \{ */
00032
00033
00034 #if USE_BOOST_ASIO_IO_CONTEXT
00035 using b_io_context = boost::asio::io_context;
00036 #else
00037 using b_io_context = boost::asio::io_service;
00038 #endif
00039
00040 class Monitor_Pvt_Udp_Sink
00041 {
00042 public:
00043     Monitor_Pvt_Udp_Sink(
00044         const std::vector<std::string>& addresses,
00045         const std::vector<std::string>& ports,
00046         bool protobuf_enabled);
00047     bool write_monitor_pvt(const Monitor_Pvt* const monitor_pvt);
00048 private:
00049     Serdes_Monitor_Pvt serdes;
00050     b_io_context io_context;
00051     boost::asio::ip::udp::socket socket;
00052     std::vector<boost::asio::ip::udp::endpoint> endpoints;
00053     boost::system::error_code error;
00054     bool use_protobuf;
00055 };
00056
00057
00058
00059 /** \} */
00060 /** \} */
00061 #endif // GNSS_SDR_MONITOR_PVT_UDP_SINK_H

```

## 11.264 nmea\_printer.h File Reference

Interface of a NMEA 2.1 printer for GNSS-SDR This class provides a implementation of a subset of the NMEA-0183 standard for interfacing marine electronic devices as defined by the National Marine Electronics Association (NMEA). See <https://www.nmea.org/> for the NMEA 183 standard.

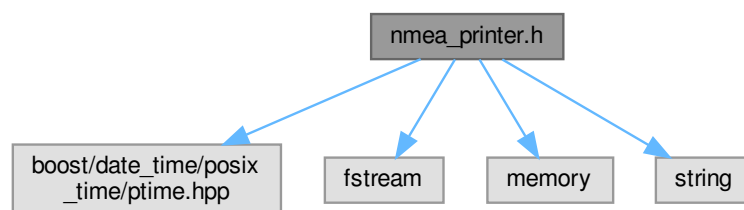
```
#include <boost/date_time/posix_time/ptime.hpp>
```

```
#include <fstream>
```

```
#include <memory>
```

```
#include <string>
```

Include dependency graph for nmea\_printer.h:



### Classes

- class [Nmea\\_Printer](#)

*This class provides a implementation of a subset of the NMEA-0183 standard for interfacing marine electronic devices as defined by the National Marine Electronics Association (NMEA).*

### 11.264.1 Detailed Description

Interface of a NMEA 2.1 printer for GNSS-SDR This class provides a implementation of a subset of the NMEA-0183 standard for interfacing marine electronic devices as defined by the National Marine Electronics Association

(NMEA). See <https://www.nmea.org/> for the NMEA 183 standard.

#### Author

Javier Arribas, 2012. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [nmea\\_printer.h](#).

---

## 11.265 nmea\_printer.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file nmea_printer.h
00003  * \brief Interface of a NMEA 2.1 printer for GNSS-SDR
00004  * This class provides a implementation of a subset of the NMEA-0183 standard for interfacing
00005  * marine electronic devices as defined by the National Marine Electronics Association (NMEA).
00006  * See https://www.nmea.org/ for the NMEA 183 standard
00007  *
00008  * \author Javier Arribas, 2012. jarribas(at)cttc.es
00009  *
00010  *
00011  * -----
00012  *
00013  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00014  * This file is part of GNSS-SDR.
00015  *
00016  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00017  * SPDX-License-Identifier: GPL-3.0-or-later
00018  *
00019  * -----
00020  */
00021
00022 #ifndef GNSS_SDR_NMEA_PRINTER_H
00023 #define GNSS_SDR_NMEA_PRINTER_H
00024
00025 #include <boost/date_time/posix_time/ptime.hpp> // for ptime
00026 #include <fstream> // for ofstream
00027 #include <memory> // for shared_ptr
00028 #include <string> // for string
00029
00030 /** \addtogroup PVT
00031  * \{ */
00032 /** \addtogroup PVT_libs
00033  * \{ */
00034
00035
00036 class Rtklib_Solver;
00037
00038 /*!
00039 * \brief This class provides a implementation of a subset of the NMEA-0183 standard for interfacing
00040 * marine electronic devices as defined by the National Marine Electronics Association (NMEA).
00041 *
00042 * See https://en.wikipedia.org/wiki/NMEA_0183
00043 */
00044 class Nmea_Printer
00045 {
00046 public:
00047  /*!
00048  * \brief Default constructor.
00049  */
00050  Nmea_Printer(const std::string& filename, bool flag_nmea_output_file, bool flag_nmea_tty_port,
00051               std::string nmea_dump_devname, const std::string& base_path = ".");
00052
00053  /*!
00054  * \brief Default destructor.
00055  */
00056  ~Nmea_Printer();
00057
00058  /*!
00059  * \brief Print NMEA PVT and satellite info to the initialized device
00060  */
00061  bool Print_Nmea_Line(const Rtklib_Solver* const pvt_data);
00062 private:
00063  int init_serial(const std::string& serial_device); // serial port control
00064  void close_serial() const;
00065  std::string get_GPGGA() const; // fix data
00066  std::string get_GPGSV() const; // satellite data
00067  std::string get_GPGSA() const; // overall satellite reception data
00068  std::string get_GPRMC() const; // minimum recommended data

```

```

00069     std::string get.UTC_NMEA_time(const boost::posix_time::ptime d_position.UTC_time) const;
00070     std::string longitude_to_hm(double longitude) const;
00071     std::string latitude_to_hm(double lat) const;
00072     char checksum(const std::string& sentence) const;
00073
00074     const Rtklib_Solver* d_PVT_data;
00075
00076     std::ofstream nmea_file_descriptor; // Output file stream for NMEA log file
00077
00078     std::string nmea_filename; // String with the NMEA log filename
00079     std::string nmea_base_path;
00080     std::string nmea_devname;
00081
00082     int nmea_dev_descriptor; // NMEA serial device descriptor (i.e. COM port)
00083     bool d_flag_nmea_output_file;
00084 };
00085
00086
00087 /** \} */
00088 /** \} */
00089 #endif // GNSS_SDR_NMEA_PRINTER_H

```

## 11.266 pvt\_conf.h File Reference

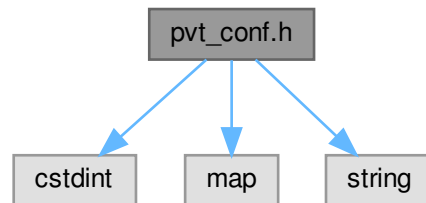
Class that contains all the configuration parameters for the PVT block.

```
#include <stdint>
```

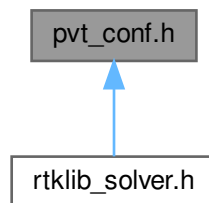
```
#include <map>
```

```
#include <string>
```

Include dependency graph for pvt\_conf.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [Pvt\\_Conf](#)

### 11.266.1 Detailed Description

Class that contains all the configuration parameters for the PVT block.

#### Author

Carles Fernandez, 2018. cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [pvt\\_conf.h](#).

---

## 11.267 pvt\_conf.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file pvt_conf.h
00003  * \brief Class that contains all the configuration parameters for the PVT block
00004  * \author Carles Fernandez, 2018. cfernandez(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_PVT_CONF_H
00018 #define GNSS_SDR_PVT_CONF_H
00019
00020 #include <cstdint>
00021 #include <map>
00022 #include <string>
00023
00024 /** \addtogroup PVT
00025  * \{ */
00026 /** \addtogroup PVT_libs
00027  * \{ */
00028
00029
00030 class Pvt_Conf
00031 {
00032 public:
00033     std::map<int, int> rtcm_msg_rate_ms;
00034
00035     std::string rinex_name = std::string("-");
00036     std::string dump_filename;
00037     std::string nmea_dump_filename;
00038     std::string nmea_dump_devname;
00039     std::string rtcm_dump_devname;
00040     std::string an_dump_devname;
00041     std::string output_path = std::string(".");
00042     std::string rinex_output_path = std::string(".");
00043     std::string gpx_output_path = std::string(".");
00044     std::string geojson_output_path = std::string(".");
00045     std::string nmea_output_file_path = std::string(".");
00046     std::string kml_output_path = std::string(".");
00047     std::string xml_output_path = std::string(".");
00048     std::string rtcm_output_file_path = std::string(".");
00049     std::string has_output_file_path = std::string(".");
00050     std::string udp_addresses;
00051     std::string udp_ports;
00052     std::string udp_eph_addresses;
00053     std::string log_source_timetag_file;
00054
00055     uint32_t type_of_receiver = 0;
00056     uint32_t observable_interval_ms = 20;
00057
00058     int32_t output_rate_ms = 0;
00059     int32_t display_rate_ms = 0;
00060     int32_t kml_rate_ms = 20;
00061     int32_t gpx_rate_ms = 20;
00062     int32_t geojson_rate_ms = 20;
00063     int32_t nmea_rate_ms = 20;
00064     int32_t rinex_version = 0;
00065     int32_t rinexobs_rate_ms = 0;
00066     int32_t an_rate_ms = 20;
00067     int32_t max_obs_block_rx_clock_offset_ms = 40;

```

```

00068     int udp_eph_port = 0;
00069     int rtk_trace_level = 0;
00070
00071     uint16_t rtcm_tcp_port = 0;
00072     uint16_t rtcm_station_id = 0;
00073
00074     bool flag_nmea_tty_port = false;
00075     bool flag_rtcn_server = false;
00076     bool flag_rtcn_tty_port = false;
00077     bool output_enabled = true;
00078     bool rinex_output_enabled = true;
00079     bool gpx_output_enabled = true;
00080     bool geojson_output_enabled = true;
00081     bool nmea_output_file_enabled = true;
00082     bool an_output_enabled = false;
00083     bool kml_output_enabled = true;
00084     bool xml_output_enabled = true;
00085     bool rtcm_output_file_enabled = true;
00086     bool monitor_enabled = false;
00087     bool monitor_ephemeris_enabled = false;
00088     bool protobuf_enabled = true;
00089     bool enable_rx_clock_correction = true;
00090     bool show_local_time_zone = false;
00091     bool pre_2009_file = false;
00092     bool dump = false;
00093     bool dump_mat = true;
00094     bool log_source_timetag = false;
00095     bool use_e6_for_pvt = true;
00096     bool use_has_corrections = true;
00097     bool use_unhealthy_sats = false;
00098     bool osnma_strict = false;
00099
00100     // PVT KF parameters
00101     bool enable_pvt_kf = false;
00102     double measures_ecef_pos_sd_m = 1.0;
00103     double measures_ecef_vel_sd_ms = 0.1;
00104     double system_ecef_pos_sd_m = 0.01;
00105     double system_ecef_vel_sd_ms = 0.001;
00106 };
00107
00108
00109 /** \} */
00110 /** \} */
00111 #endif // GNSS_SDR_PVT_CONF_H

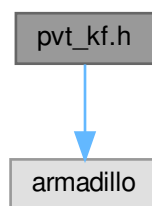
```

## 11.268 pvt\_kf.h File Reference

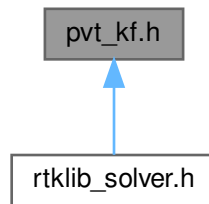
Kalman Filter for Position and Velocity.

```
#include <armadillo>
```

Include dependency graph for pvt\_kf.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Pvt\\_Kf](#)  
*Kalman Filter for Position and Velocity.*

### 11.268.1 Detailed Description

Kalman Filter for Position and Velocity.

#### Author

Javier Arribas, 2023. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2023 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [pvt\\_kf.h](#).

---

## 11.269 pvt\_kf.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file pvt_kf.h
00003  * \brief Kalman Filter for Position and Velocity
00004  * \author Javier Arribas, 2023. jarribas(at)cttc.es
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2023 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_PVT_KF_H
00019 #define GNSS_SDR_PVT_KF_H
00020
00021 #include <armadillo>
00022
00023 /** \addtogroup PVT
00024  * \{ */
00025 /** \addtogroup PVT_libs
00026  * \{ */
00027
00028
00029 /*!
00030  * \brief Kalman Filter for Position and Velocity
00031  *
00032  */
00033 class Pvt_Kf
  
```

```

00034 {
00035 public:
00036     Pvt_Kf() = default;
00037     virtual ~Pvt_Kf() = default;
00038     void init_Kf(const arma::vec& p,
00039                 const arma::vec& v,
00040                 double update_interval_s,
00041                 double measures_ecef_pos_sd_m,
00042                 double measures_ecef_vel_sd_ms,
00043                 double system_ecef_pos_sd_m,
00044                 double system_ecef_vel_sd_ms);
00045     bool is_initialized() const;
00046     void run_Kf(const arma::vec& p, const arma::vec& v);
00047     void get_pv_Kf(arma::vec& p, arma::vec& v) const;
00048     void reset_Kf();
00049 private:
00050     // Kalman Filter class variables
00051     arma::mat d_F;
00052     arma::mat d_H;
00053     arma::mat d_R;
00054     arma::mat d_Q;
00055     arma::mat d_P_old_old;
00056     arma::mat d_P_new_old;
00057     arma::mat d_P_new_new;
00058     arma::vec d_x_old_old;
00059     arma::vec d_x_new_old;
00060     arma::vec d_x_new_new;
00061     bool d_initialized{false};
00062 };
00063
00064
00065
00066 /** \} */
00067 /** \} */
00068 #endif // GNSS_SDR_Pvt_Kf_H

```

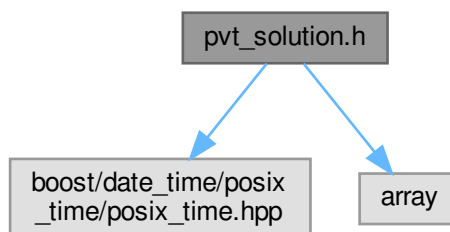
## 11.270 pvt\_solution.h File Reference

Interface of a base class for a PVT solution.

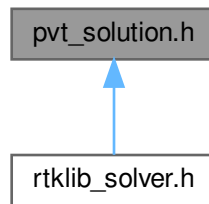
```
#include <boost/date_time/posix_time/posix_time.hpp>
```

```
#include <array>
```

Include dependency graph for pvt\_solution.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Pvt\\_Solution](#)  
*Base class for a PVT solution.*

### 11.270.1 Detailed Description

Interface of a base class for a PVT solution.

#### Author

Carles Fernandez-Prades, 2015. [cfernandez\(at\)cttc.es](mailto:cfernandez@cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [pvt\\_solution.h](#).

---

## 11.271 pvt\_solution.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file pvt_solution.h
00003  * \brief Interface of a base class for a PVT solution
00004  * \author Carles Fernandez-Prades, 2015. cfernandez(at)cttc.es
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_PVT_SOLUTION_H
00019 #define GNSS_SDR_PVT_SOLUTION_H
00020
00021 #include <boost/date_time/posix_time/posix_time.hpp>
00022 #include <array>
00023
00024 /** \addtogroup PVT
00025  * \{ */
00026 /** \addtogroup PVT_libs
00027  * \{ */
00028
00029
00030 /*!
00031 * \brief Base class for a PVT solution
00032 *
00033 */
  
```



```

00034 class Pvt_Solution
00035 {
00036 public:
00037     Pvt_Solution() = default;
00038     virtual ~Pvt_Solution() = default;
00039
00040     virtual double get_hdop() const = 0;
00041     virtual double get_vdop() const = 0;
00042     virtual double get_pdop() const = 0;
00043     virtual double get_gdop() const = 0;
00044
00045     std::array<double, 3> get_rx_pos() const;
00046     std::array<double, 3> get_rx_vel() const;
00047     boost::posix_time::ptime get_position_UTC_time() const;
00048     double get_latitude() const;          ///< Get RX position Latitude WGS84 [deg]
00049     double get_longitude() const;         ///< Get RX position Longitude WGS84 [deg]
00050     double get_height() const;            ///< Get RX position height WGS84 [m]
00051     double get_time_offset_s() const;     ///< Get RX time offset [s]
00052     double get_clock_drift_ppm() const;   ///< Get the Rx clock drift [ppm]
00053     double get_speed_over_ground() const; ///< Get RX speed over ground [m/s]
00054     double get_course_over_ground() const; ///< Get RX course over ground [deg]
00055     int get_num_valid_observations() const; ///< Get the number of valid pseudorange observations
                                (valid satellites)
00056     bool is_pre_2009() const;
00057     bool is_valid_position() const;
00058
00059     void set_rx_pos(const std::array<double, 3> &pos); ///< Set position: X, Y, Z in Cartesian ECEF
                                coordinates [m]
00060     void set_rx_vel(const std::array<double, 3> &vel); ///< Set velocity: East [m/s], North [m/s], Up
                                [m/s]
00061     void set_position_UTC_time(const boost::posix_time::ptime &pt);
00062     void set_time_offset_s(double offset);          ///< Set RX time offset [s]
00063     void set_clock_drift_ppm(double clock_drift_ppm); ///< Set the Rx clock drift [ppm]
00064     void set_speed_over_ground(double speed_m_s);   ///< Set RX speed over ground [m/s]
00065     void set_course_over_ground(double cog_deg);    ///< Set RX course over ground [deg]
00066     void set_valid_position(bool is_valid);
00067     void set_num_valid_observations(int num);        ///< Set the number of valid pseudorange observations
                                (valid satellites)
00068     void set_pre_2009_file(bool pre_2009_file);    ///< Flag for the week rollover computation in post
                                processing mode for signals older than 2009
00069
00070 private:
00071     /*
00072      * Conversion of Cartesian coordinates (X,Y,Z) to geographical
00073      * coordinates (d_latitude_d, d_longitude_d, d_height_m) on a selected reference ellipsoid.
00074      *
00075      * \param[in] X [m] Cartesian coordinate
00076      * \param[in] Y [m] Cartesian coordinate
00077      * \param[in] Z [m] Cartesian coordinate
00078      * \param[in] ellipsoid_selection. Choices of Reference Ellipsoid for Geographical Coordinates:
00079      * 0 - International Ellipsoid 1924.
00080      * 1 - International Ellipsoid 1967.
00081      * 2 - World Geodetic System 1972.
00082      * 3 - Geodetic Reference System 1980.
00083      * 4 - World Geodetic System 1984.
00084      *
00085      */
00086     int cart2geo(double X, double Y, double Z, int ellipsoid_selection);
00087
00088     std::array<double, 3> d_rx_pos{};
00089     std::array<double, 3> d_rx_vel{};
00090     boost::posix_time::ptime d_position_UTC_time;
00091
00092     double d_latitude_d{0.0};           // RX position Latitude WGS84 [deg]
00093     double d_longitude_d{0.0};          // RX position Longitude WGS84 [deg]
00094     double d_height_m{0.0};             // RX position height WGS84 [m]
00095     double d_rx_dt_s{0.0};             // RX time offset [s]
00096     double d_rx_clock_drift_ppm{0.0};   // RX clock drift [ppm]
00097     double d_speed_over_ground_m_s{0.0}; // RX speed over ground [m/s]
00098     double d_course_over_ground_d{0.0}; // RX course over ground [deg]
00099
00100     int d_valid_observations{0}; // Number of valid observations in this epoch
00101
00102     bool d_pre_2009_file{false}; // Flag to correct week rollover in post processing mode for signals
                                older than 2009
00103     bool d_valid_position{false};
00104 };
00105
00106
00107 /** \} */
00108 /** \} */
00109 #endif // GNSS_SDR_PVT_SOLUTION_H

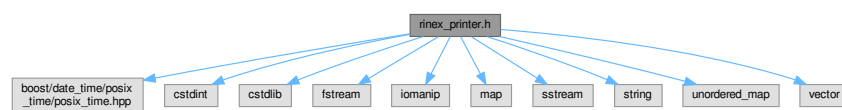
```

## 11.272 rinex\_printer.h File Reference

Interface of a RINEX 2.11 / 3.01 printer See <ftp://igs.org/pub/data/format/rinex301.pdf>.

```
#include <boost/date_time/posix_time/posix_time.hpp>
#include <cstdint>
#include <cstdlib>
#include <fstream>
#include <iomanip>
#include <map>
#include <sstream>
#include <string>
#include <unordered_map>
#include <vector>
```

Include dependency graph for rinex\_printer.h:



### Classes

- class [Rinex\\_Printer](#)

*Class that handles the generation of Receiver INdependent EXchange format (RINEX) files.*

### Functions

- std::string [asString](#) (long double x, std::string::size\_type precision)
- int64\_t [asInt](#) (const std::string &s)

### 11.272.1 Detailed Description

Interface of a RINEX 2.11 / 3.01 printer See <ftp://igs.org/pub/data/format/rinex301.pdf>.

Receiver Independent EXchange Format (RINEX): The first proposal for the Receiver Independent Exchange Format RINEX was developed by the Astronomical Institute of the University of Berne for the easy exchange of the GPS data to be collected during the large European GPS campaign EUREF 89, which involved more than 60 GPS receivers of 4 different manufacturers. The governing aspect during the development was the fact that most geodetic processing software for GPS data use a well-defined set of observables: 1) The carrier-phase measurement at one or both carriers (actually being a measurement on the beat frequency between the received carrier of the satellite signal and a receiver-generated reference frequency). 2) The pseudorange (code) measurement, equivalent to the difference of the time of reception (expressed in the time frame of the receiver) and the time of transmission (expressed in the time frame of the satellite) of a distinct satellite signal. 3) The observation time being the reading of the receiver clock at the instant of validity of the carrier-phase and/or the code measurements. Note: A collection of the formats currently used by the IGS can be found here: <https://kb.igs.org/hc/en-us/articles/201096516-IGS-Formats>

#### Author

Carles Fernandez Prades, 2011. [cfernandez\(at\)cttc.es](mailto:cfernandez@cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [rinex\\_printer.h](#).

---

## 11.273 rinex\_printer.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file rinex_printer.h
00003  * \brief Interface of a RINEX 2.11 / 3.01 printer
00004  * See ftp://igs.org/pub/data/format/rinex301.pdf
00005  *
00006  * Receiver Independent EXchange Format (RINEX):
00007  * The first proposal for the Receiver Independent Exchange Format RINEX
00008  * was developed by the Astronomical Institute of the University of Berne
00009  * for the easy exchange of the GPS data to be collected during the large
00010  * European GPS campaign EUREF 89, which involved more than 60 GPS receivers
00011  * of 4 different manufacturers.
00012  * The governing aspect during the development was the fact that most geodetic
00013  * processing software for GPS data use a well-defined set of observables:
00014  * 1) The carrier-phase measurement at one or both carriers (actually being a
00015  * measurement on the beat frequency between the received carrier of the
00016  * satellite signal and a receiver-generated reference frequency).
00017  * 2) The pseudorange (code) measurement , equivalent to the difference
00018  * of the time of reception (expressed in the time frame of the receiver)
00019  * and the time of transmission (expressed in the time frame of the satellite)
00020  * of a distinct satellite signal.
00021  * 3) The observation time being the reading of the receiver clock at the
00022  * instant of validity of the carrier-phase and/or the code measurements.
00023  * Note: A collection of the formats currently used by the IGS can be found
00024  * here: https://kb.igs.org/hc/en-us/articles/201096516-IGS-Formats
00025  * \author Carles Fernandez Prades, 2011. cfernandez(at)cttc.es
00026  *
00027  * -----
00028  *
00029  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00030  * This file is part of GNSS-SDR.
00031  *
00032  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00033  * SPDX-License-Identifier: GPL-3.0-or-later
00034  *
00035  * -----
00036  */
00037
00038 #ifndef GNSS_SDR_RINEX_PRINTER_H
00039 #define GNSS_SDR_RINEX_PRINTER_H
00040
00041 #include <boost/date_time/posix_time/posix_time.hpp>
00042 #include <stdint> // for int32_t
00043 #include <cstdlib> // for strtol, strtod
00044 #include <fstream> // for fstream
00045 #include <iomanip> // for setprecision
00046 #include <map> // for map
00047 #include <sstream> // for stringstream
00048 #include <string> // for string
00049 #include <unordered_map> // for unordered_map
00050 #include <vector> // for vector
00051
00052
00053 /** \addtogroup PVT
00054  * \{ */
00055 /** \addtogroup PVT_libs
00056  * \{ */
00057
00058
00059 class Beidou_Dnav_Ephemeris;
00060 class Beidou_Dnav_Iono;
00061 class Beidou_Dnav_Utc_Model;
00062 class Galileo_Ephemeris;
00063 class Galileo_Iono;
00064 class Galileo_Utc_Model;
00065 class Glonass_Gnav_Almanac;
00066 class Glonass_Gnav_Ephemeris;
00067 class Glonass_Gnav_Utc_Model;
00068 class Gnss_Synchro;
00069 class Gps_CNAV_Ephemeris;
00070 class Gps_CNAV_Iono;
00071 class Gps_CNAV_Utc_Model;
00072 class Gps_Ephemeris;
00073 class Gps_Iono;
00074 class Gps_Navigation_Message;
00075 class Gps_Utc_Model;
00076 class Rtklib_Solver;
00077
00078
00079 /*!
00080 * \brief Class that handles the generation of Receiver
00081 * INdependent EXchange format (RINEX) files
00082 */
00083 class Rinex_Printer
00084 {
00085 public:
00086     /*!
00087     * \brief Constructor. Creates GNSS Navigation and Observables RINEX files.

```

```

00088     */
00089     explicit Rinex_Printer(int version = 0,
00090         const std::string& base_path = ".",
00091         const std::string& base_name = "-");
00092
00093     /*!
00094     * \brief Destructor. Removes created files if empty.
00095     */
00096     ~Rinex_Printer();
00097
00098     /*!
00099     * \brief Print RINEX annotation. If it is the first annotation, it also
00100     * prints the RINEX headers for navigation and observation files. If it is
00101     * not the first annotation, it only annotates the observation, and updates
00102     * the navigation header if UTC data was not available when writing it for
00103     * the first time. The meaning of type_of_rx is as follows:
00104     *
00105     * type_of_rx      | Signals
00106     * ----- | -----
00107     * 0      | Unknown
00108     * 1      | GPS L1 C/A
00109     * 2      | GPS L2C
00110     * 3      | GPS L5
00111     * 4      | Galileo E1B
00112     * 5      | Galileo E5a
00113     * 6      | Galileo E5b
00114     * 7      | GPS L1 C/A + GPS L2C
00115     * 8      | GPS L1 C/A + GPS L5
00116     * 9      | GPS L1 C/A + Galileo E1B
00117     * 10     | GPS L1 C/A + Galileo E5a
00118     * 11     | GPS L1 C/A + Galileo E5b
00119     * 12     | Galileo E1B + GPS L2C
00120     * 13     | Galileo E5a + GPS L5
00121     * 14     | Galileo E1B + Galileo E5a
00122     * 15     | Galileo E1B + Galileo E5b
00123     * 16     | GPS L2C + GPS L5
00124     * 17     | GPS L2C + Galileo E5a
00125     * 20     | GPS L5 + Galileo E5b
00126     * 21     | GPS L1 C/A + Galileo E1B + GPS L2C
00127     * 22     | GPS L1 C/A + Galileo E1B + GPS L5
00128     * 23     | GLONASS L1 C/A
00129     * 24     | GLONASS L2 C/A
00130     * 25     | GLONASS L1 C/A + GLONASS L2 C/A
00131     * 26     | GPS L1 C/A + GLONASS L1 C/A
00132     * 27     | Galileo E1B + GLONASS L1 C/A
00133     * 28     | GPS L2C + GLONASS L1 C/A
00134     * 29     | GPS L1 C/A + GLONASS L2 C/A
00135     * 30     | Galileo E1B + GLONASS L2 C/A
00136     * 31     | GPS L2C + GLONASS L2 C/A
00137     * 32     | GPS L1 C/A + Galileo E1B + GPS L5 + Galileo E5a
00138     * 33     | GPS L1 C/A + Galileo E1B + Galileo E5a
00139     * 100    | Galileo E6B
00140     * 101    | Galileo E1B + Galileo E6B
00141     * 102    | Galileo E5a + Galileo E6B
00142     * 103    | Galileo E5b + Galileo E6B
00143     * 104    | Galileo E1B + Galileo E5a + Galileo E6B
00144     * 105    | Galileo E1B + Galileo E5b + Galileo E6B
00145     * 106    | GPS L1 C/A + Galileo E1B + Galileo E6B
00146     * 107    | GPS L1 C/A + Galileo E6B
00147     * 108    | GPS L1 C/A + Galileo E1B + GPS L5 + Galileo E5a + Galileo E6B
00148     * 500    | BeiDou B1I
00149     * 501    | BeiDou B1I + GPS L1 C/A
00150     * 502    | BeiDou B1I + Galileo E1B
00151     * 503    | BeiDou B1I + GLONASS L1 C/A
00152     * 504    | BeiDou B1I + GPS L1 C/A + Galileo E1B
00153     * 505    | BeiDou B1I + GPS L1 C/A + GLONASS L1 C/A + Galileo E1B
00154     * 506    | BeiDou B1I + BeiDou B3I
00155     * 600    | BeiDou B3I
00156     * 601    | BeiDou B3I + GPS L2C
00157     * 602    | BeiDou B3I + GLONASS L2 C/A
00158     * 603    | BeiDou B3I + GPS L2C + GLONASS L2 C/A
00159     * 604    | BeiDou B3I + GPS L1 C/A
00160     * 605    | BeiDou B3I + Galileo E1B
00161     * 606    | BeiDou B3I + GLONASS L1 C/A
00162     * 607    | BeiDou B3I + GPS L1 C/A + Galileo E1B
00163     * 608    | BeiDou B3I + GPS L1 C/A + Galileo E1B + BeiDou B1I
00164     * 609    | BeiDou B3I + GPS L1 C/A + Galileo E1B + GLONASS L1 C/A
00165     * 610    | BeiDou B3I + GPS L1 C/A + Galileo E1B + GLONASS L1 C/A + BeiDou B1I
00166     * 1000   | GPS L1 C/A + GPS L2C + GPS L5
00167     * 1001   | GPS L1 C/A + Galileo E1B + GPS L2C + GPS L5 + Galileo E5a
00168
00169     */
00170     void print_rinex_annotation(const Rtklib_Solver* pvt_solver,
00171         const std::map<int, Gnss_Synchro>& gnss_observables_map,
00172         double rx_time,
00173         int type_of_rx,
00174         bool flag_write_RINEX_obs_output);

```

```

00175
00176     /*!
00177      * \brief Print RINEX annotation for GPS NAV message
00178      */
00179     void log_rinex_nav_gps_nav(int type_of_rx,
00180                               const std::map<int32_t, Gps_Ephemeris>& new_eph);
00181
00182     /*!
00183      * \brief Print RINEX annotation for GPS CNAV message
00184      */
00185     void log_rinex_nav_gps_cnav(int type_of_rx,
00186                                 const std::map<int32_t, Gps_CNAV_Ephemeris>& new_cnav_eph);
00187
00188     /*!
00189      * \brief Print RINEX annotation for Galileo NAV message
00190      */
00191     void log_rinex_nav_gal_nav(int type_of_rx,
00192                                const std::map<int32_t, Galileo_Ephemeris>& new_gal_eph);
00193
00194     /*!
00195      * \brief Print RINEX annotation for Glonass GNAV message
00196      */
00197     void log_rinex_nav_glo_gnav(int type_of_rx,
00198                                 const std::map<int32_t, Glonass_Gnav_Ephemeris>& new_glo_eph);
00199
00200     /*!
00201      * \brief Print RINEX annotation for BeiDou DNAV message
00202      */
00203     void log_rinex_nav_bds_dnav(int type_of_rx,
00204                                 const std::map<int32_t, Beidou_Dnav_Ephemeris>& new_bds_eph);
00205
00206     /*!
00207      * \brief Set processing for signals older than 2009
00208      */
00209     void set_pre_2009_file(bool pre_2009_file);
00210
00211     /*!
00212      * \brief Returns true is the RINEX file headers are already written
00213      */
00214     inline bool is_rinex_header_written() const
00215     {
00216         return d_rinex_header_written;
00217     }
00218
00219     /*!
00220      * \brief Returns name of RINEX navigation file(s)
00221      */
00222     inline std::vector<std::string> get_navfilename() const
00223     {
00224         return output_navfilename;
00225     }
00226
00227     /*!
00228      * \brief Returns name of RINEX observation file
00229      */
00230     inline std::string get_obsfilename() const
00231     {
00232         return obsfilename;
00233     }
00234
00235 private:
00236     const std::unordered_map<std::string, std::string> satelliteSystem = {
00237         {"GPS", "G"},
00238         {"GLONASS", "R"},
00239         {"SBAS payload", "S"},
00240         {"Galileo", "E"},
00241         {"Beidou", "C"},
00242         {"Mixed", "M"}}; // RINEX v3.02 codes
00243
00244     /*
00245      * Generates the GPS Observation data header
00246      */
00247     void rinex_obs_header(std::fstream& out,
00248                           const Gps_Ephemeris& eph,
00249                           double d_TOW_first_observation);
00250
00251     /*
00252      * Generates the GPS L2 Observation data header
00253      */
00254     void rinex_obs_header(std::fstream& out,
00255                           const Gps_CNAV_Ephemeris& eph,
00256                           double d_TOW_first_observation,
00257                           const std::string& gps_bands = "2S");
00258
00259     /*
00260      * Generates the dual frequency GPS L1 & L2/L5 Observation data header

```

```

00262     */
00263 void rinex_obs_header(std::fstream& out,
00264     const Gps_Ephemeris& eph,
00265     const Gps_CNAV_Ephemeris& eph_cnav,
00266     double d_TOW_first_observation,
00267     const std::string& gps_bands = "1C 2S");
00268
00269     /*
00270     * Generates the Galileo Observation data header.
00271     * Example: bands("1B"), bands("1B 5X"), bands("5X"), ... Default: "1B".
00272     */
00273 void rinex_obs_header(std::fstream& out,
00274     const Galileo_Ephemeris& eph,
00275     double d_TOW_first_observation,
00276     const std::string& bands = "1B");
00277
00278     /*
00279     * Generates the Mixed (GPS/Galileo) Observation data header.
00280     * Example: galileo_bands("1B"), galileo_bands("1B 5X"),
00281     * galileo_bands("5X"), ... Default: "1B".
00282     */
00283 void rinex_obs_header(std::fstream& out,
00284     const Gps_Ephemeris& gps_eph,
00285     const Galileo_Ephemeris& galileo_eph,
00286     double d_TOW_first_observation,
00287     const std::string& galileo_bands = "1B");
00288
00289     /*
00290     * Generates the Mixed (GPS/Galileo) Observation data header.
00291     * Example: galileo_bands("1B"), galileo_bands("1B 5X"), galileo_bands("5X"), ... Default: "1B".
00292     */
00293 void rinex_obs_header(std::fstream& out,
00294     const Gps_Ephemeris& gps_eph,
00295     const Gps_CNAV_Ephemeris& eph_cnav,
00296     const Galileo_Ephemeris& galileo_eph,
00297     double d_TOW_first_observation,
00298     const std::string& gps_bands = "1C 2S",
00299     const std::string& galileo_bands = "1B");
00300
00301     /*
00302     * Generates the Mixed (GPS/Galileo) Observation data header.
00303     * Example: galileo_bands("1B"), galileo_bands("1B 5X"), galileo_bands("5X"), ... Default: "1B".
00304     */
00305 void rinex_obs_header(std::fstream& out,
00306     const Gps_CNAV_Ephemeris& eph_cnav,
00307     const Galileo_Ephemeris& galileo_eph,
00308     double d_TOW_first_observation,
00309     const std::string& gps_bands = "2S",
00310     const std::string& galileo_bands = "1B");
00311
00312     /*
00313     * Generates the GLONASS GNAV Observation data header.
00314     * Example: bands("1C"), bands("1C 2C"), bands("2C"), ... Default: "1C".
00315     */
00316 void rinex_obs_header(std::fstream& out,
00317     const Glonass_Gnav_Ephemeris& eph,
00318     double d_TOW_first_observation,
00319     const std::string& bands = "1G");
00320
00321     /*
00322     * Generates the Mixed (GPS L1 C/A /GLONASS) Observation data header.
00323     * Example: galileo_bands("1C"), galileo_bands("1B 5X"), galileo_bands("5X"), ... Default: "1B".
00324     */
00325 void rinex_obs_header(std::fstream& out,
00326     const Gps_Ephemeris& gps_eph,
00327     const Glonass_Gnav_Ephemeris& glonass_gnav_eph,
00328     double d_TOW_first_observation,
00329     const std::string& glonass_bands = "1C");
00330
00331     /*
00332     * Generates the Mixed (Galileo/GLONASS) Observation data header.
00333     * Example: galileo_bands("1C"), galileo_bands("1B 5X"), galileo_bands("5X"), ... Default: "1B".
00334     */
00335 void rinex_obs_header(std::fstream& out,
00336     const Galileo_Ephemeris& galileo_eph,
00337     const Glonass_Gnav_Ephemeris& glonass_gnav_eph,
00338     double d_TOW_first_observation,
00339     const std::string& galileo_bands = "1B",
00340     const std::string& glonass_bands = "1C");
00341
00342     /*
00343     * Generates the Mixed (GPS L2C/GLONASS) Observation data header.
00344     * Example: galileo_bands("1G")... Default: "1G".
00345     */
00346 void rinex_obs_header(std::fstream& out,
00347     const Gps_CNAV_Ephemeris& gps_cnav_eph,
00348     const Glonass_Gnav_Ephemeris& glonass_gnav_eph,

```

```

00349         double d_TOW_first_observation,
00350         const std::string& glonass_bands = "1G");
00351
00352     /*
00353     * Generates the a Beidou B1I Observation data header. Example: beidou_bands("B1")
00354     */
00355     void rinex_obs_header(std::fstream& out,
00356         const Beidou_Dnav_Ephemeris& eph,
00357         double d_TOW_first_observation,
00358         const std::string& bands);
00359
00360     /*
00361     * Generates the SBAS raw data header
00362     */
00363     void rinex_sbs_header(std::fstream& out) const;
00364
00365     /*
00366     * Writes GPS L1 observables into the RINEX file
00367     */
00368     void log_rinex_obs(std::fstream& out,
00369         const Gps_Ephemeris& eph,
00370         double obs_time,
00371         const std::map<int32_t, Gnss_Synchro>& observables) const;
00372
00373     /*
00374     * Writes GPS L2 observables into the RINEX file
00375     */
00376     void log_rinex_obs(std::fstream& out,
00377         const Gps_CNAV_Ephemeris& eph,
00378         double obs_time,
00379         const std::map<int32_t, Gnss_Synchro>& observables) const;
00380
00381     /*
00382     * Writes dual frequency GPS L1 and L2 observables into the RINEX file
00383     */
00384     void log_rinex_obs(std::fstream& out,
00385         const Gps_Ephemeris& eph,
00386         const Gps_CNAV_Ephemeris& eph_cnav,
00387         double obs_time,
00388         const std::map<int32_t, Gnss_Synchro>& observables,
00389         bool triple_band = false) const;
00390
00391     /*
00392     * Writes Galileo observables into the RINEX file.
00393     * Example: galileo_bands("1B"), galileo_bands("1B 5X"), galileo_bands("5X"), ... Default: "1B".
00394     */
00395     void log_rinex_obs(std::fstream& out,
00396         const Galileo_Ephemeris& eph,
00397         double obs_time,
00398         const std::map<int32_t, Gnss_Synchro>& observables,
00399         const std::string& galileo_bands = "1B") const;
00400
00401     /*
00402     * Writes Mixed GPS / Galileo observables into the RINEX file
00403     */
00404     void log_rinex_obs(std::fstream& out,
00405         const Gps_Ephemeris& gps_eph,
00406         const Galileo_Ephemeris& galileo_eph,
00407         double gps_obs_time,
00408         const std::map<int32_t, Gnss_Synchro>& observables) const;
00409
00410     /*
00411     * Writes Mixed GPS / Galileo observables into the RINEX file
00412     */
00413     void log_rinex_obs(std::fstream& out,
00414         const Gps_CNAV_Ephemeris& eph,
00415         const Galileo_Ephemeris& galileo_eph,
00416         double gps_obs_time,
00417         const std::map<int32_t, Gnss_Synchro>& observables) const;
00418
00419     /*
00420     * Writes Mixed GPS / Galileo observables into the RINEX file
00421     */
00422     void log_rinex_obs(std::fstream& out,
00423         const Gps_Ephemeris& gps_eph,
00424         const Gps_CNAV_Ephemeris& gps_cnav_eph,
00425         const Galileo_Ephemeris& galileo_eph,
00426         double gps_obs_time,
00427         const std::map<int32_t, Gnss_Synchro>& observables,
00428         bool triple_band = false) const;
00429
00430     /*
00431     * Writes GLONASS GNAV observables into the RINEX file.
00432     * Example: glonass_bands("1C"), galileo_bands("1B 5X"), galileo_bands("5X"), ... Default: "1B".
00433     */
00434     void log_rinex_obs(std::fstream& out,
00435         const Glonass_Gnav_Ephemeris& eph,

```

```

00436         double obs_time,
00437         const std::map<int32_t, Gnss_Synchro>& observables,
00438         const std::string& glonass_bands = "1C") const;
00439
00440     /*
00441     * Writes Mixed GPS L1 C/A - GLONASS observables into the RINEX file
00442     */
00443     void log_rinex_obs(std::fstream& out,
00444         const Gps_Ephemeris& gps_eph,
00445         const Glonass_Gnav_Ephemeris& glonass_gnav_eph,
00446         double gps_obs_time,
00447         const std::map<int32_t, Gnss_Synchro>& observables) const;
00448
00449     /*
00450     * Writes Mixed GPS L2C - GLONASS observables into the RINEX file
00451     */
00452     void log_rinex_obs(std::fstream& out,
00453         const Gps_CNAV_Ephemeris& gps_eph,
00454         const Glonass_Gnav_Ephemeris& glonass_gnav_eph,
00455         double gps_obs_time,
00456         const std::map<int32_t, Gnss_Synchro>& observables) const;
00457
00458     /*
00459     * Writes Mixed Galileo/GLONASS observables into the RINEX file
00460     */
00461     void log_rinex_obs(std::fstream& out,
00462         const Galileo_Ephemeris& galileo_eph,
00463         const Glonass_Gnav_Ephemeris& glonass_gnav_eph,
00464         double galileo_obs_time,
00465         const std::map<int32_t, Gnss_Synchro>& observables) const;
00466
00467     /*
00468     * Writes BDS B1I observables into the RINEX file
00469     */
00470     void log_rinex_obs(std::fstream& out,
00471         const Beidou_Dnav_Ephemeris& eph,
00472         double obs_time,
00473         const std::map<int32_t, Gnss_Synchro>& observables,
00474         const std::string& bds_bands) const;
00475
00476     /*
00477     * Generates the GPS L1 C/A Navigation Data header
00478     */
00479     void rinex_nav_header(std::fstream& out,
00480         const Gps_Iono& iono,
00481         const Gps_Utc_Model& utc_model,
00482         const Gps_Ephemeris& eph) const;
00483
00484     /*
00485     * Generates the GPS L2C(M) Navigation Data header
00486     */
00487     void rinex_nav_header(std::fstream& out,
00488         const Gps_CNAV_Iono& iono,
00489         const Gps_CNAV_Utc_Model& utc_model) const;
00490
00491     /*
00492     * Generates the Galileo Navigation Data header
00493     */
00494     void rinex_nav_header(std::fstream& out,
00495         const Galileo_Iono& iono,
00496         const Galileo_Utc_Model& utc_model) const;
00497
00498     /*
00499     * Generates the Mixed (GPS/Galileo) Navigation Data header
00500     */
00501     void rinex_nav_header(std::fstream& out,
00502         const Gps_Iono& gps_iono,
00503         const Gps_Utc_Model& gps_utc_model,
00504         const Gps_Ephemeris& eph,
00505         const Galileo_Iono& galileo_iono,
00506         const Galileo_Utc_Model& galileo_utc_model) const;
00507
00508     /*
00509     * Generates the Mixed (GPS CNAV/Galileo) Navigation Data header
00510     */
00511     void rinex_nav_header(std::fstream& out,
00512         const Gps_CNAV_Iono& iono,
00513         const Gps_CNAV_Utc_Model& utc_model,
00514         const Galileo_Iono& galileo_iono,
00515         const Galileo_Utc_Model& galileo_utc_model) const;
00516
00517     /*
00518     * Generates the GLONASS L1, L2 C/A Navigation Data header
00519     */
00520     void rinex_nav_header(std::fstream& out,
00521         const Glonass_Gnav_Utc_Model& utc_model,
00522         const Glonass_Gnav_Ephemeris& glonass_gnav_eph);

```



```

00523
00524     /*
00525     * Generates the Mixed (Galileo/GLONASS) Navigation Data header
00526     */
00527     void rinex_nav_header(std::fstream& out,
00528         const Galileo_Iono& galileo_iono,
00529         const Galileo_Utc_Model& galileo_utc_model,
00530         const Glonass_Gnav_Utc_Model& glonass_gnav_utc_model,
00531         const Glonass_Gnav_Almanac& glonass_gnav_almanac) const;
00532
00533     /*
00534     * Generates the Mixed (GPS L1 C/A/GLONASS L1, L2) Navigation Data header
00535     */
00536     void rinex_nav_header(std::fstream& out,
00537         const Gps_Iono& gps_iono,
00538         const Gps_Utc_Model& gps_utc_model,
00539         const Gps_Ephemeris& eph,
00540         const Glonass_Gnav_Utc_Model& glonass_gnav_utc_model,
00541         const Glonass_Gnav_Almanac& glonass_gnav_almanac);
00542
00543     /*
00544     * Generates the Mixed (GPS L2C C/A/GLONASS L1, L2) Navigation Data header
00545     */
00546     void rinex_nav_header(std::fstream& out,
00547         const Gps_CNAV_Iono& gps_iono,
00548         const Gps_CNAV_Utc_Model& gps_utc_model,
00549         const Glonass_Gnav_Utc_Model& glonass_gnav_utc_model,
00550         const Glonass_Gnav_Almanac& glonass_gnav_almanac);
00551
00552     /*
00553     * Generates the BDS B1I or B3I Navigation Data header
00554     */
00555     void rinex_nav_header(std::fstream& out,
00556         const Beidou_Dnav_Iono& iono,
00557         const Beidou_Dnav_Utc_Model& utc_model) const;
00558
00559     /*
00560     * Generates the Mixed GPS L1,L5 + BDS B1I, B3I Navigation Data header
00561     */
00562     void rinex_nav_header(std::fstream& out,
00563         const Gps_Iono& gps_iono,
00564         const Gps_Utc_Model& gps_utc_model,
00565         const Gps_Ephemeris& eph,
00566         const Beidou_Dnav_Iono& bds_dnav_iono,
00567         const Beidou_Dnav_Utc_Model& bds_dnav_utc_model) const;
00568
00569     /*
00570     * Generates the Mixed GPS L2C + BDS B1I, B3I Navigation Data header
00571     */
00572     void rinex_nav_header(std::fstream& out,
00573         const Gps_CNAV_Iono& gps_cnav_iono,
00574         const Gps_CNAV_Utc_Model& gps_cnav_utc_model,
00575         const Beidou_Dnav_Iono& bds_dnav_iono,
00576         const Beidou_Dnav_Utc_Model& bds_dnav_utc_model);
00577
00578     /*
00579     * Generates the Mixed GLONASS L1,L2 + BDS B1I, B3I Navigation Data header
00580     */
00581     void rinex_nav_header(std::fstream& out,
00582         const Glonass_Gnav_Utc_Model& glo_gnav_utc_model,
00583         const Beidou_Dnav_Iono& bds_dnav_iono,
00584         const Beidou_Dnav_Utc_Model& bds_dnav_utc_model) const;
00585
00586     /*
00587     * Generates the Mixed (Galileo/BDS B1I, B3I) Navigation Data header
00588     */
00589     void rinex_nav_header(std::fstream& out,
00590         const Galileo_Iono& galileo_iono,
00591         const Galileo_Utc_Model& galileo_utc_model,
00592         const Beidou_Dnav_Iono& bds_dnav_iono,
00593         const Beidou_Dnav_Utc_Model& bds_dnav_utc_model) const;
00594
00595     /*
00596     * Writes data from the GPS L1 C/A navigation message into the RINEX file
00597     */
00598     void log_rinex_nav(std::fstream& out,
00599         const std::map<int32_t, Gps_Ephemeris>& eph_map) const;
00600
00601     /*
00602     * Writes data from the GPS L2 navigation message into the RINEX file
00603     */
00604     void log_rinex_nav(std::fstream& out,
00605         const std::map<int32_t, Gps_CNAV_Ephemeris>& eph_map);
00606
00607     /*
00608     * Writes data from the Galileo navigation message into the RINEX file
00609     */

```

```

00610 void log_rinex_nav(std::fstream& out,
00611     const std::map<int32_t, Galileo_Ephemeris>& eph_map) const;
00612
00613 /*
00614  * Writes data from the Mixed (GPS/Galileo) navigation message into the RINEX file
00615  */
00616 void log_rinex_nav(std::fstream& out,
00617     const std::map<int32_t, Gps_Ephemeris>& gps_eph_map,
00618     const std::map<int32_t, Galileo_Ephemeris>& galileo_eph_map);
00619
00620 /*
00621  * Writes data from the Mixed (GPS/Galileo) navigation message into the RINEX file
00622  */
00623 void log_rinex_nav(std::fstream& out,
00624     const std::map<int32_t, Gps_CNAV_Ephemeris>& gps_cnav_eph_map,
00625     const std::map<int32_t, Galileo_Ephemeris>& galileo_eph_map);
00626
00627 /*
00628  * Writes data from the GLONASS GNAV navigation message into the RINEX file
00629  */
00630 void log_rinex_nav(std::fstream& out,
00631     const std::map<int32_t, Glonass_Gnav_Ephemeris>& eph_map) const;
00632
00633 /*
00634  * Writes data from the Mixed (GPS/GLONASS GNAV) navigation message into the RINEX file
00635  */
00636 void log_rinex_nav(std::fstream& out,
00637     const std::map<int32_t, Gps_Ephemeris>& gps_eph_map,
00638     const std::map<int32_t, Glonass_Gnav_Ephemeris>& glonass_gnav_eph_map) const;
00639
00640 /*
00641  * Writes data from the Mixed (GPS/GLONASS GNAV) navigation message into the RINEX file
00642  */
00643 void log_rinex_nav(std::fstream& out,
00644     const std::map<int32_t, Gps_CNAV_Ephemeris>& gps_cnav_eph_map,
00645     const std::map<int32_t, Glonass_Gnav_Ephemeris>& glonass_gnav_eph_map);
00646
00647 /*
00648  * Writes data from the Mixed (Galileo/ GLONASS GNAV) navigation message into the RINEX file
00649  */
00650 void log_rinex_nav(std::fstream& out,
00651     const std::map<int32_t, Galileo_Ephemeris>& galileo_eph_map,
00652     const std::map<int32_t, Glonass_Gnav_Ephemeris>& glonass_gnav_eph_map);
00653
00654 /*
00655  * Writes data from the Beidou B1I navigation message into the RINEX file
00656  */
00657 void log_rinex_nav(std::fstream& out,
00658     const std::map<int32_t, Beidou_Dnav_Ephemeris>& eph_map) const;
00659
00660 /*
00661  * Computes the BDS Time and returns a boost::posix_time::ptime object
00662  * \details Function used to convert the observation time into BDT time which is used
00663  * as the default time for RINEX files
00664  * \param eph BeiDou DNAV Ephemeris object
00665  * \param obs_time Observation time in BDT seconds of week
00666  */
00667 boost::posix_time::ptime compute_BDS_time(const Beidou_Dnav_Ephemeris& eph, double obs_time)
00668 const;
00669
00670 /*
00671  * Computes the UTC time and returns a boost::posix_time::ptime object
00672  */
00673 boost::posix_time::ptime compute_UTC_time(const Gps_Navigation_Message& nav_msg) const;
00674
00675 /*
00676  * Computes the GPS time and returns a boost::posix_time::ptime object
00677  */
00678 boost::posix_time::ptime compute_GPS_time(const Gps_Ephemeris& eph, double obs_time) const;
00679
00680 /*
00681  * Computes the GPS time and returns a boost::posix_time::ptime object
00682  */
00683 boost::posix_time::ptime compute_GPS_time(const Gps_CNAV_Ephemeris& eph, double obs_time) const;
00684
00685 /*
00686  * Computes the Galileo time and returns a boost::posix_time::ptime object
00687  */
00688 boost::posix_time::ptime compute_Galileo_time(const Galileo_Ephemeris& eph, double obs_time)
00689 const;
00690
00691 /*
00692  * Computes the UTC Time and returns a boost::posix_time::ptime object
00693  * \details Function used as a method to convert the observation time into UTC time which is used
00694  * as the default time for RINEX files
00695  * \param eph GLONASS GNAV Ephemeris object
00696  * \param obs_time Observation time in GPS seconds of week

```

```

00695     */
00696     boost::posix_time::ptime compute_UTC_time(const Glonass_Gnav_Ephemeris& eph, double obs_time)
00697     const;
00698
00699     /*
00700     * Computes number of leap seconds of GPS relative to UTC
00701     * \param eph GLONASS GNAV Ephemeris object
00702     * \param gps_obs_time Observation time in GPS seconds of week
00703     */
00704     double get_leap_second(const Glonass_Gnav_Ephemeris& eph, double gps_obs_time) const;
00705
00706     /*
00707     * Represents GPS time in the date time format. Leap years are considered, but leap seconds are
00708     not.
00709     */
00710     void to_date_time(int gps_week,
00711         int gps_tow,
00712         int& year,
00713         int& month,
00714         int& day,
00715         int& hour,
00716         int& minute,
00717         int& second) const;
00718
00719     /*
00720     * Writes raw SBAS messages into the RINEX file
00721     */
00722     // void log_rinex_sbs(std::fstream & out, const Sbas_Raw_Msg & sbs_message);
00723
00724     void update_nav_header(std::fstream& out,
00725         const Gps_Utc_Model& utc_model,
00726         const Gps_Iono& gps_iono, const Gps_Ephemeris& eph) const;
00727
00728     void update_nav_header(std::fstream& out,
00729         const Gps_CNAV_Utc_Model& utc_model,
00730         const Gps_CNAV_Iono& iono) const;
00731
00732     void update_nav_header(std::fstream& out,
00733         const Gps_Iono& gps_iono,
00734         const Gps_Utc_Model& gps_utc_model,
00735         const Gps_Ephemeris& eph,
00736         const Galileo_Iono& galileo_iono,
00737         const Galileo_Utc_Model& galileo_utc_model) const;
00738
00739     void update_nav_header(std::fstream& out,
00740         const Gps_CNAV_Utc_Model& utc_model,
00741         const Gps_CNAV_Iono& iono,
00742         const Galileo_Iono& galileo_iono,
00743         const Galileo_Utc_Model& galileo_utc_model) const;
00744
00745     void update_nav_header(std::fstream& out,
00746         const Galileo_Iono& galileo_iono,
00747         const Galileo_Utc_Model& utc_model) const;
00748
00749     void update_nav_header(std::fstream& out,
00750         const Gps_Iono& gps_iono,
00751         const Gps_Utc_Model& gps_utc,
00752         const Gps_Ephemeris& eph,
00753         const Glonass_Gnav_Utc_Model& glonass_gnav_utc_model,
00754         const Glonass_Gnav_Almanac& glonass_gnav_almanac) const;
00755
00756     void update_nav_header(std::fstream& out,
00757         const Gps_Iono& gps_iono,
00758         const Gps_Utc_Model& gps_utc,
00759         const Gps_Ephemeris& eph,
00760         const Glonass_Gnav_Utc_Model& glonass_gnav_utc_model,
00761         const Glonass_Gnav_Almanac& glonass_gnav_almanac) const;
00762
00763     void update_nav_header(std::fstream& out,
00764         const Gps_CNAV_Iono& gps_iono,
00765         const Gps_CNAV_Utc_Model& gps_utc_model,
00766         const Glonass_Gnav_Utc_Model& glonass_gnav_utc_model,
00767         const Glonass_Gnav_Almanac& glonass_gnav_almanac) const;
00768
00769     void update_nav_header(std::fstream& out,
00770         const Galileo_Iono& galileo_iono,
00771         const Galileo_Utc_Model& galileo_utc_model,
00772         const Glonass_Gnav_Utc_Model& glonass_gnav_utc_model,
00773         const Glonass_Gnav_Almanac& glonass_gnav_almanac) const;
00774
00775     void update_nav_header(std::fstream& out,
00776         const Beidou_Dnav_Utc_Model& utc_model,
00777         const Beidou_Dnav_Iono& beidou_dnav_iono) const;
00778
00779     void update_obs_header(std::fstream& out,
00780         const Gps_Utc_Model& utc_model) const;
00781
00782     void update_obs_header(std::fstream& out,
00783         const Gps_CNAV_Utc_Model& utc_model) const;
00784
00785

```

```

00780 void update_obs_header(std::fstream& out,
00781     const Galileo_Utc_Model& galileo_utc_model) const;
00782
00783 void update_obs_header(std::fstream& out,
00784     const Glonass_Gnav_Utc_Model& glonass_gnav_utc_model) const;
00785
00786 void update_obs_header(std::fstream& out,
00787     const Beidou_Dnav_Utc_Model& utc_model) const;
00788
00789 /*
00790  * Generation of RINEX signal strength indicators
00791  */
00792 int signalStrength(double snr) const;
00793
00794 /* Creates RINEX file names according to the naming convention
00795  *
00796  * See ftp://igs.org/pub/data/format/rinex301.pdf
00797  * Section 4, page 6
00798  *
00799  * \param[in] type of RINEX file. Can be:
00800  * "RINEX_FILE_TYPE_OBS" - Observation file.
00801  * "RINEX_FILE_TYPE_GPS_NAV" - GPS navigation message file.
00802  * "RINEX_FILE_TYPE_MET" - Meteorological data file.
00803  * "RINEX_FILE_TYPE_GLO_NAV" - GLONASS navigation file.
00804  * "RINEX_FILE_TYPE_GAL_NAV" - Galileo navigation message file.
00805  * "RINEX_FILE_TYPE_MIXED_NAV" - Mixed GNSS navigation message file.
00806  * "RINEX_FILE_TYPE_GEO_NAV" - SBAS Payload navigation message file.
00807  * "RINEX_FILE_TYPE_SBAS" - SBAS broadcast data file.
00808  * "RINEX_FILE_TYPE_CLK" - Clock file.
00809  */
00810 std::string createFilename(const std::string& type, const std::string& base_name) const;
00811
00812 /*
00813  * Generates the data for the PGM / RUN BY / DATE line
00814  */
00815 std::string getLocalTime() const;
00816
00817 /*
00818  * Checks that the line is 80 characters length
00819  */
00820 void lengthCheck(const std::string& line) const;
00821
00822 /*
00823  * If the string is bigger than length, truncate it from the right.
00824  * otherwise, add pad characters to its right.
00825  *
00826  * Left-justifies the input in a string of the specified
00827  * length. If the new length (\a length) is larger than the
00828  * current length, the string is extended by the pad
00829  * character (\a pad). The default pad character is a
00830  * blank.
00831  * \param[in] s string to be modified.
00832  * \param[in] length new desired length of string.
00833  * \param[in] pad character to pad string with (blank by default).
00834  * \return a reference to \a s. */
00835 inline std::string& leftJustify(std::string& s,
00836     std::string::size_type length,
00837     char pad = ' ') const;
00838
00839 /*
00840  * If the string is bigger than length, truncate it from the right.
00841  * otherwise, add pad characters to its right.
00842  *
00843  * Left-justifies the receiver in a string of the specified
00844  * length (const version). If the new length (\a length) is larger
00845  * than the current length, the string is extended by the pad
00846  * character (\a pad). The default pad character is a
00847  * blank.
00848  * \param[in] s string to be modified.
00849  * \param[in] length new desired length of string.
00850  * \param[in] pad character to pad string with (blank by default).
00851  * \return a reference to \a s. */
00852 inline std::string leftJustify(const std::string& s,
00853     std::string::size_type length,
00854     char pad = ' ') const
00855 {
00856     std::string t(s);
00857     return leftJustify(t, length, pad);
00858 }
00859
00860 /*
00861  *
00862  * Right-justifies the receiver in a string of the specified
00863  * length. If the receiver's data is shorter than the
00864  * requested length (\a length), it is padded on the left with
00865  * the pad character (\a pad). The default pad
00866  * character is a blank. */

```

```

00867     inline std::string& rightJustify(std::string& s,
00868         std::string::size_type length,
00869         char pad = ' ') const;
00870
00871     /*
00872     * Right-justifies the receiver in a string of the specified
00873     * length (const version). If the receiver's data is shorter than the
00874     * requested length (\a length), it is padded on the left with
00875     * the pad character (\a pad). The default pad
00876     * character is a blank.*/
00877     inline std::string rightJustify(const std::string& s,
00878         std::string::size_type length,
00879         char pad = ' ') const
00880     {
00881         std::string t(s);
00882         return rightJustify(t, length, pad);
00883     }
00884
00885     /*
00886     * Convert a double to a scientific notation number.
00887     * @param d the double to convert
00888     * @param length length (in characters) of output, including exponent
00889     * @param explen length (in characters) of the exponent, with sign
00890     * @param showSign if true, reserves 1 character for +/- sign
00891     * @param checkSwitch if true, keeps the exponential sanity check for
00892     * exponentials above three characters in length. If false, it removes
00893     * that check.
00894     */
00895     inline std::string doub2sci(double d,
00896         std::string::size_type length,
00897         std::string::size_type explen,
00898         bool showSign = true,
00899         bool checkSwitch = true) const;
00900
00901     /*
00902     * Convert scientific notation to FORTRAN notation.
00903     * As an example, the string "1.5636E5" becomes ".15636D6".
00904     * Note that the first character of the string will be '-' if
00905     * the number is negative or ' ' if the first character is positive.
00906     * @param aStr string with number to convert
00907     * @param startPos start position of number in string
00908     * @param length length (in characters) of number, including exponent.
00909     * @param explen length (in characters) of exponent, not including sign.
00910     * @param checkSwitch will keep the method running as originally programmed
00911     * when set to true. If false, the method will always resize exponentials,
00912     * produce an exponential with an E instead of a D, and always have a leading
00913     * zero. For example -> 0.87654E-0004 or -0.1234E00005.
00914     */
00915     inline std::string& sci2for(std::string& aStr,
00916         std::string::size_type startPos = 0,
00917         std::string::size_type length = std::string::npos,
00918         std::string::size_type explen = 3,
00919         bool checkSwitch = true) const;
00920
00921     /*
00922     * Convert double precision floating point to a string
00923     * containing the number in FORTRAN notation.
00924     * As an example, the number 156360 becomes ".15636D6".
00925     * @param d number to convert.
00926     * @param length length (in characters) of number, including exponent.
00927     * @param explen length (in characters) of exponent, including sign.
00928     * @param checkSwitch if true, keeps the exponential sanity check for
00929     * exponentials above three characters in length. If false, it removes
00930     * that check.
00931     * @return a string containing \a d in FORTRAN notation.
00932     */
00933     inline std::string doub2for(double d,
00934         std::string::size_type length,
00935         std::string::size_type explen,
00936         bool checkSwitch = true) const;
00937
00938     /*
00939     * Convert a string to a double precision floating point number.
00940     * @param s string containing a number.
00941     * @return double representation of string.
00942     */
00943     inline double asDouble(const std::string& s) const
00944     {
00945         return strtod(s.c_str(), nullptr);
00946     }
00947
00948     inline int toInt(const std::string& bitString, int sLength) const;

```

```

00954
00955     /*
00956     * Convert a string to an integer.
00957     * @param s string containing a number.
00958     * @return int64_t integer representation of string.
00959     */
00960     inline int64_t asInt(const std::string& s) const
00961     {
00962         return strtol(s.c_str(), nullptr, 10);
00963     }
00964
00965
00966     /*
00967     * Convert a double to a string in fixed notation.
00968     * @param x double.
00969     * @param precision the number of decimal places you want displayed.
00970     * @return string representation of \a x.
00971     */
00972     inline std::string asString(double x,
00973         std::string::size_type precision = 17) const;
00974
00975
00976     /*
00977     * Convert a long double to a string in fixed notation.
00978     * @param x long double.
00979     * @param precision the number of decimal places you want displayed.
00980     * @return string representation of \a x.
00981     */
00982     inline std::string asString(long double x,
00983         std::string::size_type precision = 21) const;
00984
00985
00986     /*
00987     * Convert any old object to a string.
00988     * The class must have stream operators defined.
00989     * @param x object to turn into a string.
00990     * @return string representation of \a x.
00991     */
00992     template <class X>
00993     inline std::string asString(const X x) const;
00994
00995     inline std::string asFixWidthString(int x, int width, char fill_digit) const;
00996
00997     std::map<std::string, std::string> observationType; // PSEUDORANGE, CARRIER_PHASE, DOPPLER,
00998     SIGNAL_STRENGTH
00999     std::map<std::string, std::string> observationCode; // GNSS observation descriptors
01000
01001     std::fstream obsFile; // Output file stream for RINEX observation file
01002     std::fstream navFile; // Output file stream for RINEX navigation data file
01003     std::fstream sbsFile; // Output file stream for RINEX SBAS raw data file
01004     std::fstream navGalFile; // Output file stream for RINEX Galileo navigation data file
01005     std::fstream navGloFile; // Output file stream for RINEX GLONASS navigation data file
01006     std::fstream navBdsFile; // Output file stream for RINEX Galileo navigation data file
01007     std::fstream navMixFile; // Output file stream for RINEX Mixed navigation data file
01008
01009     std::string navfilename; // Name of RINEX navigation file for GPS L1
01010     std::string obsfilename; // Name of RINEX observation file
01011     std::string sbsfilename; // Name of RINEX SBAS file
01012     std::string navGalfilename; // Name of RINEX navigation file for Galileo
01013     std::string navGlofilename; // Name of RINEX navigation file for Glonass
01014     std::string navBdsfilename; // Name of RINEX navigation file for BeiDou
01015     std::string navMixfilename; // Name of RINEX navigation file for fixed signals
01016     std::vector<std::string> output_navfilename; // Name of output RINEX navigation file(s)
01017
01018     std::string d_stringVersion; // RINEX version (2.10/2.11 or 3.01/3.02)
01019
01020     double d_fake_cnav_iode;
01021     int d_version; // RINEX version (2 for 2.10/2.11 and 3 for 3.01)
01022     int d_numberTypesObservations; // Number of available types of observable in the system. Should
01023     be public?
01024     bool d_rinex_header_updated;
01025     bool d_rinex_header_written;
01026     bool d_pre_2009_file;
01027 };
01028
01029 // Implementation of inline functions (modified versions from GNSSTk https://github.com/SGL-UT/gnsstk)
01030
01031 inline std::string& Rinex_Printer::leftJustify(std::string& s,
01032     std::string::size_type length,
01033     char pad) const
01034 {
01035     if (length < s.length())
01036     {
01037         s = s.substr(0, length);
01038     }
01039     else

```

```

01039     {
01040         s.append(length - s.length(), pad);
01041     }
01042     return s;
01043 }
01044
01045
01046 // if the string is bigger than length, truncate it from the left.
01047 // otherwise, add pad characters to its left.
01048 inline std::string& Rinex_Printer::rightJustify(std::string& s,
01049     std::string::size_type length,
01050     char pad) const
01051 {
01052     if (length < s.length())
01053     {
01054         s = s.substr(s.length() - length, std::string::npos);
01055     }
01056     else
01057     {
01058         s.insert(static_cast<std::string::size_type>(0), length - s.length(), pad);
01059     }
01060     return s;
01061 }
01062
01063
01064 inline std::string Rinex_Printer::doub2for(double d,
01065     std::string::size_type length,
01066     std::string::size_type expLen,
01067     bool checkSwitch) const
01068 {
01069     int16_t exponentLength = expLen;
01070
01071     /* Validate the assumptions regarding the input arguments */
01072     if (exponentLength < 0)
01073     {
01074         exponentLength = 1;
01075     }
01076
01077     if (exponentLength > 3 && checkSwitch)
01078     {
01079         exponentLength = 3;
01080     }
01081
01082     std::string toReturn = doub2sci(d, length, exponentLength, true, checkSwitch);
01083     sci2for(toReturn, 0, length, exponentLength, checkSwitch);
01084
01085     return toReturn;
01086 }
01087
01088
01089 inline std::string Rinex_Printer::doub2sci(double d,
01090     std::string::size_type length,
01091     std::string::size_type expLen,
01092     bool showSign,
01093     bool checkSwitch) const
01094 {
01095     std::string toReturn;
01096     int16_t exponentLength = expLen;
01097
01098     /* Validate the assumptions regarding the input arguments */
01099     if (exponentLength < 0)
01100     {
01101         exponentLength = 1;
01102     }
01103
01104     if (exponentLength > 3 && checkSwitch)
01105     {
01106         exponentLength = 3;
01107     }
01108
01109     std::stringstream c;
01110     c.setf(std::ios::scientific, std::ios::floatfield);
01111
01112     // length - 3 for special characters ('.', 'e', '+' or '-')
01113     // - exponentlength (e04)
01114     // - 1 for the digit before the decimal (2.)
01115     // and if showSign == true,
01116     //   an extra -1 for '-' or ' ' if it's positive or negative
01117     int expSize = 0;
01118     if (showSign)
01119     {
01120         expSize = 1;
01121     }
01122
01123     c.precision(length - 3 - exponentLength - 1 - expSize);
01124     c << d;
01125     c >> toReturn;

```

```

01126     return toReturn;
01127 }
01128
01129
01130 inline std::string& Rinex_Printer::sci2for(std::string& aStr,
01131     std::string::size_type startPos,
01132     std::string::size_type length,
01133     std::string::size_type expLen,
01134     bool checkSwitch) const
01135 {
01136     std::string::size_type idx = aStr.find('.', startPos);
01137     int expAdd = 0;
01138     std::string exp;
01139     int64_t iexp;
01140     // If checkSwitch is false, always redo the exponential. Otherwise,
01141     // set it to false.
01142     bool redoexp = !checkSwitch;
01143
01144     // Check for decimal place within specified boundaries
01145     if ((idx <= 0) || (idx >= (startPos + length - expLen - 1)))
01146     {
01147         // Error: no decimal point in string
01148         return aStr;
01149     }
01150
01151     // Here, account for the possibility that there are
01152     // no numbers to the left of the decimal, but do not
01153     // account for the possibility of non-scientific
01154     // notation (more than one digit to the left of the
01155     // decimal)
01156     if (idx > startPos)
01157     {
01158         redoexp = true;
01159         // Swap digit and decimal.
01160         aStr[idx] = aStr[idx - 1];
01161         aStr[idx - 1] = '.';
01162         // Only add one to the exponent if the number is non-zero
01163         if (asDouble(aStr.substr(startPos, length)) != 0.0)
01164         {
01165             expAdd = 1;
01166         }
01167     }
01168
01169     idx = aStr.find('e', startPos);
01170     if (idx == std::string::npos)
01171     {
01172         idx = aStr.find('E', startPos);
01173         if (idx == std::string::npos)
01174         {
01175             // Error: no 'e' or 'E' in string";
01176         }
01177     }
01178
01179     // Change the exponent character to D normally, or E if checkSwitch is false.
01180     if (checkSwitch)
01181     {
01182         aStr[idx] = 'D';
01183     }
01184     else
01185     {
01186         aStr[idx] = 'E';
01187     }
01188
01189     // Change the exponent itself
01190     if (redoexp)
01191     {
01192         exp = aStr.substr(idx + 1, std::string::npos);
01193         iexp = asInt(exp);
01194         iexp += expAdd;
01195
01196         aStr.erase(idx + 1);
01197         if (iexp < 0)
01198         {
01199             aStr += "-";
01200             iexp -= iexp * 2;
01201         }
01202         else
01203         {
01204             aStr += "+";
01205         }
01206
01207         aStr += Rinex_Printer::rightJustify(asString(iexp), expLen, '0');
01208     }
01209
01210     // if the number is positive, append a space
01211     // (if it's negative, there's a leading '-')
01212     if (aStr[0] == '.')

```



```

01213     {
01214         aStr.insert(static_cast<std::string::size_type>(0), 1, ' ');
01215     }
01216
01217     // If checkSwitch is false, add on one leading zero to the string
01218     if (!checkSwitch)
01219     {
01220         aStr.insert(static_cast<std::string::size_type>(1), 1, '0');
01221     }
01222
01223     return aStr;
01224 } // end sci2for
01225
01226
01227 inline std::string asString(long double x, std::string::size_type precision)
01228 {
01229     std::ostringstream ss;
01230     ss << std::fixed << std::setprecision(precision) << x;
01231     return ss.str();
01232 }
01233
01234
01235 inline std::string Rinex_Printer::asString(double x, std::string::size_type precision) const
01236 {
01237     std::ostringstream ss;
01238     ss << std::fixed << std::setprecision(precision) << x;
01239     return ss.str();
01240 }
01241
01242
01243 inline std::string Rinex_Printer::asFixWidthString(int x, int width, char fill_digit) const
01244 {
01245     std::ostringstream ss;
01246     ss << std::setfill(fill_digit) << std::setw(width) << x;
01247     return ss.str().substr(ss.str().size() - width);
01248 }
01249
01250
01251 inline int64_t asInt(const std::string& s)
01252 {
01253     return strtol(s.c_str(), nullptr, 10);
01254 }
01255
01256
01257 inline int Rinex_Printer::toInt(const std::string& bitString, int sLength) const
01258 {
01259     int tempInt;
01260     int num = 0;
01261     for (int i = 0; i < sLength; i++)
01262     {
01263         tempInt = bitString[i] - '0';
01264         num |= (1 << (sLength - 1 - i)) * tempInt;
01265     }
01266     return num;
01267 }
01268
01269
01270 template <class X>
01271 inline std::string Rinex_Printer::asString(const X x) const
01272 {
01273     std::ostringstream ss;
01274     ss << x;
01275     return ss.str();
01276 }
01277
01278
01279 /** \} */
01280 /** \} */
01281 #endif // GNSS_SDR_RINEX_PRINTER_H

```

## 11.274 rtcm.h File Reference

Interface for the RTCM 3.2 Standard.

```

#include "concurrent_queue.h"
#include "galileo_ephemeris.h"
#include "galileo_has_data.h"
#include "glonass_gnav_ephemeris.h"
#include "glonass_gnav_utc_model.h"
#include "gnss_synchro.h"
#include "gps_cnav_ephemeris.h"

```

```
#include "gps_ephemeris.h"
#include <boost/asio.hpp>
#include <boost/date_time/posix_time/posix_time.hpp>
#include <algorithm>
#include <array>
#include <bitset>
#include <cstdint>
#include <stdint>
#include <deque>
#include <iomanip>
#include <list>
#include <map>
#include <memory>
#include <set>
#include <sstream>
#include <string>
#include <thread>
#include <utility>
#include <vector>
#include <absl/log/log.h>
```

Include dependency graph for `rtcm.h`:



## Classes

- class [Rtcm](#)

*This class implements the generation and reading of some Message Types defined in the RTCM 3.2 Standard, plus some utilities to handle messages.*

## Typedefs

- using [b\\_io\\_context](#) = `boost::asio::io_service`

## 11.274.1 Detailed Description

Interface for the RTCM 3.2 Standard.

### Author

Carles Fernandez-Prades, 2015. [cfernandez\(at\)cttc.es](mailto:cfernandez(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [rtcm.h](#).

---

## 11.275 rtcm.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file rtcm.h
00003  * \brief Interface for the RTCM 3.2 Standard
00004  * \author Carles Fernandez-Prades, 2015. cfernandez(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
```

```

00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_RTCM_H
00019 #define GNSS_SDR_RTCM_H
00020
00021
00022 #include "concurrent_queue.h"
00023 #include "galileo_ephemeris.h"
00024 #include "galileo_has_data.h"
00025 #include "glonass_gnav_ephemeris.h"
00026 #include "glonass_gnav_utc_model.h"
00027 #include "gnss_synchro.h"
00028 #include "gps_cnav_ephemeris.h"
00029 #include "gps_ephemeris.h"
00030 #include <boost/asio.hpp>
00031 #include <boost/date_time/posix_time/posix_time.hpp>
00032 #include <algorithm> // for std::max, std::min, std::copy_n
00033 #include <array>
00034 #include <bitset>
00035 #include <cstdint> // for size_t
00036 #include <cstdint>
00037 #include <deque>
00038 #include <iomanip> // for std::setw
00039 #include <list>
00040 #include <map>
00041 #include <memory>
00042 #include <set>
00043 #include <sstream> // for std::stringstream
00044 #include <string>
00045 #include <thread>
00046 #include <utility>
00047 #include <vector>
00048
00049 #if USE_GLOG_AND_GFLAGS
00050 #include <glog/logging.h>
00051 #else
00052 #include <absl/log/log.h>
00053 #endif
00054
00055 /** \addtogroup PVT
00056  * \{ */
00057 /** \addtogroup PVT_libs
00058  * \{ */
00059
00060
00061 #if USE_BOOST_ASIO_IO_CONTEXT
00062 using b_io_context = boost::asio::io_context;
00063 #else
00064 using b_io_context = boost::asio::io_service;
00065 #endif
00066
00067
00068 /*!
00069  * \brief This class implements the generation and reading of some Message Types
00070  * defined in the RTCM 3.2 Standard, plus some utilities to handle messages.
00071  *
00072  * Generation of the following Message Types:
00073  * 1001, 1002, 1003, 1004, 1005, 1006, 1008, 1019, 1020, 1029, 1045
00074  *
00075  * Decoding of the following Message Types:
00076  * 1019, 1045
00077  *
00078  * Generation of the following Multiple Signal Messages:
00079  * MSM1 (message types 1071, 1091)
00080  * MSM2 (message types 1072, 1092)
00081  * MSM3 (message types 1073, 1093)
00082  * MSM4 (message types 1074, 1094)
00083  * MSM5 (message types 1075, 1095)
00084  * MSM6 (message types 1076, 1096)
00085  * MSM7 (message types 1077, 1097)
00086  *
00087  * RTCM 3 message format (size in bits):
00088  * +-----+-----+-----+-----+-----+
00089  * | preamble | 000000 | length | data message | parity |
00090  * +-----+-----+-----+-----+-----+
00091  * |<-- 8 --->|<-- 6 --->|<-- 10 --->|<-- length x 8 --->|<-- 24 --->|
00092  * +-----+-----+-----+-----+-----+
00093  *
00094  *
00095  * (C) Carles Fernandez-Prades, 2015. cfernandez(at)cttc.es
00096  */
00097 class Rtcm

```

```

00098 {
00099 public:
00100     explicit RtcM(uint16_t port = 2101); ///< Default constructor that sets TCP port of the RTCM
message server and RTCM Station ID. 2101 is the standard RTCM port according to the Internet Assigned
Numbers Authority (IANA). See
https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml
00101     ~RtcM();
00102
00103     /*!
00104      * \brief Prints message type 1001 (L1-Only GPS RTK Observables)
00105      */
00106     std::string print_MT1001(const Gps_Ephemeris& gps_eph, double obs_time, const std::map<int32_t,
Gnss_Synchro>& observables, uint16_t station_id);
00107
00108     /*!
00109      * \brief Prints message type 1002 (Extended L1-Only GPS RTK Observables)
00110      */
00111     std::string print_MT1002(const Gps_Ephemeris& gps_eph, double obs_time, const std::map<int32_t,
Gnss_Synchro>& observables, uint16_t station_id);
00112
00113     /*!
00114      * \brief Prints message type 1003 (L1 & L2 GPS RTK Observables)
00115      */
00116     std::string print_MT1003(const Gps_Ephemeris& ephL1, const Gps_CNAV_Ephemeris& ephL2, double
obs_time, const std::map<int32_t, Gnss_Synchro>& observables, uint16_t station_id);
00117
00118     /*!
00119      * \brief Prints message type 1004 (Extended L1 & L2 GPS RTK Observables)
00120      */
00121     std::string print_MT1004(const Gps_Ephemeris& ephL1, const Gps_CNAV_Ephemeris& ephL2, double
obs_time, const std::map<int32_t, Gnss_Synchro>& observables, uint16_t station_id);
00122
00123     /*!
00124      * \brief Prints message type 1005 (Stationary Antenna Reference Point)
00125      */
00126     std::string print_MT1005(uint32_t ref_id, double ecef_x, double ecef_y, double ecef_z, bool gps,
bool glonass, bool galileo, bool non_physical, bool single_oscillator, uint32_t
quarter_cycle_indicator);
00127
00128     /*!
00129      * \brief Verifies and reads messages of type 1005 (Stationary Antenna Reference Point). Returns 1
if anything goes wrong, 0 otherwise.
00130      */
00131     int32_t read_MT1005(const std::string& message, uint32_t& ref_id, double& ecef_x, double& ecef_y,
double& ecef_z, bool& gps, bool& glonass, bool& galileo);
00132
00133     /*!
00134      * \brief Prints message type 1006 (Stationary Antenna Reference Point, with Height Information)
00135      */
00136     std::string print_MT1006(uint32_t ref_id, double ecef_x, double ecef_y, double ecef_z, bool gps,
bool glonass, bool galileo, bool non_physical, bool single_oscillator, uint32_t
quarter_cycle_indicator, double height);
00137
00138     std::string print_MT1005_test(); ///< For testing purposes
00139
00140     /*!
00141      * \brief Prints message type 1008 (Antenna Descriptor & Serial Number)
00142      */
00143     std::string print_MT1008(uint32_t ref_id, const std::string& antenna_descriptor, uint32_t
antenna_setup_id, const std::string& antenna_serial_number);
00144
00145     /*!
00146      * \brief Prints L1-Only GLONASS RTK Observables
00147      * \details This GLONASS message type is not generally used or supported; type 1012 is to be
preferred.
00148      * \note Code added as part of GSoC 2017 program
00149      * \param glonass_gnav_eph GLONASS GNAV Broadcast Ephemeris
00150      * \param obs_time Time of observation at the moment of printing
00151      * \param observables Set of observables as defined by the platform
00152      * \return string with message contents
00153      */
00154     std::string print_MT1009(const Glonass_Gnav_Ephemeris& glonass_gnav_eph, double obs_time, const
std::map<int32_t, Gnss_Synchro>& observables, uint16_t station_id);
00155
00156     /*!
00157      * \brief Prints Extended L1-Only GLONASS RTK Observables
00158      * \details This GLONASS message type is used when only L1 data is present and bandwidth is very
tight, often 1012 is used in such cases.
00159      * \note Code added as part of GSoC 2017 program
00160      * \param glonass_gnav_eph GLONASS GNAV Broadcast Ephemeris
00161      * \param obs_time Time of observation at the moment of printing
00162      * \param observables Set of observables as defined by the platform
00163      * \return string with message contents
00164      */
00165     std::string print_MT1010(const Glonass_Gnav_Ephemeris& glonass_gnav_eph, double obs_time, const
std::map<int32_t, Gnss_Synchro>& observables, uint16_t station_id);
00166

```

```

00167     /*!
00168     * \brief Prints L1&L2 GLONASS RTK Observables
00169     * \details This GLONASS message type is not generally used or supported; type 1012 is to be
    preferred
00170     * \note Code added as part of GSoC 2017 program
00171     * \param glonass_gnav_eph GLONASS GNAV Broadcast Ephemeris
00172     * \param obs_time Time of observation at the moment of printing
00173     * \param observables Set of observables as defined by the platform
00174     * \return string with message contents
00175     */
00176     std::string print_MT1011(const Glonass_Gnav_Ephemeris& glonass_gnav_ephL1, const
    Glonass_Gnav_Ephemeris& glonass_gnav_ephL2, double obs_time, const std::map<int32_t, Gnss_Synchro>&
    observables, uint16_t station_id);
00177
00178     /*!
00179     * \brief Prints Extended L1&L2 GLONASS RTK Observables
00180     * \details This GLONASS message type is the most common observational message type, with
    L1/L2/SNR content. This is one of the most common messages found.
00181     * \note Code added as part of GSoC 2017 program
00182     * \param glonass_gnav_eph GLONASS GNAV Broadcast Ephemeris
00183     * \param obs_time Time of observation at the moment of printing
00184     * \param observables Set of observables as defined by the platform
00185     * \return string with message contents
00186     */
00187     std::string print_MT1012(const Glonass_Gnav_Ephemeris& glonass_gnav_ephL1, const
    Glonass_Gnav_Ephemeris& glonass_gnav_ephL2, double obs_time, const std::map<int32_t, Gnss_Synchro>&
    observables, uint16_t station_id);
00188
00189     /*!
00190     * \brief Prints message type 1019 (GPS Ephemeris), should be broadcast in the event that
00191     * the IODC does not match the IODE, and every 2 minutes.
00192     */
00193     std::string print_MT1019(const Gps_Ephemeris& gps_eph);
00194
00195     /*!
00196     * \brief Verifies and reads messages of type 1019 (GPS Ephemeris). Returns 1 if anything goes
    wrong, 0 otherwise.
00197     */
00198     int32_t read_MT1019(const std::string& message, Gps_Ephemeris& gps_eph) const;
00199
00200     /*!
00201     * \brief Prints message type 1020 (GLONASS Ephemeris).
00202     * \note Code added as part of GSoC 2017 program
00203     * \param glonass_gnav_eph GLONASS GNAV Broadcast Ephemeris
00204     * \param glonass_gnav_utc_model GLONASS GNAV Clock Information
00205     * \return Returns message type as a string type
00206     */
00207     std::string print_MT1020(const Glonass_Gnav_Ephemeris& glonass_gnav_eph, const
    Glonass_Gnav_Utc_Model& glonass_gnav_utc_model);
00208
00209     /*!
00210     * \brief Verifies and reads messages of type 1020 (GLONASS Ephemeris).
00211     * \note Code added as part of GSoC 2017 program
00212     * \param message Message to read as a string type
00213     * \param glonass_gnav_eph GLONASS GNAV Broadcast Ephemeris
00214     * \param glonass_gnav_utc_model GLONASS GNAV Clock Information
00215     * \return Returns 1 if anything goes wrong, 0 otherwise.
00216     */
00217     int32_t read_MT1020(const std::string& message, Glonass_Gnav_Ephemeris& glonass_gnav_eph,
    Glonass_Gnav_Utc_Model& glonass_gnav_utc_model) const;
00218
00219     /*!
00220     * \brief Prints message type 1029 (Unicode Text String)
00221     */
00222     std::string print_MT1029(uint32_t ref_id, const Gps_Ephemeris& gps_eph, double obs_time, const
    std::string& message);
00223
00224     /*!
00225     * \brief Prints message type 1045 (Galileo Ephemeris), should be broadcast every 2 minutes
00226     */
00227     std::string print_MT1045(const Galileo_Ephemeris& gal_eph);
00228
00229     /*!
00230     * \brief Verifies and reads messages of type 1045 (Galileo Ephemeris). Returns 1 if anything goes
    wrong, 0 otherwise.
00231     */
00232     int32_t read_MT1045(const std::string& message, Galileo_Ephemeris& gal_eph) const;
00233
00234     /*!
00235     * \brief Prints messages of type MSM1 (Compact GNSS observables)
00236     */
00237     std::string print_MSM_1(const Gps_Ephemeris& gps_eph,
    const Gps_CNAV_Ephemeris& gps_cnav_eph,
    const Galileo_Ephemeris& gal_eph,
    const Glonass_Gnav_Ephemeris& glo_gnav_eph,
    double obs_time,
    const std::map<int32_t, Gnss_Synchro>& observables,

```

```

00243         uint32_t ref_id,
00244         uint32_t clock_steering_indicator,
00245         uint32_t external_clock_indicator,
00246         int32_t smooth_int,
00247         bool divergence_free,
00248         bool more_messages);
00249
00250     /*!
00251     * \brief Prints messages of type MSM2 (Compact GNSS phaseranges)
00252     */
00253     std::string print_MSM_2(const Gps_Ephemeris& gps_eph,
00254         const Gps_CNAV_Ephemeris& gps_cnav_eph,
00255         const Galileo_Ephemeris& gal_eph,
00256         const Glonass_Gnav_Ephemeris& glo_gnav_eph,
00257         double obs_time,
00258         const std::map<int32_t, Gnss_Synchro>& observables,
00259         uint32_t ref_id,
00260         uint32_t clock_steering_indicator,
00261         uint32_t external_clock_indicator,
00262         int32_t smooth_int,
00263         bool divergence_free,
00264         bool more_messages);
00265
00266     /*!
00267     * \brief Prints messages of type MSM3 (Compact GNSS pseudoranges and phaseranges)
00268     */
00269     std::string print_MSM_3(const Gps_Ephemeris& gps_eph,
00270         const Gps_CNAV_Ephemeris& gps_cnav_eph,
00271         const Galileo_Ephemeris& gal_eph,
00272         const Glonass_Gnav_Ephemeris& glo_gnav_eph,
00273         double obs_time,
00274         const std::map<int32_t, Gnss_Synchro>& observables,
00275         uint32_t ref_id,
00276         uint32_t clock_steering_indicator,
00277         uint32_t external_clock_indicator,
00278         int32_t smooth_int,
00279         bool divergence_free,
00280         bool more_messages);
00281
00282     /*!
00283     * \brief Prints messages of type MSM4 (Full GNSS pseudoranges and phaseranges plus CNR)
00284     */
00285     std::string print_MSM_4(const Gps_Ephemeris& gps_eph,
00286         const Gps_CNAV_Ephemeris& gps_cnav_eph,
00287         const Galileo_Ephemeris& gal_eph,
00288         const Glonass_Gnav_Ephemeris& glo_gnav_eph,
00289         double obs_time,
00290         const std::map<int32_t, Gnss_Synchro>& observables,
00291         uint32_t ref_id,
00292         uint32_t clock_steering_indicator,
00293         uint32_t external_clock_indicator,
00294         int32_t smooth_int,
00295         bool divergence_free,
00296         bool more_messages);
00297
00298     /*!
00299     * \brief Prints messages of type MSM5 (Full GNSS pseudoranges, phaseranges, phaserange rate and
00300     CNR)
00301     */
00301     std::string print_MSM_5(const Gps_Ephemeris& gps_eph,
00302         const Gps_CNAV_Ephemeris& gps_cnav_eph,
00303         const Galileo_Ephemeris& gal_eph,
00304         const Glonass_Gnav_Ephemeris& glo_gnav_eph,
00305         double obs_time,
00306         const std::map<int32_t, Gnss_Synchro>& observables,
00307         uint32_t ref_id,
00308         uint32_t clock_steering_indicator,
00309         uint32_t external_clock_indicator,
00310         int32_t smooth_int,
00311         bool divergence_free,
00312         bool more_messages);
00313
00314     /*!
00315     * \brief Prints messages of type MSM6 (Full GNSS pseudoranges and phaseranges plus CNR, high
00316     resolution)
00317     */
00317     std::string print_MSM_6(const Gps_Ephemeris& gps_eph,
00318         const Gps_CNAV_Ephemeris& gps_cnav_eph,
00319         const Galileo_Ephemeris& gal_eph,
00320         const Glonass_Gnav_Ephemeris& glo_gnav_eph,
00321         double obs_time,
00322         const std::map<int32_t, Gnss_Synchro>& observables,
00323         uint32_t ref_id,
00324         uint32_t clock_steering_indicator,
00325         uint32_t external_clock_indicator,
00326         int32_t smooth_int,
00327         bool divergence_free,

```



```

        otherwise
00400
00401 private:
00402     //
00403     // Generation of messages content
00404     //
00405     std::bitset<64> get_MT1001_4_header(uint32_t msg_number,
00406         double obs_time,
00407         const std::map<int32_t, Gnss_Synchro>& observables,
00408         uint32_t ref_id,
00409         uint32_t smooth_int,
00410         bool sync_flag,
00411         bool divergence_free);
00412
00413     std::bitset<58> get_MT1001_sat_content(const Gps_Ephemeris& eph, double obs_time, const
00414 Gnss_Synchro& gnss_synchro);
00415     std::bitset<74> get_MT1002_sat_content(const Gps_Ephemeris& eph, double obs_time, const
00416 Gnss_Synchro& gnss_synchro);
00417     std::bitset<101> get_MT1003_sat_content(const Gps_Ephemeris& ephL1, const Gps_CNAV_Ephemeris&
00418 ephL2, double obs_time, const Gnss_Synchro& gnss_synchroL1, const Gnss_Synchro& gnss_synchroL2);
00419     std::bitset<125> get_MT1004_sat_content(const Gps_Ephemeris& ephL1, const Gps_CNAV_Ephemeris&
00420 ephL2, double obs_time, const Gnss_Synchro& gnss_synchroL1, const Gnss_Synchro& gnss_synchroL2);
00421
00422     std::bitset<152> get_MT1005_test();
00423
00424     /*!
00425      * \brief Generates contents of message header for types 1009, 1010, 1011 and 1012. GLONASS RTK
00426      * Message
00427      * \note Code added as part of GSoC 2017 program
00428      * \param msg_number Message type number, acceptable options include 1009 to 1012
00429      * \param obs_time Time of observation at the moment of printing
00430      * \param observables Set of observables as defined by the platform
00431      * \param ref_id
00432      * \param smooth_int
00433      * \param divergence_free
00434      * \return Returns the message header content as set of bits
00435      */
00436     std::bitset<61> get_MT1009_12_header(uint32_t msg_number,
00437         double obs_time,
00438         const std::map<int32_t, Gnss_Synchro>& observables,
00439         uint32_t ref_id,
00440         uint32_t smooth_int,
00441         bool sync_flag,
00442         bool divergence_free);
00443
00444     /*!
00445      * \brief Get the contents of the satellite specific portion of a type 1009 Message (GLONASS Basic
00446      * RTK, L1 Only)
00447      * \details Contents generated for each satellite. See table 3.5-11
00448      * \note Code added as part of GSoC 2017 program
00449      * \param ephGNAV Ephemeris for GLONASS GNAV in L1 satellites
00450      * \param obs_time Time of observation at the moment of printing
00451      * \param gnss_synchro Information generated by channels while processing the satellite
00452      * \return Returns the message content as set of bits
00453      */
00454     std::bitset<64> get_MT1009_sat_content(const Glonass_Gnav_Ephemeris& ephGNAV, double obs_time,
00455 const Gnss_Synchro& gnss_synchro);
00456
00457     /*!
00458      * \brief Get the contents of the satellite specific portion of a type 1010 Message (GLONASS
00459      * Extended RTK, L1 Only)
00460      * \details Contents generated for each satellite. See table 3.5-12
00461      * \note Code added as part of GSoC 2017 program
00462      * \param ephGNAV Ephemeris for GLONASS GNAV in L1 satellites
00463      * \param obs_time Time of observation at the moment of printing
00464      * \param gnss_synchro Information generated by channels while processing the satellite
00465      * \return Returns the message content as set of bits
00466      */
00467     std::bitset<79> get_MT1010_sat_content(const Glonass_Gnav_Ephemeris& ephGNAV, double obs_time,
00468 const Gnss_Synchro& gnss_synchro);
00469
00470     /*!
00471      * \brief Get the contents of the satellite specific portion of a type 1011 Message (GLONASS Basic
00472      * RTK, L1 & L2)
00473      * \details Contents generated for each satellite. See table 3.5-13
00474      * \note Code added as part of GSoC 2017 program
00475      * \param ephGNAV L1 Ephemeris for GLONASS GNAV in L1 satellites
00476      * \param ephGNAV L2 Ephemeris for GLONASS GNAV in L2 satellites
00477      * \param obs_time Time of observation at the moment of printing
00478      * \param gnss_synchroL1 Information generated by channels while processing the GLONASS GNAV L1
00479      * satellite
00480      * \param gnss_synchroL2 Information generated by channels while processing the GLONASS GNAV L2
00481      * satellite
00482      * \return Returns the message content as set of bits
00483      */
00484     std::bitset<107> get_MT1011_sat_content(const Glonass_Gnav_Ephemeris& ephL1, const
00485 Glonass_Gnav_Ephemeris& ephL2, double obs_time, const Gnss_Synchro& gnss_synchroL1, const
00486 Gnss_Synchro& gnss_synchroL2);
00487
00488     /*!

```



```

00472     * \brief Get the contents of the satellite specific portion of a type 1012 Message (GLONASS
Extended RTK, L1 & L2)
00473     * \details Contents generated for each satellite. See table 3.5-14
00474     * \note Code added as part of GSoc 2017 program
00475     * \param ephGNAV1 Ephemeris for GLONASS GNAV in L1 satellites
00476     * \param ephGNAV2 Ephemeris for GLONASS GNAV in L2 satellites
00477     * \param obs_time Time of observation at the moment of printing
00478     * \param gnss_synchroL1 Information generated by channels while processing the GLONASS GNAV L1
satellite
00479     * \param gnss_synchroL2 Information generated by channels while processing the GLONASS GNAV L2
satellite
00480     * \return Returns the message content as set of bits
00481     */
00482     std::bitset<130> get_MT1012_sat_content(const Glonass_Gnav_Ephemeris& ephL1, const
Glonass_Gnav_Ephemeris& ephL2, double obs_time, const Gnss_Synchro& gnss_synchroL1, const
Gnss_Synchro& gnss_synchroL2);

00483
00484     std::string get_MSM_header(uint32_t msg_number,
00485                               double obs_time,
00486                               const std::map<int32_t, Gnss_Synchro>& observables,
00487                               uint32_t ref_id,
00488                               uint32_t clock_steering_indicator,
00489                               uint32_t external_clock_indicator,
00490                               int32_t smooth_int,
00491                               bool divergence_free,
00492                               bool more_messages);
00493
00494     std::string get_MSM_1_content_sat_data(const std::map<int32_t, Gnss_Synchro>& observables);
00495     std::string get_MSM_4_content_sat_data(const std::map<int32_t, Gnss_Synchro>& observables);
00496     std::string get_MSM_5_content_sat_data(const std::map<int32_t, Gnss_Synchro>& observables);
00497
00498     std::string get_MSM_1_content_signal_data(const std::map<int32_t, Gnss_Synchro>& observables);
00499     std::string get_MSM_2_content_signal_data(const Gps_Ephemeris& ephNAV, const Gps_CNAV_Ephemeris&
ephCNAV, const Galileo_Ephemeris& ephFNAV, const Glonass_Gnav_Ephemeris& ephGNAV, double obs_time,
const std::map<int32_t, Gnss_Synchro>& observables);
00500     std::string get_MSM_3_content_signal_data(const Gps_Ephemeris& ephNAV, const Gps_CNAV_Ephemeris&
ephCNAV, const Galileo_Ephemeris& ephFNAV, const Glonass_Gnav_Ephemeris& ephGNAV, double obs_time,
const std::map<int32_t, Gnss_Synchro>& observables);
00501     std::string get_MSM_4_content_signal_data(const Gps_Ephemeris& ephNAV, const Gps_CNAV_Ephemeris&
ephCNAV, const Galileo_Ephemeris& ephFNAV, const Glonass_Gnav_Ephemeris& ephGNAV, double obs_time,
const std::map<int32_t, Gnss_Synchro>& observables);
00502     std::string get_MSM_5_content_signal_data(const Gps_Ephemeris& ephNAV, const Gps_CNAV_Ephemeris&
ephCNAV, const Galileo_Ephemeris& ephFNAV, const Glonass_Gnav_Ephemeris& ephGNAV, double obs_time,
const std::map<int32_t, Gnss_Synchro>& observables);
00503     std::string get_MSM_6_content_signal_data(const Gps_Ephemeris& ephNAV, const Gps_CNAV_Ephemeris&
ephCNAV, const Galileo_Ephemeris& ephFNAV, const Glonass_Gnav_Ephemeris& ephGNAV, double obs_time,
const std::map<int32_t, Gnss_Synchro>& observables);
00504     std::string get_MSM_7_content_signal_data(const Gps_Ephemeris& ephNAV, const Gps_CNAV_Ephemeris&
ephCNAV, const Galileo_Ephemeris& ephFNAV, const Glonass_Gnav_Ephemeris& ephGNAV, double obs_time,
const std::map<int32_t, Gnss_Synchro>& observables);
00505
00506     std::string get_IGM01_header(const Galileo_HAS_data& has_data, uint8_t nsys, bool
ssr_multiple_msg_indicator);
00507     std::string get_IGM01_content_sat(const Galileo_HAS_data& has_data, uint8_t nsys_index);
00508     std::string get_IGM02_header(const Galileo_HAS_data& has_data, uint8_t nsys, bool
ssr_multiple_msg_indicator);
00509     std::string get_IGM02_content_sat(const Galileo_HAS_data& has_data, uint8_t nsys_index);
00510     std::string get_IGM03_header(const Galileo_HAS_data& has_data, uint8_t nsys, bool
ssr_multiple_msg_indicator);
00511     std::string get_IGM03_content_sat(const Galileo_HAS_data& has_data, uint8_t nsys_index);
00512     std::string get_IGM05_header(const Galileo_HAS_data& has_data, uint8_t nsys, bool
ssr_multiple_msg_indicator);
00513     std::string get_IGM05_content_sat(const Galileo_HAS_data& has_data, uint8_t nsys_index);
00514
00515     //
00516     // Utilities
00517     //
00518     static std::map<std::string, int> galileo_signal_map;
00519     static std::map<std::string, int> gps_signal_map;
00520     std::vector<std::pair<int32_t, Gnss_Synchro>> sort_by_signal(const std::vector<std::pair<int32_t,
Gnss_Synchro>& synchro_map) const;
00521     std::vector<std::pair<int32_t, Gnss_Synchro>> sort_by_PRN_mask(const std::vector<std::pair<int32_t,
Gnss_Synchro>& synchro_map) const;
00522     boost::posix_time::ptime compute_GPS_time(const Gps_Ephemeris& eph, double obs_time) const;
00523     boost::posix_time::ptime compute_GPS_time(const Gps_CNAV_Ephemeris& eph, double obs_time) const;
00524     boost::posix_time::ptime compute_Galileo_time(const Galileo_Ephemeris& eph, double obs_time)
const;
00525     boost::posix_time::ptime compute_GLONASS_time(const Glonass_Gnav_Ephemeris& eph, double obs_time)
const;
00526     boost::posix_time::ptime gps_L1_last_lock_time[64];
00527     boost::posix_time::ptime gps_L2_last_lock_time[64];
00528     boost::posix_time::ptime gal_E1_last_lock_time[64];
00529     boost::posix_time::ptime gal_E5_last_lock_time[64];
00530     boost::posix_time::ptime glo_L1_last_lock_time[64];
00531     boost::posix_time::ptime glo_L2_last_lock_time[64];
00532     uint32_t lock_time_indicator(uint32_t lock_time_period_s);
00533     uint32_t msm_lock_time_indicator(uint32_t lock_time_period_s);

```

```

00534     uint32_t msm_extended_lock_time_indicator(uint32_t lock_time_period_s);
00535     // SSR utilities
00536     uint8_t ssr_update_interval(uint16_t validity_seconds) const;
00537
00538     //
00539     // Classes for TCP communication
00540     //
00541     uint16_t RTCM_port;
00542     // uint16_t RTCM_Station_ID;
00543     class RtcM_Message
00544     {
00545     public:
00546         static const std::size_t header_length = 6;
00547         static const std::size_t max_body_length = 1029;
00548
00549         RtcM_Message()
00550             : body_length_(0)
00551         {
00552         }
00553
00554         const char* data() const
00555         {
00556             return data_.data();
00557         }
00558
00559         char* data()
00560         {
00561             return data_.data();
00562         }
00563
00564         inline std::size_t length() const
00565         {
00566             return header_length + body_length_;
00567         }
00568
00569         const char* body() const
00570         {
00571             return data_.data() + header_length;
00572         }
00573
00574         char* body()
00575         {
00576             return data_.data() + header_length;
00577         }
00578
00579         std::size_t body_length() const
00580         {
00581             return body_length_;
00582         }
00583
00584         void body_length(std::size_t new_length)
00585         {
00586             body_length_ = new_length;
00587             if (body_length_ > max_body_length)
00588             {
00589                 body_length_ = max_body_length;
00590             }
00591         }
00592
00593         inline bool decode_header()
00594         {
00595             std::string header(data_.data(), header_length);
00596             if (header[0] != 'G' || header[1] != 'S')
00597             {
00598                 return false;
00599             }
00600
00601             auto header2 = header.substr(2);
00602             try
00603             {
00604                 body_length_ = std::stoi(header2);
00605             }
00606             catch (const std::exception& e)
00607             {
00608                 // invalid stoi conversion
00609                 body_length_ = 0;
00610                 return false;
00611             }
00612
00613             if (body_length_ == 0)
00614             {
00615                 return false;
00616             }
00617
00618             if (body_length_ > max_body_length)
00619             {
00620                 body_length_ = 0;

```

```

00621         return false;
00622     }
00623     return true;
00624 }
00625
00626 inline void encode_header()
00627 {
00628     std::stringstream ss;
00629     ss << "GS" << std::setw(4) << std::max(std::min(static_cast<int>(body_length_),
static_cast<int>(max_body_length)), 0);
00630     std::string header = ss.str();
00631     header.resize(header_length, ' ');
00632     std::copy(header.begin(), header.end(), data_.begin());
00633 }
00634
00635 private:
00636     std::array<char, header_length + max_body_length> data_{};
00637     std::size_t body_length_;
00638 };
00639
00640
00641 class Rtcmlistener
00642 {
00643 public:
00644     virtual ~Rtcmlistener() = default;
00645     virtual void deliver(const Rtcmlistener& msg) = 0;
00646 };
00647
00648
00649 class Rtcmlistener_Room
00650 {
00651 public:
00652     inline void join(const std::shared_ptr<Rtcmlistener>& participant)
00653     {
00654         participants_.insert(participant);
00655         for (const auto& msg : recent_msgs_)
00656             participant->deliver(msg);
00657     }
00658
00659     inline void leave(const std::shared_ptr<Rtcmlistener>& participant)
00660     {
00661         participants_.erase(participant);
00662     }
00663
00664     inline void deliver(const Rtcmlistener& msg)
00665     {
00666         recent_msgs_.push_back(msg);
00667         while (recent_msgs_.size() > max_recent_msgs)
00668             recent_msgs_.pop_front();
00669
00670         for (const auto& participant : participants_)
00671             participant->deliver(msg);
00672     }
00673
00674 private:
00675     std::set<std::shared_ptr<Rtcmlistener>> participants_;
00676     enum
00677     {
00678         max_recent_msgs = 1
00679     };
00680     std::deque<Rtcmlistener> recent_msgs_;
00681 };
00682
00683
00684 class Rtcmlistener_Session
00685 : public Rtcmlistener,
00686   public std::enable_shared_from_this<Rtcmlistener_Session>
00687 {
00688 public:
00689     Rtcmlistener_Session(boost::asio::ip::tcp::socket socket, Rtcmlistener_Room& room) :
00690         socket_(std::move(socket)), room_(room) {}
00691     inline void start()
00692     {
00693         room_.join(shared_from_this());
00694         do_read_message_header();
00695     }
00696
00697     inline void deliver(const Rtcmlistener& msg)
00698     {
00699         bool write_in_progress = !write_msgs_.empty();
00700         write_msgs_.push_back(msg);
00701     }

```

```

00706         if (!write_in_progress)
00707         {
00708             do_write();
00709         }
00710     }
00711
00712 private:
00713     inline void do_read_message_header()
00714     {
00715         auto self(shared_from_this());
00716         boost::asio::async_read(socket_,
00717             boost::asio::buffer(read_msg_.data(), RtcM_Message::header_length),
00718             [this, self](boost::system::error_code ec, std::size_t /*length*/) {
00719                 if (!ec and read_msg_.decode_header())
00720                 {
00721                     do_read_message_body();
00722                 }
00723                 else if (!ec and !read_msg_.decode_header())
00724                 {
00725                     client_says += read_msg_.data();
00726                     bool first = true;
00727                     while (client_says.length() >= 80)
00728                     {
00729                         if (first == true)
00730                         {
00731                             DLOG(INFO) << "Client says:";
00732                             first = false;
00733                         }
00734                         DLOG(INFO) << client_says;
00735                         client_says = client_says.substr(80, client_says.length() - 80);
00736                     }
00737                     do_read_message_header();
00738                 }
00739                 else
00740                 {
00741                     std::cout << "Closing connection with RTCM client\n";
00742                     room_.leave(shared_from_this());
00743                 }
00744             });
00745     }
00746
00747     inline void do_read_message_body()
00748     {
00749         auto self(shared_from_this());
00750         boost::asio::async_read(socket_,
00751             boost::asio::buffer(read_msg_.body(), read_msg_.body_length()),
00752             [this, self](boost::system::error_code ec, std::size_t /*length*/) {
00753                 if (!ec)
00754                 {
00755                     room_.deliver(read_msg_);
00756                     // std::cout << "Delivered message (session): ";
00757                     // std::cout.write(read_msg_.body(), read_msg_.body_length());
00758                     // std::cout << '\n';
00759                     do_read_message_header();
00760                 }
00761                 else
00762                 {
00763                     std::cout << "Closing connection with RTCM client\n";
00764                     room_.leave(shared_from_this());
00765                 }
00766             });
00767     }
00768
00769     inline void do_write()
00770     {
00771         auto self(shared_from_this());
00772         boost::asio::async_write(socket_,
00773             boost::asio::buffer(write_msgs_.front().body(), write_msgs_.front().body_length()),
00774             [this, self](boost::system::error_code ec, std::size_t /*length*/) {
00775                 if (!ec)
00776                 {
00777                     write_msgs_.pop_front();
00778                     if (!write_msgs_.empty())
00779                     {
00780                         do_write();
00781                     }
00782                 }
00783                 else
00784                 {
00785                     std::cout << "Closing connection with RTCM client\n";
00786                     room_.leave(shared_from_this());
00787                 }
00788             });
00789     }
00790
00791     boost::asio::ip::tcp::socket socket_;
00792     RtcM_Listener_Room& room_;

```

```

00793         Rtcmm_Message read_msg_;
00794         std::deque<Rtcmm_Message> write_msgs_;
00795         std::string client_says;
00796     };
00797
00798
00799     class Tcp_Internal_Client
00800     : public std::enable_shared_from_this<Tcp_Internal_Client>
00801     {
00802     public:
00803         Tcp_Internal_Client(b_io_context& io_context,
00804 #if BOOST_ASIO_USE_RESOLVER_ITERATOR
00805             boost::asio::ip::tcp::resolver::iterator endpoint_iterator)
00806             : io_context_(io_context), socket_(io_context)
00807             {
00808                 do_connect(std::move(endpoint_iterator));
00809             }
00810 #else
00811             boost::asio::ip::tcp::resolver::results_type endpoints)
00812             : io_context_(io_context), socket_(io_context)
00813             {
00814                 do_connect(std::move(endpoints));
00815             }
00816 #endif
00817
00818         inline void close()
00819         {
00820 #if BOOST_ASIO_USE_IOCONTEXT_POST
00821             io_context_.post([this]() { socket_.close(); });
00822 #else
00823             boost::asio::post(io_context_, [this]() { socket_.close(); });
00824 #endif
00825         }
00826
00827         inline void write(const Rtcmm_Message& msg)
00828         {
00829 #if BOOST_ASIO_USE_IOCONTEXT_POST
00830             io_context_.post(
00831                 [this, &msg]() {
00832 #else
00833             boost::asio::post(io_context_,
00834                 [this, &msg]() {
00835 #endif
00836                 bool write_in_progress = !write_msgs_.empty();
00837                 write_msgs_.push_back(msg);
00838                 if (!write_in_progress)
00839                 {
00840                     do_write();
00841                 }
00842             });
00843         }
00844
00845     private:
00846 #if BOOST_ASIO_USE_RESOLVER_ITERATOR
00847         inline void do_connect(boost::asio::ip::tcp::resolver::iterator endpoint_iterator)
00848         {
00849             boost::asio::async_connect(socket_, std::move(endpoint_iterator),
00850                 [this](boost::system::error_code ec, boost::asio::ip::tcp::resolver::iterator) {
00851 #else
00852         inline void do_connect(boost::asio::ip::tcp::resolver::results_type endpoints)
00853         {
00854             boost::asio::async_connect(socket_, std::move(endpoints),
00855                 [this](boost::system::error_code ec, boost::asio::ip::tcp::endpoint) {
00856 #endif
00857                 if (!ec)
00858                 {
00859                     do_read_message();
00860                 }
00861                 else
00862                 {
00863                     std::cout << "Server is down.\n";
00864                 }
00865             });
00866         }
00867
00868         inline void do_read_message()
00869         {
00870             boost::asio::async_read(socket_,
00871                 boost::asio::buffer(read_msg_.data(), 1029),
00872                 [this](boost::system::error_code ec, std::size_t /*length*/) {
00873                     if (!ec)
00874                     {
00875                         do_read_message();
00876                     }
00877                     else
00878                     {
00879                         std::cout << "Error in client\n";

```

```

00880         socket_.close();
00881     }
00882     });
00883 }
00884
00885 inline void do_write()
00886 {
00887     boost::asio::async_write(socket_,
00888         boost::asio::buffer(write_msgs_.front().data(), write_msgs_.front().length()),
00889         [this](boost::system::error_code ec, std::size_t /*length*/) {
00890             if (!ec)
00891             {
00892                 write_msgs_.pop_front();
00893                 if (!write_msgs_.empty())
00894                 {
00895                     do_write();
00896                 }
00897             }
00898             else
00899             {
00900                 socket_.close();
00901             }
00902         });
00903 }
00904
00905 b_io_context& io_context_;
00906 boost::asio::ip::tcp::socket socket_;
00907 RtcM_Message read_msg_;
00908 std::deque<RtcM_Message> write_msgs_;
00909 };
00910
00911
00912 class Queue_Reader
00913 {
00914 public:
00915     Queue_Reader(b_io_context& io_context, std::shared_ptr<Concurrent_Queue<std::string>& queue,
00916 int32_t port) : queue_(queue)
00917     {
00918         boost::asio::ip::tcp::resolver resolver(io_context);
00919         std::string host("localhost");
00920         std::string port_str = std::to_string(port);
00921 #if BOOST_ASIO_USE_RESOLVER_ITERATOR
00922         auto queue_endpoint_iterator = resolver.resolve({host.c_str(), port_str.c_str()});
00923         c = std::make_shared<Tcp_Internal_Client>(io_context, queue_endpoint_iterator);
00924 #else
00925         auto endpoints = resolver.resolve(host, port_str);
00926         c = std::make_shared<Tcp_Internal_Client>(io_context, endpoints);
00927 #endif
00928     }
00929
00930     inline void do_read_queue()
00931     {
00932         for (;;)
00933         {
00934             std::string message;
00935             RtcM_Message msg;
00936             queue_>wait_and_pop(message); // message += '\n';
00937             if (message == "Goodbye")
00938             {
00939                 break;
00940             }
00941
00942             const char* char_msg = message.c_str();
00943             msg.body_length(message.length());
00944             std::copy_n(char_msg, msg.body_length(), msg.body());
00945             msg.encode_header();
00946             c->write(msg);
00947         }
00948     }
00949
00950 private:
00951     std::shared_ptr<Tcp_Internal_Client> c;
00952     std::shared_ptr<Concurrent_Queue<std::string>& queue_;
00953 };
00954
00955 class Tcp_Server
00956 {
00957 public:
00958     Tcp_Server(b_io_context& io_context, const boost::asio::ip::tcp::endpoint& endpoint)
00959         : acceptor_(io_context), socket_(io_context)
00960     {
00961         acceptor_.open(endpoint.protocol());
00962         acceptor_.set_option(boost::asio::ip::tcp::acceptor::reuse_address(true));
00963         acceptor_.bind(endpoint);
00964         acceptor_.listen();
00965         do_accept();

```

```

00966     }
00967
00968     inline void close_server()
00969     {
00970         socket_.close();
00971         acceptor_.close();
00972     }
00973
00974     private:
00975     inline void do_accept()
00976     {
00977         acceptor_.async_accept(socket_, [this](boost::system::error_code ec) {
00978             if (!ec)
00979             {
00980                 if (first_client)
00981                 {
00982                     std::cout << "The TCP/IP server of RTCM messages is up and running.
Accepting connections ...\n";
00983                     first_client = false;
00984                 }
00985                 else
00986                 {
00987                     std::cout << "Starting RTCM TCP/IP server session...\n";
00988                     boost::system::error_code ec2;
00989                     boost::asio::ip::tcp::endpoint endpoint =
socket_.remote_endpoint(ec2);
00990                     if (ec2)
00991                     {
00992                         // Error creating remote_endpoint
00993                         std::cout << "Error getting remote IP address, closing
session.\n";
00994                         LOG(INFO) << "Error getting remote IP address";
00995                         start_session = false;
00996                     }
00997                     else
00998                     {
00999                         std::string remote_addr = endpoint.address().to_string();
01000                         std::cout << "Serving client from " << remote_addr << '\n';
01001                         LOG(INFO) << "Serving client from " << remote_addr;
01002                     }
01003                 }
01004                 if (start_session)
01005                 {
01006                     std::make_shared<Rtcm_Session>(std::move(socket_), room_)->start();
01007                 }
01008             }
01009             else
01010             {
01011                 std::cout << "Error when invoking a RTCM session. " << ec << '\n';
01012             }
01013             start_session = true;
01014             do_accept();
01015         });
01016     }
01017
01018     boost::asio::ip::tcp::acceptor acceptor_;
01019     boost::asio::ip::tcp::socket socket_;
01020     Rtcm_Listener_Room room_;
01021     bool first_client = true;
01022     bool start_session = true;
01023 };
01024
01025 b_io_context io_context;
01026 std::shared_ptr<Concurrent_Queue<std::string> rtcm_message_queue;
01027 std::thread t;
01028 std::thread tq;
01029 std::list<Rtcm::Tcp_Server> servers;
01030 bool server_is_running;
01031 void stop_service();
01032
01033 //
01034 // Transport Layer
01035 //
01036 std::bitset<8> preamble;
01037 std::bitset<6> reserved_field;
01038 std::string add_CRC(const std::string& m) const;
01039 std::string build_message(const std::string& data) const; // adds 0s to complete a byte and adds
the CRC
01040
01041 //
01042 // Data Fields
01043 //
01044 std::bitset<12> DF002;
01045 int32_t set_DF002(uint32_t message_number);
01046
01047 std::bitset<12> DF003;
01048 int32_t set_DF003(uint32_t ref_station_ID);

```

```

01049
01050     std::bitset<30> DF004;
01051     int32_t set_DF004(double obs_time);
01052
01053     std::bitset<1> DF005;
01054     int32_t set_DF005(bool sync_flag);
01055
01056     std::bitset<5> DF006;
01057     int32_t set_DF006(const std::map<int32_t, Gnss_Synchro>& observables);
01058
01059     std::bitset<1> DF007;
01060     int32_t set_DF007(bool divergence_free_smoothing_indicator); // 0 - Divergence-free smoothing not
used 1 - Divergence-free smoothing used
01061
01062     std::bitset<3> DF008;
01063     int32_t set_DF008(int16_t smoothing_interval);
01064
01065     std::bitset<6> DF009;
01066     int32_t set_DF009(const Gnss_Synchro& gnss_synchro);
01067     int32_t set_DF009(const Gps_Ephemeris& gps_eph);
01068
01069     std::bitset<1> DF010;
01070     int32_t set_DF010(bool code_indicator);
01071
01072     std::bitset<24> DF011;
01073     int32_t set_DF011(const Gnss_Synchro& gnss_synchro);
01074
01075     std::bitset<20> DF012;
01076     int32_t set_DF012(const Gnss_Synchro& gnss_synchro);
01077
01078     std::bitset<7> DF013;
01079     int32_t set_DF013(const Gps_Ephemeris& eph, double obs_time, const Gnss_Synchro& gnss_synchro);
01080
01081     std::bitset<8> DF014;
01082     int32_t set_DF014(const Gnss_Synchro& gnss_synchro);
01083
01084     std::bitset<8> DF015;
01085     int32_t set_DF015(const Gnss_Synchro& gnss_synchro);
01086
01087     std::bitset<14> DF017;
01088     int32_t set_DF017(const Gnss_Synchro& gnss_synchroL1, const Gnss_Synchro& gnss_synchroL2);
01089
01090     std::bitset<20> DF018;
01091     int32_t set_DF018(const Gnss_Synchro& gnss_synchroL1, const Gnss_Synchro& gnss_synchroL2);
01092
01093     std::bitset<7> DF019;
01094     int32_t set_DF019(const Gps_CNAV_Ephemeris& eph, double obs_time, const Gnss_Synchro&
gnss_synchro);
01095
01096     std::bitset<8> DF020;
01097     int32_t set_DF020(const Gnss_Synchro& gnss_synchro);
01098
01099     std::bitset<6> DF021;
01100     int32_t set_DF021();
01101
01102     std::bitset<1> DF022;
01103     int32_t set_DF022(bool gps_indicator);
01104
01105     std::bitset<1> DF023;
01106     int32_t set_DF023(bool glonass_indicator);
01107
01108     std::bitset<1> DF024;
01109     int32_t set_DF024(bool galileo_indicator);
01110
01111     std::bitset<38> DF025;
01112     int32_t set_DF025(double antenna_ECEF_X_m);
01113
01114     std::bitset<38> DF026;
01115     int32_t set_DF026(double antenna_ECEF_Y_m);
01116
01117     std::bitset<38> DF027;
01118     int32_t set_DF027(double antenna_ECEF_Z_m);
01119
01120     std::bitset<16> DF028;
01121     int32_t set_DF028(double height);
01122
01123     std::bitset<8> DF029;
01124
01125     std::bitset<8> DF031;
01126     int32_t set_DF031(uint32_t antenna_setup_id);
01127
01128     std::bitset<8> DF032;
01129
01130     /*!
01131     * \brief Sets the Data Field value
01132     * \note Code added as part of GSoC 2017 program
01133     * \param obs_time Time of observation at the moment of printing

```



```

01134     * \return returns 0 upon success
01135     */
01136     int32_t set_DF034(double obs_time);
01137     std::bitset<27> DF034;    //!< GLONASS Epoch Time (tk)
01138
01139     std::bitset<5> DF035;    //!< No. of GLONASS Satellite Signals Processed
01140     int32_t set_DF035(const std::map<int32_t, Gnss_Synchro>& observables);
01141
01142     std::bitset<1> DF036;    //!< GLONASS Divergence-free Smoothing Indicator
01143     int32_t set_DF036(bool divergence_free_smoothing_indicator);
01144
01145     std::bitset<3> DF037;    //!< GLONASS Smoothing Interval
01146     int32_t set_DF037(int16_t smoothing_interval);
01147
01148     std::bitset<6> DF038;    //!< GLONASS Satellite ID (Satellite Slot Number)
01149     int32_t set_DF038(const Gnss_Synchro& gnss_synchro);
01150     int32_t set_DF038(const Glonass_Gnav_Ephemeris& glonass_gnav_eph);
01151
01152     std::bitset<1> DF039;    //!< GLONASS L1 Code Indicator
01153     int32_t set_DF039(bool code_indicator);
01154
01155     std::bitset<5> DF040;    //!< GLONASS Satellite Frequency Number
01156     int32_t set_DF040(int32_t frequency_channel_number);
01157     int32_t set_DF040(const Glonass_Gnav_Ephemeris& glonass_gnav_eph);
01158
01159     std::bitset<25> DF041;    //!< GLONASS L1 Pseudorange
01160     int32_t set_DF041(const Gnss_Synchro& gnss_synchro);
01161
01162     std::bitset<20> DF042;    //!< GLONASS L1 PhaseRange - L1 Pseudorange
01163     int32_t set_DF042(const Gnss_Synchro& gnss_synchro);
01164
01165     std::bitset<7> DF043;    //!< GLONASS L1 Lock Time Indicator
01166     int32_t set_DF043(const Glonass_Gnav_Ephemeris& eph, double obs_time, const Gnss_Synchro&
gnss_synchro);
01167
01168     std::bitset<7> DF044;    //!< GLONASS Integer L1 Pseudorange Modulus Ambiguity
01169     int32_t set_DF044(const Gnss_Synchro& gnss_synchro);
01170
01171     std::bitset<8> DF045;    //!< GLONASS L1 CNR
01172     int32_t set_DF045(const Gnss_Synchro& gnss_synchro);
01173
01174     std::bitset<2> DF046;    //!< GLONASS L2 code indicator
01175     int32_t set_DF046(uint16_t code_indicator);
01176
01177     std::bitset<14> DF047;    //!< GLONASS L2 - L1 Pseudorange Difference
01178     int32_t set_DF047(const Gnss_Synchro& gnss_synchroL1, const Gnss_Synchro& gnss_synchroL2);
01179
01180     std::bitset<20> DF048;    //!< GLONASS L2 PhaseRange - L1 Pseudorange
01181     int32_t set_DF048(const Gnss_Synchro& gnss_synchroL1, const Gnss_Synchro& gnss_synchroL2);
01182
01183     std::bitset<7> DF049;    //!< GLONASS L2 Lock Time Indicator
01184     int32_t set_DF049(const Glonass_Gnav_Ephemeris& eph, double obs_time, const Gnss_Synchro&
gnss_synchro);
01185
01186     std::bitset<8> DF050;    //!< GLONASS L2 CNR
01187     int32_t set_DF050(const Gnss_Synchro& gnss_synchro);
01188
01189     std::bitset<16> DF051;
01190     int32_t set_DF051(const Gps_Ephemeris& gps_eph, double obs_time);
01191
01192     std::bitset<17> DF052;
01193     int32_t set_DF052(const Gps_Ephemeris& gps_eph, double obs_time);
01194
01195     // Contents of GPS Satellite Ephemeris Data, Message Type 1019
01196     std::bitset<8> DF071;
01197     int32_t set_DF071(const Gps_Ephemeris& gps_eph);
01198
01199     std::bitset<10> DF076;
01200     int32_t set_DF076(const Gps_Ephemeris& gps_eph);
01201
01202     std::bitset<4> DF077;
01203     int32_t set_DF077(const Gps_Ephemeris& gps_eph);
01204
01205     std::bitset<2> DF078;
01206     int32_t set_DF078(const Gps_Ephemeris& gps_eph);
01207
01208     std::bitset<14> DF079;
01209     int32_t set_DF079(const Gps_Ephemeris& gps_eph);
01210
01211     std::bitset<8> DF080;
01212     int32_t set_DF080(const Gps_Ephemeris& gps_eph);
01213
01214     std::bitset<16> DF081;
01215     int32_t set_DF081(const Gps_Ephemeris& gps_eph);
01216
01217     std::bitset<8> DF082;
01218     int32_t set_DF082(const Gps_Ephemeris& gps_eph);

```

```
01219
01220     std::bitset<16> DF083;
01221     int32_t set_DF083(const Gps_Ephemeris& gps_eph);
01222
01223     std::bitset<22> DF084;
01224     int32_t set_DF084(const Gps_Ephemeris& gps_eph);
01225
01226     std::bitset<10> DF085;
01227     int32_t set_DF085(const Gps_Ephemeris& gps_eph);
01228
01229     std::bitset<16> DF086;
01230     int32_t set_DF086(const Gps_Ephemeris& gps_eph);
01231
01232     std::bitset<16> DF087;
01233     int32_t set_DF087(const Gps_Ephemeris& gps_eph);
01234
01235     std::bitset<32> DF088;
01236     int32_t set_DF088(const Gps_Ephemeris& gps_eph);
01237
01238     std::bitset<16> DF089;
01239     int32_t set_DF089(const Gps_Ephemeris& gps_eph);
01240
01241     std::bitset<32> DF090;
01242     int32_t set_DF090(const Gps_Ephemeris& gps_eph);
01243
01244     std::bitset<16> DF091;
01245     int32_t set_DF091(const Gps_Ephemeris& gps_eph);
01246
01247     std::bitset<32> DF092;
01248     int32_t set_DF092(const Gps_Ephemeris& gps_eph);
01249
01250     std::bitset<16> DF093;
01251     int32_t set_DF093(const Gps_Ephemeris& gps_eph);
01252
01253     std::bitset<16> DF094;
01254     int32_t set_DF094(const Gps_Ephemeris& gps_eph);
01255
01256     std::bitset<32> DF095;
01257     int32_t set_DF095(const Gps_Ephemeris& gps_eph);
01258
01259     std::bitset<16> DF096;
01260     int32_t set_DF096(const Gps_Ephemeris& gps_eph);
01261
01262     std::bitset<32> DF097;
01263     int32_t set_DF097(const Gps_Ephemeris& gps_eph);
01264
01265     std::bitset<16> DF098;
01266     int32_t set_DF098(const Gps_Ephemeris& gps_eph);
01267
01268     std::bitset<32> DF099;
01269     int32_t set_DF099(const Gps_Ephemeris& gps_eph);
01270
01271     std::bitset<24> DF100;
01272     int32_t set_DF100(const Gps_Ephemeris& gps_eph);
01273
01274     std::bitset<8> DF101;
01275     int32_t set_DF101(const Gps_Ephemeris& gps_eph);
01276
01277     std::bitset<6> DF102;
01278     int32_t set_DF102(const Gps_Ephemeris& gps_eph);
01279
01280     std::bitset<1> DF103;
01281     int32_t set_DF103(const Gps_Ephemeris& gps_eph);
01282
01283     std::bitset<1> DF104; //!< GLONASS Almanac Health
01284     int32_t set_DF104(uint32_t glonass_gnav_alm_health);
01285
01286     std::bitset<1> DF105; //!< GLONASS Almanac Health Availability Indicator
01287     int32_t set_DF105(uint32_t glonass_gnav_alm_health_ind);
01288
01289     std::bitset<2> DF106; //!< GLONASS P1 Word
01290     int32_t set_DF106(const Glonass_Gnav_Ephemeris& glonass_gnav_eph);
01291
01292     std::bitset<12> DF107; //!< GLONASS Epoch (tk)
01293     int32_t set_DF107(const Glonass_Gnav_Ephemeris& glonass_gnav_eph);
01294
01295     std::bitset<1> DF108; //!< GLONASS MSB of Bn Word
01296     int32_t set_DF108(const Glonass_Gnav_Ephemeris& glonass_gnav_eph);
01297
01298     std::bitset<1> DF109; //!< GLONASS P2 Word
01299     int32_t set_DF109(const Glonass_Gnav_Ephemeris& glonass_gnav_eph);
01300
01301     std::bitset<7> DF110; //!< GLONASS Ephemeris Epoch (tb)
01302     int32_t set_DF110(const Glonass_Gnav_Ephemeris& glonass_gnav_eph);
01303
01304     std::bitset<24> DF111; //!< GLONASS Xn first derivative
01305     int32_t set_DF111(const Glonass_Gnav_Ephemeris& glonass_gnav_eph);
```

```
01306
01307     std::bitset<27> DF112;    //!< GLONASS Xn
01308     int32_t set_DF112(const Glonass_Gnav_Ephemeris& glonass_gnav_eph);
01309
01310     std::bitset<5> DF113;    //!< GLONASS Xn second derivative
01311     int32_t set_DF113(const Glonass_Gnav_Ephemeris& glonass_gnav_eph);
01312
01313     std::bitset<24> DF114;    //!< GLONASS Yn first derivative
01314     int32_t set_DF114(const Glonass_Gnav_Ephemeris& glonass_gnav_eph);
01315
01316     std::bitset<27> DF115;    //!< GLONASS Yn
01317     int32_t set_DF115(const Glonass_Gnav_Ephemeris& glonass_gnav_eph);
01318
01319     std::bitset<5> DF116;    //!< GLONASS Yn second derivative
01320     int32_t set_DF116(const Glonass_Gnav_Ephemeris& glonass_gnav_eph);
01321
01322     std::bitset<24> DF117;    //!< GLONASS Zn first derivative
01323     int32_t set_DF117(const Glonass_Gnav_Ephemeris& glonass_gnav_eph);
01324
01325     std::bitset<27> DF118;    //!< GLONASS Zn
01326     int32_t set_DF118(const Glonass_Gnav_Ephemeris& glonass_gnav_eph);
01327
01328     std::bitset<5> DF119;    //!< GLONASS Zn second derivative
01329     int32_t set_DF119(const Glonass_Gnav_Ephemeris& glonass_gnav_eph);
01330
01331     std::bitset<1> DF120;    //!< GLONASS P3
01332     int32_t set_DF120(const Glonass_Gnav_Ephemeris& glonass_gnav_eph);
01333
01334     std::bitset<11> DF121;    //!< GLONASS GAMMA_N
01335     int32_t set_DF121(const Glonass_Gnav_Ephemeris& glonass_gnav_eph);
01336
01337     std::bitset<2> DF122;    //!< GLONASS P
01338     int32_t set_DF122(const Glonass_Gnav_Ephemeris& glonass_gnav_eph);
01339
01340     std::bitset<1> DF123;    //!< GLONASS ln (third string)
01341     int32_t set_DF123(const Glonass_Gnav_Ephemeris& glonass_gnav_eph);
01342
01343     std::bitset<22> DF124;    //!< GLONASS TAU_N
01344     int32_t set_DF124(const Glonass_Gnav_Ephemeris& glonass_gnav_eph);
01345
01346     std::bitset<5> DF125;    //!< GLONASS DELTA_TAU_N
01347     int32_t set_DF125(const Glonass_Gnav_Ephemeris& glonass_gnav_eph);
01348
01349     std::bitset<5> DF126;    //!< GLONASS Eccentricity
01350     int32_t set_DF126(const Glonass_Gnav_Ephemeris& glonass_gnav_eph);
01351
01352     std::bitset<1> DF127;    //!< GLONASS P4
01353     int32_t set_DF127(const Glonass_Gnav_Ephemeris& glonass_gnav_eph);
01354
01355     std::bitset<4> DF128;    //!< GLONASS F_T
01356     int32_t set_DF128(const Glonass_Gnav_Ephemeris& glonass_gnav_eph);
01357
01358     std::bitset<11> DF129;    //!< GLONASS N_T
01359     int32_t set_DF129(const Glonass_Gnav_Ephemeris& glonass_gnav_eph);
01360
01361     std::bitset<2> DF130;    //!< GLONASS M
01362     int32_t set_DF130(const Glonass_Gnav_Ephemeris& glonass_gnav_eph);
01363
01364     std::bitset<1> DF131;    //!< GLONASS Availability of additional data
01365     int32_t set_DF131(uint32_t fifth_str_additional_data_ind);
01366
01367     std::bitset<11> DF132;    //!< GLONASS N_A
01368     int32_t set_DF132(const Glonass_Gnav_Utc_Model& glonass_gnav_utc_model);
01369
01370     std::bitset<32> DF133;    //!< GLONASS TAU_C
01371     int32_t set_DF133(const Glonass_Gnav_Utc_Model& glonass_gnav_utc_model);
01372
01373     std::bitset<5> DF134;    //!< GLONASS N_4
01374     int32_t set_DF134(const Glonass_Gnav_Utc_Model& glonass_gnav_utc_model);
01375
01376     std::bitset<22> DF135;    //!< GLONASS TAU_GPS
01377     int32_t set_DF135(const Glonass_Gnav_Utc_Model& glonass_gnav_utc_model);
01378
01379     std::bitset<1> DF136;    //!< GLONASS L_N (FIFTH STRING)
01380     int32_t set_DF136(const Glonass_Gnav_Ephemeris& glonass_gnav_eph);
01381
01382     std::bitset<1> DF137;
01383     int32_t set_DF137(const Gps_Ephemeris& gps_eph);
01384
01385
01386     std::bitset<1> DF141;
01387     int32_t set_DF141(const Gps_Ephemeris& gps_eph);
01388
01389     std::bitset<1> DF142;
01390     int32_t set_DF142(const Gps_Ephemeris& gps_eph);
01391
01392     std::bitset<30> DF248;
```

```
01393     int32_t set_DF248(double obs_time);
01394
01395     // Contents of Galileo F/NAV Satellite Ephemeris Data, Message Type 1045
01396     std::bitset<6> DF252;
01397     int32_t set_DF252(const Galileo_Ephemeris& gal_eph);
01398
01399     std::bitset<12> DF289;
01400     int32_t set_DF289(const Galileo_Ephemeris& gal_eph);
01401
01402     std::bitset<10> DF290;
01403     int32_t set_DF290(const Galileo_Ephemeris& gal_eph);
01404
01405     std::bitset<8> DF291;
01406     int32_t set_DF291(const Galileo_Ephemeris& gal_eph);
01407
01408     std::bitset<14> DF292;
01409     int32_t set_DF292(const Galileo_Ephemeris& gal_eph);
01410
01411     std::bitset<14> DF293;
01412     int32_t set_DF293(const Galileo_Ephemeris& gal_eph);
01413
01414     std::bitset<6> DF294;
01415     int32_t set_DF294(const Galileo_Ephemeris& gal_eph);
01416
01417     std::bitset<21> DF295;
01418     int32_t set_DF295(const Galileo_Ephemeris& gal_eph);
01419
01420     std::bitset<31> DF296;
01421     int32_t set_DF296(const Galileo_Ephemeris& gal_eph);
01422
01423     std::bitset<16> DF297;
01424     int32_t set_DF297(const Galileo_Ephemeris& gal_eph);
01425
01426     std::bitset<16> DF298;
01427     int32_t set_DF298(const Galileo_Ephemeris& gal_eph);
01428
01429     std::bitset<32> DF299;
01430     int32_t set_DF299(const Galileo_Ephemeris& gal_eph);
01431
01432     std::bitset<16> DF300;
01433     int32_t set_DF300(const Galileo_Ephemeris& gal_eph);
01434
01435     std::bitset<32> DF301;
01436     int32_t set_DF301(const Galileo_Ephemeris& gal_eph);
01437
01438     std::bitset<16> DF302;
01439     int32_t set_DF302(const Galileo_Ephemeris& gal_eph);
01440
01441     std::bitset<32> DF303;
01442     int32_t set_DF303(const Galileo_Ephemeris& gal_eph);
01443
01444     std::bitset<14> DF304;
01445     int32_t set_DF304(const Galileo_Ephemeris& gal_eph);
01446
01447     std::bitset<16> DF305;
01448     int32_t set_DF305(const Galileo_Ephemeris& gal_eph);
01449
01450     std::bitset<32> DF306;
01451     int32_t set_DF306(const Galileo_Ephemeris& gal_eph);
01452
01453     std::bitset<16> DF307;
01454     int32_t set_DF307(const Galileo_Ephemeris& gal_eph);
01455
01456     std::bitset<32> DF308;
01457     int32_t set_DF308(const Galileo_Ephemeris& gal_eph);
01458
01459     std::bitset<16> DF309;
01460     int32_t set_DF309(const Galileo_Ephemeris& gal_eph);
01461
01462     std::bitset<32> DF310;
01463     int32_t set_DF310(const Galileo_Ephemeris& gal_eph);
01464
01465     std::bitset<24> DF311;
01466     int32_t set_DF311(const Galileo_Ephemeris& gal_eph);
01467
01468     std::bitset<10> DF312;
01469     int32_t set_DF312(const Galileo_Ephemeris& gal_eph);
01470
01471     std::bitset<10> DF313;
01472     int32_t set_DF313(const Galileo_Ephemeris& gal_eph);
01473
01474     std::bitset<2> DF314;
01475     int32_t set_DF314(const Galileo_Ephemeris& gal_eph);
01476
01477     std::bitset<1> DF315;
01478     int32_t set_DF315(const Galileo_Ephemeris& gal_eph);
01479
```

```

01480     std::bitset<2> DF364;
01481
01482     // Content of message header for MSM1, MSM2, MSM3, MSM4, MSM5, MSM6 and MSM7
01483     std::bitset<1> DF393;
01484     int32_t set_DF393(bool more_messages); // 1 indicates that more MSMs follow for given physical
time and reference station ID
01485
01486     std::bitset<64> DF394;
01487     int32_t set_DF394(const std::map<int32_t, Gnss_Synchro>& gnss_synchro);
01488
01489     std::bitset<32> DF395;
01490     int32_t set_DF395(const std::map<int32_t, Gnss_Synchro>& gnss_synchro);
01491
01492     std::string set_DF396(const std::map<int32_t, Gnss_Synchro>& observables);
01493
01494     std::bitset<8> DF397;
01495     int32_t set_DF397(const Gnss_Synchro& gnss_synchro);
01496
01497     std::bitset<10> DF398;
01498     int32_t set_DF398(const Gnss_Synchro& gnss_synchro);
01499
01500     std::bitset<14> DF399;
01501     int32_t set_DF399(const Gnss_Synchro& gnss_synchro);
01502
01503     std::bitset<15> DF400;
01504     int32_t set_DF400(const Gnss_Synchro& gnss_synchro);
01505
01506     std::bitset<22> DF401;
01507     int32_t set_DF401(const Gnss_Synchro& gnss_synchro);
01508
01509     std::bitset<4> DF402;
01510     int32_t set_DF402(const Gps_Ephemeris& ephNAV, const Gps_CNAV_Ephemeris& ephCNAV, const
Galileo_Ephemeris& ephFNAV, const Glonass_Gnav_Ephemeris& ephGNAV, double obs_time, const
Gnss_Synchro& gnss_synchro);
01511
01512     std::bitset<6> DF403;
01513     int32_t set_DF403(const Gnss_Synchro& gnss_synchro);
01514
01515     std::bitset<15> DF404;
01516     int32_t set_DF404(const Gnss_Synchro& gnss_synchro);
01517
01518     std::bitset<20> DF405;
01519     int32_t set_DF405(const Gnss_Synchro& gnss_synchro);
01520
01521     std::bitset<24> DF406;
01522     int32_t set_DF406(const Gnss_Synchro& gnss_synchro);
01523
01524     std::bitset<10> DF407;
01525     int32_t set_DF407(const Gps_Ephemeris& ephNAV, const Gps_CNAV_Ephemeris& ephCNAV, const
Galileo_Ephemeris& ephFNAV, const Glonass_Gnav_Ephemeris& ephGNAV, double obs_time, const
Gnss_Synchro& gnss_synchro);
01526
01527     std::bitset<10> DF408;
01528     int32_t set_DF408(const Gnss_Synchro& gnss_synchro);
01529
01530     std::bitset<3> DF409;
01531     int32_t set_DF409(uint32_t iods);
01532
01533     std::bitset<2> DF411;
01534     int32_t set_DF411(uint32_t clock_steering_indicator);
01535
01536     std::bitset<2> DF412;
01537     int32_t set_DF412(uint32_t external_clock_indicator);
01538
01539     std::bitset<1> DF417;
01540     int32_t set_DF417(bool using_divergence_free_smoothing);
01541
01542     std::bitset<3> DF418;
01543     int32_t set_DF418(int32_t carrier_smoothing_interval_s);
01544
01545     std::bitset<1> DF420;
01546     int32_t set_DF420(const Gnss_Synchro& gnss_synchro);
01547
01548     // IGS State Space Representation (SSR) data fields
01549     // see https://files.igs.org/pub/data/format/igs\_ssr\_v1.pdf
01550     std::bitset<3> IDF001;
01551     void set_IDF001(uint8_t version);
01552
01553     std::bitset<8> IDF002;
01554     void set_IDF002(uint8_t igs_message_number);
01555
01556     std::bitset<20> IDF003;
01557     void set_IDF003(uint32_t tow);
01558
01559     std::bitset<4> IDF004;
01560     void set_IDF004(uint8_t ssr_update_interval);
01561

```

```

01562     std::bitset<1> IDF005;
01563     void set_IDF005(bool ssr_multiple_message_indicator);
01564
01565     std::bitset<1> IDF006;
01566     void set_IDF006(bool regional_indicator);
01567
01568     std::bitset<4> IDF007;
01569     void set_IDF007(uint8_t ssr_iod);
01570
01571     std::bitset<16> IDF008;
01572     void set_IDF008(uint16_t ssr_provider_id);
01573
01574     std::bitset<4> IDF009;
01575     void set_IDF009(uint8_t ssr_solution_id);
01576
01577     std::bitset<6> IDF010;
01578     void set_IDF010(uint8_t num_satellites);
01579
01580     std::bitset<6> IDF011;
01581     void set_IDF011(uint8_t gnss_satellite_id);
01582
01583     std::bitset<8> IDF012;
01584     void set_IDF012(uint8_t gnss_iod);
01585
01586     std::bitset<22> IDF013;
01587     void set_IDF013(float delta_orbit_radial_m);
01588
01589     std::bitset<20> IDF014;
01590     void set_IDF014(float delta_orbit_in_track_m);
01591
01592     std::bitset<20> IDF015;
01593     void set_IDF015(float delta_orbit_cross_track_m);
01594
01595     std::bitset<21> IDF016;
01596     void set_IDF016(float dot_orbit_delta_track_m_s);
01597
01598     std::bitset<19> IDF017;
01599     void set_IDF017(float dot_orbit_delta_in_track_m_s);
01600
01601     std::bitset<19> IDF018;
01602     void set_IDF018(float dot_orbit_delta_cross_track_m_s);
01603
01604     std::bitset<22> IDF019;
01605     void set_IDF019(float delta_clock_c0_m);
01606
01607     std::bitset<21> IDF020;
01608     void set_IDF020(float delta_clock_c1_m_s);
01609
01610     std::bitset<27> IDF021;
01611     void set_IDF021(float delta_clock_c2_m_s2);
01612
01613     std::bitset<5> IDF023;
01614     void set_IDF023(uint8_t num_bias_processed);
01615
01616     std::bitset<5> IDF024;
01617     void set_IDF024(uint8_t gnss_signal_tracking_mode_id);
01618
01619     std::bitset<14> IDF025;
01620     void set_IDF025(float code_bias_m);
01621 };
01622
01623
01624 /** \} */
01625 /** \} */
01626 #endif // GNSS_SDR_RTCM_H

```

## 11.276 rtcm\_printer.h File Reference

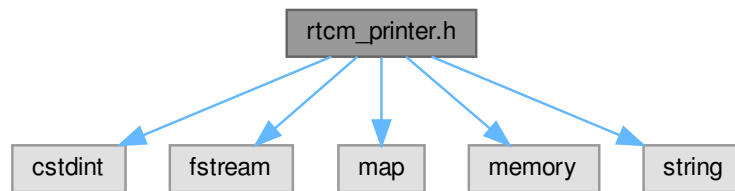
Interface of a RTCM 3.2 printer for GNSS-SDR This class provides a implementation of a subset of the RTCM Standard 10403.2 for Differential GNSS Services.

```

#include <stdint>
#include <fstream>
#include <map>
#include <memory>
#include <string>

```

Include dependency graph for rtcn\_printer.h:



## Classes

- class [Rtcn\\_Printer](#)

*This class provides a implementation of a subset of the RTCM Standard 10403.2 messages.*

## 11.276.1 Detailed Description

Interface of a RTCM 3.2 printer for GNSS-SDR This class provides a implementation of a subset of the RTCM Standard 10403.2 for Differential GNSS Services.

### Author

Carles Fernandez-Prades, 2014. cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [rtcn\\_printer.h](#).

---

## 11.277 rtcn\_printer.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file rtcn_printer.h
00003  * \brief Interface of a RTCM 3.2 printer for GNSS-SDR
00004  * This class provides a implementation of a subset of the RTCM Standard 10403.2
00005  * for Differential GNSS Services
00006  *
00007  * \author Carles Fernandez-Prades, 2014. cfernandez(at)cttc.es
00008  *
00009  * -----
00010  *
00011  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00012  * This file is part of GNSS-SDR.
00013  *
00014  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00015  * SPDX-License-Identifier: GPL-3.0-or-later
00016  *
00017  * -----
00018  */
00019
00020 #ifndef GNSS_SDR_RTCN_PRINTER_H
00021 #define GNSS_SDR_RTCN_PRINTER_H
00022
00023 #include <cstdint> // for int32_t
00024 #include <fstream> // for std::ofstream
00025 #include <map> // for std::map
00026 #include <memory> // for std::shared_ptr
00027 #include <string>
00028
00029 /** \addtogroup PVT
00030  * \{ */
00031 /** \addtogroup PVT_libs
00032  * \{ */
00033
  
```

```

00034
00035 class Galileo_Ephemeris;
00036 class Glonass_Gnav_Ephemeris;
00037 class Glonass_Gnav_Utc_Model;
00038 class Gnss_Synchro;
00039 class Gps_CNAV_Ephemeris;
00040 class Gps_Ephemeris;
00041 class RtcM;
00042 class Rtklib_Solver;
00043 class Galileo_HAS_data;
00044
00045 /*!
00046  * \brief This class provides a implementation of a subset of the RTCM Standard 10403.2 messages
00047  */
00048 class RtcM_Printer
00049 {
00050 public:
00051     /*!
00052      * \brief Default constructor.
00053      */
00054     RtcM_Printer(const std::string& filename,
00055                 bool flag_rtcM_file_dump,
00056                 bool flag_rtcM_server,
00057                 bool flag_rtcM_tty_port,
00058                 uint16_t rtcM_tcp_port,
00059                 uint16_t rtcM_station_id,
00060                 const std::string& rtcM_dump_devname,
00061                 bool time_tag_name = true,
00062                 const std::string& base_path = ".");
00063
00064     /*!
00065      * \brief Default destructor.
00066      */
00067     ~RtcM_Printer();
00068
00069     /*!
00070      * \brief Print RTCM messages.
00071      */
00072     void Print_RtcM_Messages(const Rtklib_Solver* pvt_solver,
00073                             const std::map<int, Gnss_Synchro>& gnss_observables_map,
00074                             double rx_time,
00075                             int32_t type_of_rx,
00076                             int32_t rtcM_MSM_rate_ms,
00077                             int32_t rtcM_MT1019_rate_ms,
00078                             int32_t rtcM_MT1020_rate_ms,
00079                             int32_t rtcM_MT1045_rate_ms,
00080                             int32_t rtcM_MT1077_rate_ms,
00081                             int32_t rtcM_MT1097_rate_ms,
00082                             bool flag_write_RTCM_MSM_output,
00083                             bool flag_write_RTCM_1019_output,
00084                             bool flag_write_RTCM_1020_output,
00085                             bool flag_write_RTCM_1045_output,
00086                             bool enable_rx_clock_correction);
00087
00088     uint32_t lock_time(const Gps_Ephemeris& eph, double obs_time, const Gnss_Synchro& gnss_synchro);
00089     uint32_t lock_time(const Gps_CNAV_Ephemeris& eph, double obs_time, const Gnss_Synchro& gnss_synchro);
00090     uint32_t lock_time(const Galileo_Ephemeris& eph, double obs_time, const Gnss_Synchro& gnss_synchro);
00091
00092     /*!
00093      * \brief Locks time for logging given GLONASS GNAV Broadcast Ephemeris
00094      * \note Code added as part of GSoC 2017 program
00095      * \params glonass_gnav_eph GLONASS GNAV Broadcast Ephemeris
00096      * \params obs_time Time of observation at the moment of printing
00097      * \params observables Set of observables as defined by the platform
00098      * \return locked time during logging process
00099      */
00100     uint32_t lock_time(const Glonass_Gnav_Ephemeris& eph, double obs_time, const Gnss_Synchro& gnss_synchro);
00101
00102     void Print_IGM_Messages(const Galileo_HAS_data& has_data);
00103
00104     std::string print_MT1005_test(); //!< For testing purposes
00105
00106 private:
00107     bool Print_RtcM_MT1001(const Gps_Ephemeris& gps_eph, double obs_time, const std::map<int32_t, Gnss_Synchro>& observables);
00108     bool Print_RtcM_MT1002(const Gps_Ephemeris& gps_eph, double obs_time, const std::map<int32_t, Gnss_Synchro>& observables);
00109     bool Print_RtcM_MT1003(const Gps_Ephemeris& gps_eph, const Gps_CNAV_Ephemeris& cnav_eph, double obs_time, const std::map<int32_t, Gnss_Synchro>& observables);
00110     bool Print_RtcM_MT1004(const Gps_Ephemeris& gps_eph, const Gps_CNAV_Ephemeris& cnav_eph, double obs_time, const std::map<int32_t, Gnss_Synchro>& observables);
00111
00112     /*
00113      * \brief Prints L1-Only GLONASS RTK Observables

```



```

00114     * \details This GLONASS message type is not generally used or supported; type 1012 is to be
    preferred.
00115     * \note Code added as part of GSoC 2017 program
00116     * \param glonass_gnav_eph GLONASS GNAV Broadcast Ephemeris
00117     * \param obs_time Time of observation at the moment of printing
00118     * \param observables Set of observables as defined by the platform
00119     * \return true or false upon operation success
00120     */
00121     bool Print_Rtcm_MT1009(const Glonass_Gnav_Ephemeris& glonass_gnav_eph, double obs_time, const
std::map<int32_t, Gnss_Synchro>& observables);
00122
00123     /*
00124     * \brief Prints Extended L1-Only GLONASS RTK Observables
00125     * \details This GLONASS message type is used when only L1 data is present and bandwidth is very
    tight, often 1012 is used in such cases.
00126     * \note Code added as part of GSoC 2017 program
00127     * \param glonass_gnav_eph GLONASS GNAV Broadcast Ephemeris
00128     * \param obs_time Time of observation at the moment of printing
00129     * \param observables Set of observables as defined by the platform
00130     * \return true or false upon operation success
00131     */
00132     bool Print_Rtcm_MT1010(const Glonass_Gnav_Ephemeris& glonass_gnav_eph, double obs_time, const
std::map<int32_t, Gnss_Synchro>& observables);
00133
00134     /*
00135     * \brief Prints L1&L2 GLONASS RTK Observables
00136     * \details This GLONASS message type is not generally used or supported; type 1012 is to be
    preferred
00137     * \note Code added as part of GSoC 2017 program
00138     * \param glonass_gnav_ephL1 GLONASS L1 GNAV Broadcast Ephemeris for satellite
00139     * \param glonass_gnav_ephL2 GLONASS L2 GNAV Broadcast Ephemeris for satellite
00140     * \param obs_time Time of observation at the moment of printing
00141     * \param observables Set of observables as defined by the platform
00142     * \return true or false upon operation success
00143     */
00144     bool Print_Rtcm_MT1011(const Glonass_Gnav_Ephemeris& glonass_gnav_ephL1, const
Glonass_Gnav_Ephemeris& glonass_gnav_ephL2, double obs_time, const std::map<int32_t, Gnss_Synchro>&
observables);
00145
00146     /*
00147     * \brief Prints Extended L1&L2 GLONASS RTK Observables
00148     * \details This GLONASS message type is the most common observational message type, with
    L1/L2/SNR content. This is one of the most common messages found.
00149     * \note Code added as part of GSoC 2017 program
00150     * \param glonass_gnav_ephL1 GLONASS L1 GNAV Broadcast Ephemeris for satellite
00151     * \param glonass_gnav_ephL2 GLONASS L2 GNAV Broadcast Ephemeris for satellite
00152     * \param obs_time Time of observation at the moment of printing
00153     * \param observables Set of observables as defined by the platform
00154     * \return true or false upon operation success
00155     */
00156     bool Print_Rtcm_MT1012(const Glonass_Gnav_Ephemeris& glonass_gnav_ephL1, const
Glonass_Gnav_Ephemeris& glonass_gnav_ephL2, double obs_time, const std::map<int32_t, Gnss_Synchro>&
observables);
00157
00158     bool Print_Rtcm_MT1019(const Gps_Ephemeris& gps_eph); // GPS Ephemeris, should be broadcast
    in the event that the IODC does not match the IODE, and every 2 minutes.
00159     bool Print_Rtcm_MT1045(const Galileo_Ephemeris& gal_eph); // Galileo Ephemeris, should be
    broadcast every 2 minutes
00160
00161     /*
00162     * \brief Prints GLONASS GNAV Ephemeris
00163     * \details This GLONASS message should be broadcast every 2 minutes
00164     * \note Code added as part of GSoC 2017 program
00165     * \param glonass_gnav_eph GLONASS GNAV Broadcast Ephemeris
00166     * \param utc_model GLONASS GNAV Clock Information broadcast in string 5
00167     * \return true or false upon operation success
00168     */
00169     bool Print_Rtcm_MT1020(const Glonass_Gnav_Ephemeris& glonass_gnav_eph, const
Glonass_Gnav_Utc_Model& utc_model);
00170
00171     bool Print_Rtcm_MSM(uint32_t msm_number,
00172         const Gps_Ephemeris& gps_eph,
00173         const Gps_CNAV_Ephemeris& gps_cnav_eph,
00174         const Galileo_Ephemeris& gal_eph,
00175         const Glonass_Gnav_Ephemeris& glo_gnav_eph,
00176         double obs_time,
00177         const std::map<int32_t, Gnss_Synchro>& observables,
00178         uint32_t clock_steering_indicator,
00179         uint32_t external_clock_indicator,
00180         int32_t smooth_int,
00181         bool divergence_free,
00182         bool more_messages);
00183
00184     bool Print_IGM01(const Galileo_HAS_data& has_data); // SSR Orbit Corrections
00185     bool Print_IGM02(const Galileo_HAS_data& has_data); // SSR Clock Corrections
00186     bool Print_IGM03(const Galileo_HAS_data& has_data); // SSR Combined Orbit & Clock Corrections
00187     bool Print_IGM05(const Galileo_HAS_data& has_data); // SSR Bias Corrections

```

```

00188
00189     int32_t init_serial(const std::string& serial_device); // serial port control
00190     void close_serial() const;
00191     bool Print_Message(const std::string& message);
00192
00193     std::unique_ptr<Rtcm> rtcm;
00194     std::ofstream rtcm_file_descriptor; // Output file stream for RTCM log file
00195     std::string rtcm_filename; // String with the RTCM log filename
00196     std::string rtcm_base_path;
00197     std::string rtcm_devname;
00198     int32_t rtcm_dev_descriptor; // RTCM serial device descriptor (i.e. COM port)
00199     uint16_t port;
00200     uint16_t station_id;
00201     bool d_rtcn_writing_started;
00202     bool d_rtcn_file_dump;
00203 };
00204
00205
00206 /** \} */
00207 /** \} */
00208 #endif // GNSS_SDR_RTCM_PRINTER_H

```

## 11.278 rtklib\_solver.h File Reference

PVT solver based on rtklib library functions adapted to the GNSS-SDR data flow and structures.

```

#include "beidou_dnav_almanac.h"
#include "beidou_dnav_ephemeris.h"
#include "beidou_dnav_iono.h"
#include "beidou_dnav_utc_model.h"
#include "galileo_almanac.h"
#include "galileo_ephemeris.h"
#include "galileo_has_data.h"
#include "galileo_iono.h"
#include "galileo_utc_model.h"
#include "glonass_gnav_almanac.h"
#include "glonass_gnav_ephemeris.h"
#include "glonass_gnav_utc_model.h"
#include "gnss_synchro.h"
#include "gps_almanac.h"
#include "gps_cnav_ephemeris.h"
#include "gps_cnav_iono.h"
#include "gps_cnav_utc_model.h"
#include "gps_ephemeris.h"
#include "gps_iono.h"
#include "gps_utc_model.h"
#include "monitor_pvt.h"
#include "pvt_conf.h"
#include "pvt_kf.h"
#include "pvt_solution.h"
#include "rtklib.h"
#include "rtklib_conversions.h"
#include <array>
#include <cstdint>
#include <fstream>
#include <map>
#include <string>
#include <utility>

```

Include dependency graph for rtklib\_solver.h:



**Classes**

- class [Rtklib\\_Solver](#)

*This class implements a PVT solution based on RTKLIB.*

**11.278.1 Detailed Description**

PVT solver based on rtklib library functions adapted to the GNSS-SDR data flow and structures.

**Authors**

- 2017, Javier Arribas
- 2017-2023, Carles Fernandez
- 2007-2013, T. Takasu

This is a derived work from RTKLIB <http://www.rtklib.com/> The original source code at <https://github.com/tomojitakasu/RTKLIB> is released under the BSD 2-clause license with an additional exclusive clause that does not apply here. This additional clause is reproduced below:

" The software package includes some companion executive binaries or shared libraries necessary to execute APs on Windows. These licenses succeed to the original ones of these software. "

Neither the executive binaries nor the shared libraries are required by, used or included in GNSS-SDR.

Copyright (C) 2007-2013, T. Takasu Copyright (C) 2017-2019, Javier Arribas Copyright (C) 2017-2023, Carles Fernandez All rights reserved.

SPDX-License-Identifier: BSD-2-Clause

Definition in file [rtklib\\_solver.h](#).

**11.279 rtklib\_solver.h**

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file rtklib_solver.h
00003  * \brief PVT solver based on rtklib library functions adapted to the GNSS-SDR
00004  * data flow and structures
00005  * \authors <ul>
00006  *         <li> 2017, Javier Arribas
00007  *         <li> 2017-2023, Carles Fernandez
00008  *         <li> 2007-2013, T. Takasu
00009  *     </ul>
00010  *
00011  * This is a derived work from RTKLIB http://www.rtklib.com/
00012  * The original source code at https://github.com/tomojitakasu/RTKLIB is
00013  * released under the BSD 2-clause license with an additional exclusive clause
00014  * that does not apply here. This additional clause is reproduced below:
00015  *
00016  * " The software package includes some companion executive binaries or shared
00017  * libraries necessary to execute APs on Windows. These licenses succeed to the
00018  * original ones of these software. "
00019  *
00020  * Neither the executive binaries nor the shared libraries are required by, used
00021  * or included in GNSS-SDR.
00022  *
00023  * -----
00024  * Copyright (C) 2007-2013, T. Takasu
00025  * Copyright (C) 2017-2019, Javier Arribas
00026  * Copyright (C) 2017-2023, Carles Fernandez
00027  * All rights reserved.
00028  *
00029  * SPDX-License-Identifier: BSD-2-Clause
00030  *
00031  * -----
00032  */
00033
00034 #ifndef GNSS_SDR_RTKLIB_SOLVER_H
00035 #define GNSS_SDR_RTKLIB_SOLVER_H
00036
00037
00038 #include "beidou_dnav_almanac.h"
00039 #include "beidou_dnav_ephemeris.h"
00040 #include "beidou_dnav_iono.h"
00041 #include "beidou_dnav_utc_model.h"
00042 #include "galileo_almanac.h"
00043 #include "galileo_ephemeris.h"
00044 #include "galileo_has_data.h"
00045 #include "galileo_iono.h"
```

```

00046 #include "galileo_utc_model.h"
00047 #include "glonass_gnav_almanac.h"
00048 #include "glonass_gnav_ephemeris.h"
00049 #include "glonass_gnav_utc_model.h"
00050 #include "gnss_synchro.h"
00051 #include "gps_almanac.h"
00052 #include "gps_cnav_ephemeris.h"
00053 #include "gps_cnav_iono.h"
00054 #include "gps_cnav_utc_model.h"
00055 #include "gps_ephemeris.h"
00056 #include "gps_iono.h"
00057 #include "gps_utc_model.h"
00058 #include "monitor_pvt.h"
00059 #include "pvt_conf.h"
00060 #include "pvt_kf.h"
00061 #include "pvt_solution.h"
00062 #include "rtklib.h"
00063 #include "rtklib_conversions.h"
00064 #include <array>
00065 #include <stdint>
00066 #include <fstream>
00067 #include <map>
00068 #include <string>
00069 #include <utility>
00070
00071 /** \addtogroup PVT
00072  * \{ */
00073 /** \addtogroup PVT_libs pvt_libs
00074  * Library for the computation of PVT solutions.
00075  * \{ */
00076
00077
00078 /*!
00079  * \brief This class implements a PVT solution based on RTKLIB
00080  */
00081 class Rtklib_Solver : public Pvt_Solution
00082 {
00083 public:
00084     Rtklib_Solver(const rtk_t& rtk,
00085                  const Pvt_Conf& conf,
00086                  const std::string& dump_filename,
00087                  uint32_t type_of_rx,
00088                  bool flag_dump_to_file,
00089                  bool flag_dump_to_mat);
00090
00091     ~Rtklib_Solver();
00092
00093     bool get_PVT(const std::map<int, Gnss_Synchro>& gnss_observables_map, double
kf_update_interval_s);
00094
00095     double get_hdop() const override;
00096     double get_vdop() const override;
00097     double get_pdop() const override;
00098     double get_gdop() const override;
00099     Monitor_Pvt get_monitor_pvt() const;
00100     void store_has_data(const Galileo_HAS_data& new_has_data);
00101     void update_has_corrections(const std::map<int, Gnss_Synchro>& obs_map);
00102
00103     sol_t pvt_sol{};
00104     std::array<ssat_t, MAXSAT> pvt_ssat{};
00105
00106     std::map<int, Galileo_Ephemeris> galileo_ephemeris_map;           //!< Map storing new
Galileo_Ephemeris
00107     std::map<int, Gps_Ephemeris> gps_ephemeris_map;                 //!< Map storing new
GPS_Ephemeris
00108     std::map<int, Gps_CNAV_Ephemeris> gps_cnav_ephemeris_map;       //!< Map storing new
GPS_CNAV_Ephemeris
00109     std::map<int, Glonass_Gnav_Ephemeris> glonass_gnav_ephemeris_map; //!< Map storing new GLONASS
GNAV Ephemeris
00110     std::map<int, Beidou_Dnav_Ephemeris> beidou_dnav_ephemeris_map; //!< Map storing new BeiDou
DNAV Ephemeris
00111
00112     Galileo_Utc_Model galileo_utc_model;
00113     Galileo_Iono galileo_iono;
00114     std::map<int, Galileo_Almanac> galileo_almanac_map;
00115
00116     Gps_Utc_Model gps_utc_model;
00117     Gps_Iono gps_iono;
00118     std::map<int, Gps_Almanac> gps_almanac_map;
00119
00120     Gps_CNAV_Iono gps_cnav_iono;
00121     Gps_CNAV_Utc_Model gps_cnav_utc_model;
00122
00123     Glonass_Gnav_Utc_Model glonass_gnav_utc_model; //!< Map storing GLONASS GNAV UTC Model
00124     Glonass_Gnav_Almanac glonass_gnav_almanac;    //!< Map storing GLONASS GNAV Almanac Model
00125
00126     Beidou_Dnav_Utc_Model beidou_dnav_utc_model;

```

```

00127     Beidou_Dnav_Iono beidou_dnav_iono;
00128     std::map<int, Beidou_Dnav_Almanac> beidou_dnav_almanac_map;
00129
00130 private:
00131     bool save_matfile() const;
00132
00133     void check_has_orbit_clock_validity(const std::map<int, Gnss_Synchro>& obs_map);
00134     void get_has_biases(const std::map<int, Gnss_Synchro>& obs_map);
00135     void get_current_has_obs_correction(const std::string& signal, uint32_t tow_obs, int prn);
00136
00137     std::array<obsd_t, MAXOBS> d_obs_data{};
00138     std::array<double, 4> d_dop{};
00139     std::map<int, int> d_rtklib_freq_index;
00140     std::map<std::string, int> d_rtklib_band_index;
00141
00142     std::map<std::string, std::map<int, HAS_orbit_corrections> d_has_orbit_corrections_store_map; //
00143     first key is system, second key is PRN
00144     std::map<std::string, std::map<int, HAS_clock_corrections> d_has_clock_corrections_store_map; //
00145     first key is system, second key is PRN
00146
00147     std::map<std::string, std::map<int, std::pair<float, uint32_t>> d_has_code_bias_store_map; //
00148     first key is signal, second key is PRN
00149     std::map<std::string, std::map<int, std::pair<float, uint32_t>> d_has_phase_bias_store_map; //
00150     first key is signal, second key is PRN
00151
00152     std::map<std::string, std::map<int, HAS_obs_corrections> d_has_obs_corr_map; // first key is
00153     signal, second key is PRN
00154
00155     std::string d_dump_filename;
00156     std::ofstream d_dump_file;
00157     rtk_t d_rtk{};
00158     nav_t d_nav_data{};
00159     Monitor_Pvt d_monitor_pvt{};
00160     Pvt_Conf d_conf;
00161     Pvt_Kf d_pvt_kf;
00162     uint32_t d_type_of_rx;
00163     bool d_flag_dump_enabled;
00164     bool d_flag_dump_mat_enabled;
00165 };
00166
00167 /** \} */
00168 /** \} */
00169 #endif // GNSS_SDR_RTKLIB_SOLVER_H

```

## 11.280 serdes\_galileo\_eph.h File Reference

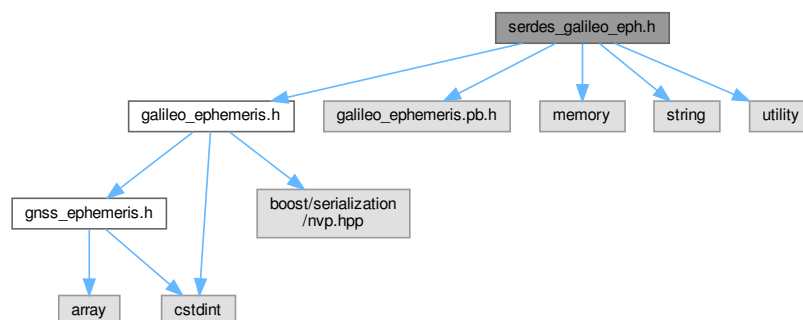
Serialization / Deserialization of [Galileo\\_Ephemeris](#) objects using Protocol Buffers.

```

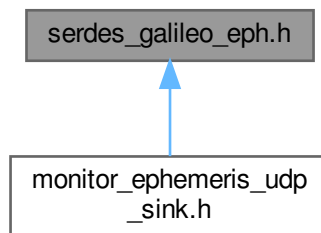
#include "galileo_ephemeris.h"
#include "galileo_ephemeris.pb.h"
#include <memory>
#include <string>
#include <utility>

```

Include dependency graph for serdes\_galileo\_eph.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Serdes\\_Galileo\\_Eph](#)

*This class implements serialization and deserialization of [Galileo\\_Ephemeris](#) using Protocol Buffers.*

### 11.280.1 Detailed Description

Serialization / Deserialization of [Galileo\\_Ephemeris](#) objects using Protocol Buffers.

#### Author

Javier Arribas, 2021. [jarribas\(at\)cttc.es](mailto:jarribas(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [serdes\\_galileo\\_eph.h](#).

---

## 11.281 serdes\_galileo\_eph.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file serdes_galileo_eph.h
00003  * \brief Serialization / Deserialization of Galileo_Ephemeris objects using
00004  * Protocol Buffers
00005  * \author Javier Arribas, 2021. jarribas\(at\)cttc.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_SERDES_GALILEO_EPH_H
00019 #define GNSS_SDR_SERDES_GALILEO_EPH_H
00020
00021 #include "galileo_ephemeris.h"
00022 #include "galileo_ephemeris.pb.h" // file created by Protocol Buffers at compile time
00023 #include <memory>
00024 #include <string>
00025 #include <utility>
00026
00027 /** \addtogroup PVT
00028  * \{ */
00029 /** \addtogroup PVT_libs
00030  * \{ */
00031
00032
  
```

```

00033  /*!
00034  * \brief This class implements serialization and deserialization of
00035  * Galileo_Ephemeris using Protocol Buffers.
00036  */
00037  class Serdes_Galileo_Eph
00038  {
00039  public:
00040      Serdes_Galileo_Eph()
00041      {
00042          // Verify that the version of the library that we linked against is
00043          // compatible with the version of the headers we compiled against.
00044          GOOGLE_PROTOBUF_VERIFY_VERSION;
00045      }
00046
00047      ~Serdes_Galileo_Eph()
00048      {
00049          // google::protobuf::ShutdownProtobufLibrary();
00050      }
00051
00052      inline Serdes_Galileo_Eph(const Serdes_Galileo_Eph& other) noexcept : monitor_(other.monitor_)
00053      { //!< Copy constructor
00054      }
00055
00056      inline Serdes_Galileo_Eph& operator=(const Serdes_Galileo_Eph& rhs) noexcept //!< Copy assignment
00057      {
00058          Serdes_Galileo_Eph temp(rhs);
00059          std::swap(this->monitor_, temp.monitor_);
00060          return *this;
00061      }
00062
00063      inline Serdes_Galileo_Eph(Serdes_Galileo_Eph&& other) noexcept :
00064      monitor_(std::move(other.monitor_)) //!< Move constructor
00065      {
00066      }
00067
00068      inline Serdes_Galileo_Eph& operator=(Serdes_Galileo_Eph&& other) noexcept //!< Move assignment
00069      {
00070          std::swap(this->monitor_, other.monitor_);
00071          return *this;
00072      }
00073
00074      inline std::string createProtobuffer(const std::shared_ptr<Galileo_Ephemeris> monitor) //!<
00075      { //!< Serialization into a string
00076          monitor_.Clear();
00077          std::string data;
00078
00079          monitor_.set_prn(monitor->PRN);
00080          monitor_.set_m_0(monitor->M_0);
00081          monitor_.set_delta_n(monitor->delta_n);
00082          monitor_.set_ecc(monitor->ecc);
00083          monitor_.set_sqrtA(monitor->sqrtA);
00084          monitor_.set_omega_0(monitor->OMEGA_0);
00085          monitor_.set_i_0(monitor->i_0);
00086          monitor_.set_omega(monitor->omega);
00087          monitor_.set_omegadot(monitor->OMEGAdot);
00088          monitor_.set_idot(monitor->idot);
00089          monitor_.set_cuc(monitor->Cuc);
00090          monitor_.set_cus(monitor->Cus);
00091          monitor_.set_crc(monitor->Crc);
00092          monitor_.set_crs(monitor->Crs);
00093          monitor_.set_cic(monitor->Cic);
00094          monitor_.set_cis(monitor->Cis);
00095          monitor_.set_toe(monitor->toe);
00096          monitor_.set_toc(monitor->toc);
00097          monitor_.set_af0(monitor->af0);
00098          monitor_.set_af1(monitor->af1);
00099          monitor_.set_af2(monitor->af2);
00100          monitor_.set_satclkdrift(monitor->satClkDrift);
00101          monitor_.set_dtr(monitor->dtr);
00102          monitor_.set_wn(monitor->WN);
00103          monitor_.set_tow(monitor->tow);
00104
00105          // Galileo-specific parameters
00106          monitor_.set_iod_ephemeris(monitor->IOD_ephemeris);
00107          monitor_.set_iod_nav(monitor->IOD_nav);
00108          monitor_.set_sisa(monitor->SISA);
00109          monitor_.set_e5a_hs(monitor->E5a_HS);
00110          monitor_.set_e5b_hs(monitor->E5b_HS);
00111          monitor_.set_e1b_hs(monitor->E1B_HS);
00112          monitor_.set_e5a_dvs(monitor->E5a_DVS);
00113          monitor_.set_e5b_dvs(monitor->E5b_DVS);
00114          monitor_.set_e1b_dvs(monitor->E1B_DVS);

```

```

00115         monitor_.set_bgd_ele5a(monitor->BGD_E1E5a);
00116         monitor_.set_bgd_ele5b(monitor->BGD_E1E5b);
00117
00118         monitor_.SerializeToString(&data);
00119         return data;
00120     }
00121
00122     inline Galileo_Ephemeris readProtobuffer(const gnss_sdr::GalileoEphemeris& mon) const //!<
Deserialization
00123     {
00124         Galileo_Ephemeris monitor;
00125
00126         monitor.PRN = mon.prn();
00127         monitor.M_0 = mon.m_0();
00128         monitor.delta_n = mon.delta_n();
00129         monitor.ecc = mon.ecc();
00130         monitor.sqrta = mon.sqrta();
00131         monitor.OMEGA_0 = mon.omega_0();
00132         monitor.i_0 = mon.i_0();
00133         monitor.omega = mon.omega();
00134         monitor.OMEGAdot = mon.omegadot();
00135         monitor.idot = mon.idot();
00136         monitor.Cuc = mon.cuc();
00137         monitor.Cus = mon.cus();
00138         monitor.Crc = mon.crc();
00139         monitor.Crs = mon.crs();
00140         monitor.Cic = mon.cic();
00141         monitor.Cis = mon.cis();
00142         monitor.toe = mon.toe();
00143         monitor.toc = mon.toc();
00144         monitor.af0 = mon.af0();
00145         monitor.af1 = mon.af1();
00146         monitor.af2 = mon.af2();
00147         monitor.satClkDrift = mon.satclkdrift();
00148         monitor.dtr = mon.dtr();
00149         monitor.WN = mon.wn();
00150         monitor.tow = mon.tow();
00151
00152         // Galileo-specific parameters
00153         monitor.IOD_ephemeris = mon.iod_ephemeris();
00154         monitor.IOD_nav = mon.iod_nav();
00155         monitor.SISA = mon.sisa();
00156         monitor.E5a_HS = mon.e5a_hs();
00157         monitor.E5b_HS = mon.e5b_hs();
00158         monitor.E1B_HS = mon.e1b_hs();
00159         monitor.E5a_DVS = mon.e5a_dvs();
00160         monitor.E5b_DVS = mon.e5b_dvs();
00161         monitor.E1B_DVS = mon.e1b_dvs();
00162         monitor.BGD_E1E5a = mon.bgd_ele5a();
00163         monitor.BGD_E1E5b = mon.bgd_ele5b();
00164
00165         return monitor;
00166     }
00167
00168 private:
00169     gnss_sdr::GalileoEphemeris monitor_{};
00170 };
00171
00172
00173 /** \} */
00174 /** \} */
00175 #endif // GGNSS_SDR_SERDES_GALILEO_EPH_H

```

## 11.282 serdes\_gps\_eph.h File Reference

Serialization / Deserialization of [Gps\\_Ephemeris](#) objects using Protocol Buffers.

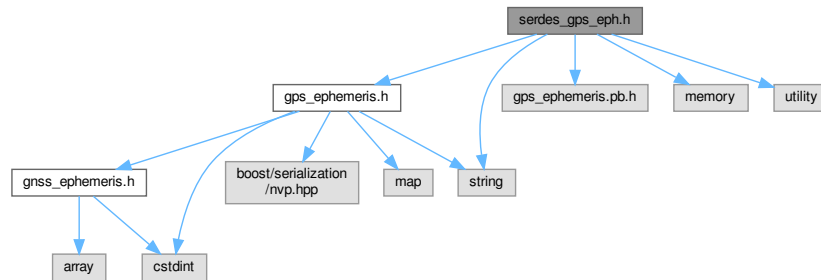
```

#include "gps_ephemeris.h"
#include "gps_ephemeris.pb.h"
#include <memory>
#include <string>
#include <utility>

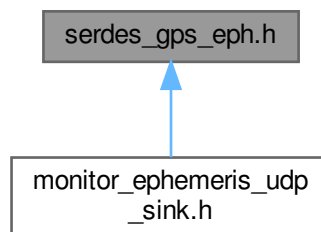
```



Include dependency graph for serdes\_gps\_eph.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Serdes\\_Gps\\_Eph](#)

*This class implements serialization and deserialization of [Gps\\_Ephemeris](#) objects using Protocol Buffers.*

### 11.282.1 Detailed Description

Serialization / Deserialization of [Gps\\_Ephemeris](#) objects using Protocol Buffers.

#### Author

Javier Arribas, 2021. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [serdes\\_gps\\_eph.h](#).

---

## 11.283 serdes\_gps\_eph.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file serdes_gps_eph.h
00003  * \brief Serialization / Deserialization of Gps_Ephemeris objects using
00004  * Protocol Buffers
00005  * \author Javier Arribas, 2021. jarribas(at)cttc.es
00006  *
00007  * -----
  
```

```

00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_SERDES_GPS_EPH_H
00019 #define GNSS_SDR_SERDES_GPS_EPH_H
00020
00021 #include "gps_ephemeris.h"
00022 #include "gps_ephemeris.pb.h" // file created by Protocol Buffers at compile time
00023 #include <memory>
00024 #include <string>
00025 #include <utility>
00026
00027 /** \addtogroup PVT
00028  * \{ */
00029 /** \addtogroup PVT_libs
00030  * \{ */
00031
00032 /*!
00033  * \brief This class implements serialization and deserialization of
00034  * Gps_Ephemeris objects using Protocol Buffers.
00035  */
00036 class Serdes_Gps_Eph
00037 {
00038 public:
00039     Serdes_Gps_Eph()
00040     {
00041         // Verify that the version of the library that we linked against is
00042         // compatible with the version of the headers we compiled against.
00043         GOOGLE_PROTOBUF_VERIFY_VERSION;
00044     }
00045
00046     ~Serdes_Gps_Eph()
00047     {
00048         // google::protobuf::ShutdownProtobufLibrary();
00049     }
00050
00051     inline Serdes_Gps_Eph(const Serdes_Gps_Eph& other) noexcept : monitor_(other.monitor_) //!< Copy
00052     constructor
00053     {
00054     }
00055
00056     inline Serdes_Gps_Eph& operator=(const Serdes_Gps_Eph& rhs) noexcept //!< Copy assignment
00057     operator
00058     {
00059         Serdes_Gps_Eph temp(rhs);
00060         std::swap(this->monitor_, temp.monitor_);
00061         return *this;
00062     }
00063
00064     inline Serdes_Gps_Eph(Serdes_Gps_Eph&& other) noexcept : monitor_(std::move(other.monitor_)) //!<
00065     Move constructor
00066     {
00067     }
00068
00069     inline Serdes_Gps_Eph& operator=(Serdes_Gps_Eph&& other) noexcept //!< Move assignment operator
00070     {
00071         std::swap(this->monitor_, other.monitor_);
00072         return *this;
00073     }
00074
00075     inline std::string createProtobuffer(const std::shared_ptr<Gps_Ephemeris> monitor) //!<
00076     Serialization into a string
00077     {
00078         monitor_.Clear();
00079         std::string data;
00080
00081         monitor_.set_prn(monitor->PRN);
00082         monitor_.set_m_0(monitor->M_0);
00083         monitor_.set_delta_n(monitor->delta_n);
00084         monitor_.set_ecc(monitor->ecc);
00085         monitor_.set_sqrtA(monitor->sqrtA);
00086         monitor_.set_omega_0(monitor->OMEGA_0);
00087         monitor_.set_i_0(monitor->i_0);
00088         monitor_.set_omega(monitor->omega);
00089         monitor_.set_omegadot(monitor->OMEGAdot);
00090         monitor_.set_idot(monitor->idot);
00091         monitor_.set_cuc(monitor->Cuc);
00092         monitor_.set_cus(monitor->Cus);
00093         monitor_.set_crc(monitor->Crc);
00094         monitor_.set_crs(monitor->Crs);

```

```

00091     monitor_.set_cic(monitor->Cic);
00092     monitor_.set_cis(monitor->Cis);
00093     monitor_.set_toe(monitor->toe);
00094     monitor_.set_toc(monitor->toc);
00095     monitor_.set_af0(monitor->af0);
00096     monitor_.set_af1(monitor->af1);
00097     monitor_.set_af2(monitor->af2);
00098     monitor_.set_satclkdrift(monitor->satClkDrift);
00099     monitor_.set_dtr(monitor->dtr);
00100     monitor_.set_wn(monitor->WN);
00101     monitor_.set_tow(monitor->tow);
00102
00103     // GPS-specific parameters
00104     monitor_.set_code_on_l2(monitor->code_on_L2);
00105     monitor_.set_l2_p_data_flag(monitor->L2_P_data_flag);
00106     monitor_.set_sv_accuracy(monitor->SV_accuracy);
00107     monitor_.set_sv_health(monitor->SV_health);
00108     monitor_.set_tgd(monitor->TGD);
00109     monitor_.set_iode(monitor->IODC);
00110     monitor_.set_iode_sf2(monitor->IODE_SF2);
00111     monitor_.set_iode_sf3(monitor->IODE_SF3);
00112     monitor_.set_aodo(monitor->AODO);
00113     monitor_.set_fit_interval_flag(monitor->fit_interval_flag);
00114     monitor_.set_spare1(monitor->spare1);
00115     monitor_.set_spare2(monitor->spare2);
00116     monitor_.set_integrity_status_flag(monitor->integrity_status_flag);
00117     monitor_.set_alert_flag(monitor->alert_flag);
00118     monitor_.set_antispoofing_flag(monitor->antispoofing_flag);
00119
00120     monitor_.SerializeToString(&data);
00121     return data;
00122 }
00123
00124 inline Gps_Ephemeris readProtobuffer(const gnss_sdr::GpsEphemeris& mon) const //!<
Deserialization
00125 {
00126     Gps_Ephemeris monitor;
00127
00128     monitor.PRN = mon.prn();
00129     monitor.M_0 = mon.m_0();
00130     monitor.delta_n = mon.delta_n();
00131     monitor.ecc = mon.ecc();
00132     monitor.sqrta = mon.sqrta();
00133     monitor.OMEGA_0 = mon.omega_0();
00134     monitor.i_0 = mon.i_0();
00135     monitor.omega = mon.omega();
00136     monitor.OMEGAdot = mon.omegadot();
00137     monitor.idot = mon.idot();
00138     monitor.Cuc = mon.cuc();
00139     monitor.Cus = mon.cus();
00140     monitor.Crc = mon.crc();
00141     monitor.Crs = mon.crs();
00142     monitor.Cic = mon.cic();
00143     monitor.Cis = mon.cis();
00144     monitor.toe = mon.toe();
00145     monitor.toc = mon.toc();
00146     monitor.af0 = mon.af0();
00147     monitor.af1 = mon.af1();
00148     monitor.af2 = mon.af2();
00149     monitor.satClkDrift = mon.satclkdrift();
00150     monitor.dtr = mon.dtr();
00151     monitor.WN = mon.wn();
00152     monitor.tow = mon.tow();
00153
00154     // GPS-specific parameters
00155     monitor.code_on_L2 = mon.code_on_l2();
00156     monitor.L2_P_data_flag = mon.l2_p_data_flag();
00157     monitor.SV_accuracy = mon.sv_accuracy();
00158     monitor.SV_health = mon.sv_health();
00159     monitor.TGD = mon.tgd();
00160     monitor.IODC = mon.iode();
00161     monitor.IODE_SF2 = mon.iode_sf2();
00162     monitor.IODE_SF3 = mon.iode_sf3();
00163     monitor.AODO = mon.aodo();
00164     monitor.fit_interval_flag = mon.fit_interval_flag();
00165     monitor.spare1 = mon.spare1();
00166     monitor.spare2 = mon.spare2();
00167     monitor.integrity_status_flag = mon.integrity_status_flag();
00168     monitor.alert_flag = mon.alert_flag();
00169     monitor.antispoofing_flag = mon.antispoofing_flag();
00170
00171     return monitor;
00172 }
00173
00174 private:
00175     gnss_sdr::GpsEphemeris monitor_{};
00176 };

```

```

00177
00178
00179 /** \} */
00180 /** \} */
00181 #endif // GNSS_SDR_SERDES_GPS_EPH_H

```

## 11.284 serdes\_monitor\_pvt.h File Reference

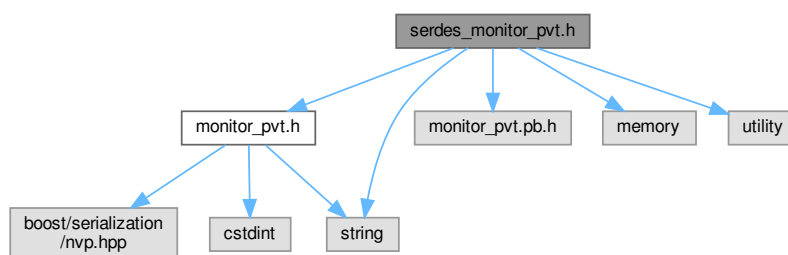
Serialization / Deserialization of [Monitor\\_Pvt](#) objects using Protocol Buffers.

```

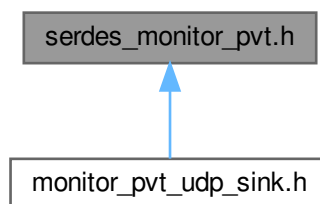
#include "monitor_pvt.h"
#include "monitor_pvt.pb.h"
#include <memory>
#include <string>
#include <utility>

```

Include dependency graph for serdes\_monitor\_pvt.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [Serdes\\_Monitor\\_Pvt](#)

*This class implements serialization and deserialization of [Monitor\\_Pvt](#) objects using Protocol Buffers.*

### 11.284.1 Detailed Description

Serialization / Deserialization of [Monitor\\_Pvt](#) objects using Protocol Buffers.

## Author

Carles Fernandez-Prades, 2019. cfernandez(at)cttc.es

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [serdes\\_monitor\\_pvt.h](#).

## 11.285 serdes\_monitor\_pvt.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file serdes_monitor_pvt.h
00003  * \brief Serialization / Deserialization of Monitor_Pvt objects using
00004  * Protocol Buffers
00005  * \author Carles Fernandez-Prades, 2019. cfernandez(at)cttc.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_SERDES_MONITOR_PVT_H
00019 #define GNSS_SDR_SERDES_MONITOR_PVT_H
00020
00021 #include "monitor_pvt.h"
00022 #include "monitor_pvt.pb.h" // file created by Protocol Buffers at compile time
00023 #include <memory>
00024 #include <string>
00025 #include <utility>
00026
00027 /** \addtogroup PVT
00028  * \{ */
00029 /** \addtogroup PVT_libs
00030  * \{ */
00031
00032
00033 /*!
00034  * \brief This class implements serialization and deserialization of
00035  * Monitor_Pvt objects using Protocol Buffers.
00036  */
00037 class Serdes_Monitor_Pvt
00038 {
00039 public:
00040     Serdes_Monitor_Pvt()
00041     {
00042         // Verify that the version of the library that we linked against is
00043         // compatible with the version of the headers we compiled against.
00044         GOOGLE_PROTOBUF_VERIFY_VERSION;
00045     }
00046
00047     ~Serdes_Monitor_Pvt()
00048     {
00049         // google::protobuf::ShutdownProtobufLibrary();
00050     }
00051
00052     inline Serdes_Monitor_Pvt(const Serdes_Monitor_Pvt& other) noexcept : monitor_(other.monitor_)
00053     /**< Copy constructor
00054     {
00055     }
00056
00057     inline Serdes_Monitor_Pvt& operator=(const Serdes_Monitor_Pvt& rhs) noexcept /**< Copy assignment
00058     operator
00059     {
00060         if (this != &rhs)
00061         {
00062             this->monitor_.CopyFrom(rhs.monitor_);
00063         }
00064         return *this;
00065     }
00066
00067     inline Serdes_Monitor_Pvt(Serdes_Monitor_Pvt&& other) noexcept :
00068     monitor_(std::move(other.monitor_)) /**< Move constructor
00069     {
00070         // Set the other object's monitor_ to a default-constructed state
00071         other.monitor_ = gnss_sdr::MonitorPvt{};
00072     }

```

```

00070
00071     inline Serdes_Monitor_Pvt& operator=(Serdes_Monitor_Pvt&& other) noexcept //!< Move assignment
operator
00072     {
00073         if (this != &other)
00074         {
00075             this->monitor_ = std::move(other.monitor_);
00076         }
00077         return *this;
00078     }
00079
00080     inline std::string createProtobuffer(const Monitor_Pvt* const monitor) //!< Serialization into a
string
00081     {
00082         monitor_.Clear();
00083
00084         std::string data;
00085
00086         monitor_.set_tow_at_current_symbol_ms(monitor->TOW_at_current_symbol_ms);
00087         monitor_.set_week(monitor->week);
00088         monitor_.set_rx_time(monitor->RX_time);
00089         monitor_.set_user_clk_offset(monitor->user_clk_offset);
00090         monitor_.set_pos_x(monitor->pos_x);
00091         monitor_.set_pos_y(monitor->pos_y);
00092         monitor_.set_pos_z(monitor->pos_z);
00093         monitor_.set_vel_x(monitor->vel_x);
00094         monitor_.set_vel_y(monitor->vel_y);
00095         monitor_.set_vel_z(monitor->vel_z);
00096         monitor_.set_cov_xx(monitor->cov_xx);
00097         monitor_.set_cov_yy(monitor->cov_yy);
00098         monitor_.set_cov_zz(monitor->cov_zz);
00099         monitor_.set_cov_xy(monitor->cov_xy);
00100         monitor_.set_cov_yz(monitor->cov_yz);
00101         monitor_.set_cov_zx(monitor->cov_zx);
00102         monitor_.set_latitude(monitor->latitude);
00103         monitor_.set_longitude(monitor->longitude);
00104         monitor_.set_height(monitor->height);
00105         monitor_.set_valid_sats(monitor->valid_sats);
00106         monitor_.set_solution_status(monitor->solution_status);
00107         monitor_.set_solution_type(monitor->solution_type);
00108         monitor_.set_ar_ratio_factor(monitor->AR_ratio_factor);
00109         monitor_.set_ar_ratio_threshold(monitor->AR_ratio_threshold);
00110         monitor_.set_gdop(monitor->gdop);
00111         monitor_.set_pdop(monitor->pdop);
00112         monitor_.set_hdop(monitor->hdop);
00113         monitor_.set_vdop(monitor->vdop);
00114         monitor_.set_user_clk_drift_ppm(monitor->user_clk_drift_ppm);
00115         monitor_.set_utc_time(monitor->utc_time);
00116         monitor_.set_vel_e(monitor->vel_e);
00117         monitor_.set_vel_n(monitor->vel_n);
00118         monitor_.set_vel_u(monitor->vel_u);
00119         monitor_.set_cog(monitor->cog);
00120         monitor_.set_galhas_status(monitor->galhas_status);
00121         monitor_.set_geohash(monitor->geohash);
00122
00123         monitor_.SerializeToString(&data);
00124         return data;
00125     }
00126
00127     inline Monitor_Pvt readProtobuffer(const gnss_sdr::MonitorPvt& mon) const //!< Deserialization
00128     {
00129         Monitor_Pvt monitor;
00130
00131         monitor.TOW_at_current_symbol_ms = mon.tow_at_current_symbol_ms();
00132         monitor.week = mon.week();
00133         monitor.RX_time = mon.rx_time();
00134         monitor.user_clk_offset = mon.user_clk_offset();
00135         monitor.pos_x = mon.pos_x();
00136         monitor.pos_y = mon.pos_y();
00137         monitor.pos_z = mon.pos_z();
00138         monitor.vel_x = mon.vel_x();
00139         monitor.vel_y = mon.vel_y();
00140         monitor.vel_z = mon.vel_z();
00141         monitor.cov_xx = mon.cov_xx();
00142         monitor.cov_yy = mon.cov_yy();
00143         monitor.cov_zz = mon.cov_zz();
00144         monitor.cov_xy = mon.cov_xy();
00145         monitor.cov_yz = mon.cov_yz();
00146         monitor.cov_zx = mon.cov_zx();
00147         monitor.latitude = mon.latitude();
00148         monitor.longitude = mon.longitude();
00149         monitor.height = mon.height();
00150         monitor.valid_sats = static_cast<uint8_t>(mon.valid_sats());
00151         monitor.solution_status = static_cast<uint8_t>(mon.solution_status());
00152         monitor.solution_type = static_cast<uint8_t>(mon.solution_type());
00153         monitor.AR_ratio_factor = mon.ar_ratio_factor();
00154         monitor.AR_ratio_threshold = mon.ar_ratio_threshold();

```

```

00155     monitor.gdop = mon.gdop();
00156     monitor.pdop = mon.pdop();
00157     monitor.hdop = mon.hdop();
00158     monitor.vdop = mon.vdop();
00159     monitor.user_clk_drift_ppm = mon.user_clk_drift_ppm();
00160     monitor.utc_time = mon.utc_time();
00161     monitor.vel_e = mon.vel_e();
00162     monitor.vel_n = mon.vel_n();
00163     monitor.vel_u = mon.vel_u();
00164     monitor.cog = mon.cog();
00165     monitor.galhas_status = mon.galhas_status();
00166     monitor.geohash = mon.geohash();
00167
00168     return monitor;
00169 }
00170
00171 private:
00172     gnss_sdr::MonitorPvt monitor_{};
00173 };
00174
00175
00176 /** \} */
00177 /** \} */
00178 #endif // GNSS_SDR_SERDES_MONITOR_PVT_H

```

## 11.286 direct\_resampler\_conditioner.h File Reference

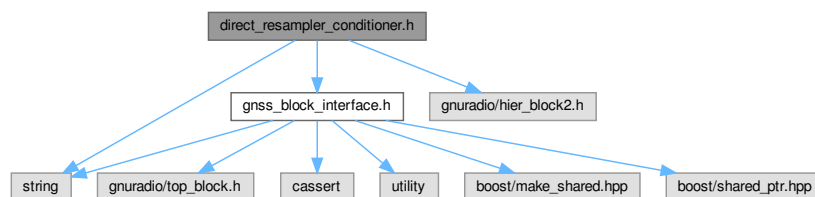
Interface of an adapter of a direct resampler conditioner block to a SignalConditionerInterface.

```

#include "gnss_block_interface.h"
#include <gnuradio/hier_block2.h>
#include <string>

```

Include dependency graph for direct\_resampler\_conditioner.h:



### Classes

- class [DirectResamplerConditioner](#)

*Interface of an adapter of a direct resampler conditioner block to a SignalConditionerInterface.*

### 11.286.1 Detailed Description

Interface of an adapter of a direct resampler conditioner block to a SignalConditionerInterface.

#### Author

Carlos Aviles, 2010. carlos.avilesr(at)gmail.com

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [direct\\_resampler\\_conditioner.h](#).

---

## 11.287 direct\_resampler\_conditioner.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002 * \file direct_resampler_conditioner.h

```

```

00003  * \brief Interface of an adapter of a direct resampler conditioner block
00004  * to a SignalConditionerInterface
00005  * \author Carlos Aviles, 2010. carlos.avilesr(at)googlemail.com
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018
00019 #ifndef GNSS_SDR_DIRECT_RESAMPLER_CONDITIONER_H
00020 #define GNSS_SDR_DIRECT_RESAMPLER_CONDITIONER_H
00021
00022 #include "gnss_block_interface.h"
00023 #include <gnuradio/hier_block2.h>
00024 #include <string>
00025
00026 /** \addtogroup Resampler
00027  * \{ */
00028 /** \addtogroup Resampler_adapters
00029  * \{ */
00030
00031
00032 class ConfigurationInterface;
00033
00034 /*!
00035  * \brief Interface of an adapter of a direct resampler conditioner block
00036  * to a SignalConditionerInterface
00037  */
00038 class DirectResamplerConditioner : public GNSSBlockInterface
00039 {
00040 public:
00041     DirectResamplerConditioner(const ConfigurationInterface* configuration,
00042                               const std::string& role, unsigned int in_stream,
00043                               unsigned int out_stream);
00044
00045     ~DirectResamplerConditioner() = default;
00046
00047     inline std::string role() override
00048     {
00049         return role_;
00050     }
00051
00052     ///! Returns "Direct_Resampler"
00053     inline std::string implementation() override
00054     {
00055         return "Direct_Resampler";
00056     }
00057
00058     inline size_t item_size() override
00059     {
00060         return item_size_;
00061     }
00062
00063     void connect(gr::top_block_sptr top_block) override;
00064     void disconnect(gr::top_block_sptr top_block) override;
00065     gr::basic_block_sptr get_left_block() override;
00066     gr::basic_block_sptr get_right_block() override;
00067
00068 private:
00069     gr::block_sptr resampler_;
00070     gr::block_sptr file_sink_;
00071     std::string role_;
00072     std::string item_type_;
00073     std::string dump_filename_;
00074     double sample_freq_in_;
00075     double sample_freq_out_;
00076     size_t item_size_;
00077     unsigned int in_stream_;
00078     unsigned int out_stream_;
00079     bool dump_;
00080 };
00081
00082
00083 /** \} */
00084 /** \} */
00085 #endif // GNSS_SDR_DIRECT_RESAMPLER_CONDITIONER_H

```

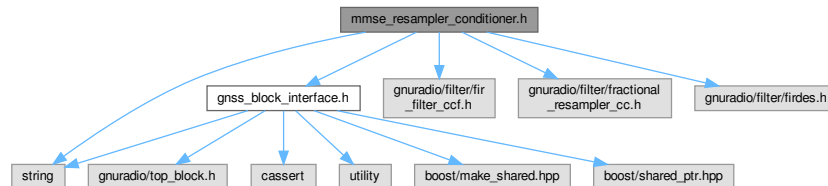


## 11.288 mmse\_resampler\_conditioner.h File Reference

Interface of an adapter of a mmse resampler conditioner block to a SignalConditionerInterface.

```
#include "gnss_block_interface.h"
#include <gnuradio/filter/fir_filter_ccf.h>
#include <gnuradio/filter/fractional_resampler_cc.h>
#include <gnuradio/filter/firdes.h>
#include <string>
```

Include dependency graph for mmse\_resampler\_conditioner.h:



### Classes

- class [MmseResamplerConditioner](#)

*Interface of a MMSE resampler block adapter to a SignalConditionerInterface.*

### 11.288.1 Detailed Description

Interface of an adapter of a mmse resampler conditioner block to a SignalConditionerInterface.

#### Author

Antonio Ramos, 2018. antonio.ramos(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [mmse\\_resampler\\_conditioner.h](#).

---

## 11.289 mmse\_resampler\_conditioner.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file mmse_resampler_conditioner.h
00003  * \brief Interface of an adapter of a mmse resampler conditioner block
00004  * to a SignalConditionerInterface
00005  * \author Antonio Ramos, 2018. antonio.ramos(at)cttc.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018
00019 #ifndef GNSS_SDR_MMSE_RESAMPLER_CONDITIONER_H
00020 #define GNSS_SDR_MMSE_RESAMPLER_CONDITIONER_H
00021
00022 #include "gnss_block_interface.h"
00023 #ifdef GR_GREATER_38
00024 #include <gnuradio/filter/fir_filter_blk.h>
00025 #include <gnuradio/filter/mmse_resampler_cc.h>
00026 #else
00027 #include <gnuradio/filter/fir_filter_ccf.h>
```

```

00028 #include <gnuradio/filter/fractional_resampler_cc.h>
00029 #endif
00030
00031 #include <gnuradio/filter/firdes.h>
00032 #include <string>
00033
00034 /** \addtogroup Resampler
00035  * Classes for input signal resampling
00036  * \{ */
00037 /** \addtogroup Resampler_adapters resampler_adapters
00038  * Classes that wrap GNU Radio resampler blocks with a GNSSBlockInterface
00039  * \{ */
00040
00041
00042 class ConfigurationInterface;
00043
00044 /**
00045  * \brief Interface of a MMSE resampler block adapter
00046  * to a SignalConditionerInterface
00047  */
00048 class MmseResamplerConditioner : public GNSSBlockInterface
00049 {
00050 public:
00051     MmseResamplerConditioner(const ConfigurationInterface* configuration,
00052                             const std::string& role, unsigned int in_stream,
00053                             unsigned int out_stream);
00054
00055     ~MmseResamplerConditioner() = default;
00056
00057     inline std::string role() override
00058     {
00059         return role_;
00060     }
00061
00062     inline std::string implementation() override
00063     {
00064         return "Mmse_Resampler";
00065     }
00066
00067     inline size_t item_size() override
00068     {
00069         return item_size_;
00070     }
00071
00072     void connect(gr::top_block_sptr top_block) override;
00073     void disconnect(gr::top_block_sptr top_block) override;
00074     gr::basic_block_sptr get_left_block() override;
00075     gr::basic_block_sptr get_right_block() override;
00076
00077 private:
00078     #ifdef GR_GREATER_38
00079         gr::filter::mmse_resampler_cc::sptr resampler_;
00080     #else
00081         gr::filter::fractional_resampler_cc::sptr resampler_;
00082     #endif
00083     gr::filter::fir_filter_ccf::sptr fir_filter_ccf_;
00084     gr::block_sptr file_sink_;
00085     std::string role_;
00086     std::string item_type_;
00087     std::string dump_filename_;
00088     size_t item_size_;
00089     double sample_freq_in_;
00090     double sample_freq_out_;
00091     unsigned int in_stream_;
00092     unsigned int out_stream_;
00093     bool dump_;
00094 };
00095
00096
00097 /** \} */
00098 /** \} */
00099 #endif // GNSS_SDR_FRACTIONAL_RESAMPLER_CONDITIONER_H

```

## 11.290 direct\_resampler\_conditioner\_cb.h File Reference

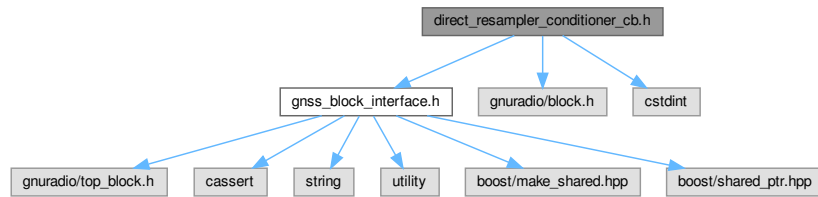
Nearest neighborhood resampler with `std::complex<signed char>` input and `std::complex<signed char>` output.

```

#include "gnss_block_interface.h"
#include <gnuradio/block.h>
#include <cstdint>

```

Include dependency graph for direct\_resampler\_conditioner\_cb.h:



## Classes

- class [direct\\_resampler\\_conditioner\\_cb](#)  
*This class implements a direct resampler conditioner for `std::complex<signed char>`*

## Typedefs

- using [direct\\_resampler\\_conditioner\\_cb\\_sptr](#) = `gnss_shared_ptr<direct\_resampler\_conditioner\_cb>`

## Functions

- `direct_resampler_conditioner_cb_sptr` **direct\_resampler\_make\_conditioner\_cb** (double sample\_freq\_in, double sample\_freq\_out)

### 11.290.1 Detailed Description

Nearest neighborhood resampler with `std::complex<signed char>` input and `std::complex<signed char>` output.

#### Author

Luis Esteve, 2011. [luis\(at\)epsilon-formacion.com](mailto:luis(at)epsilon-formacion.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [direct\\_resampler\\_conditioner\\_cb.h](#).

---

## 11.291 direct\_resampler\_conditioner\_cb.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file direct_resampler_conditioner_cb.h
00003  * \brief Nearest neighborhood resampler with
00004  *        std::complex<signed char> input and std::complex<signed char> output
00005  * \author Luis Esteve, 2011. luis\(at\)epsilon-formacion.com
00006  *
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019 #ifndef GNSS_SDR_DIRECT_RESAMPLER_CONDITIONER_CB_H
00020 #define GNSS_SDR_DIRECT_RESAMPLER_CONDITIONER_CB_H
00021
00022 #include "gnss_block_interface.h"
00023 #include <gnuradio/block.h>
00024 #include <cstdint>
00025

```

```

00026 /** \addtogroup Resampler
00027 * \{ */
00028 /** \addtogroup Resampler_gnuradio_blocks
00029 * GNU Radio blocks for input signal resampling
00030 * \{ */
00031
00032
00033 class direct_resampler_conditioner_cb;
00034
00035 using direct_resampler_conditioner_cb_sptr = gnss_shared_ptr<direct_resampler_conditioner_cb>;
00036
00037 direct_resampler_conditioner_cb_sptr direct_resampler_make_conditioner_cb(
00038     double sample_freq_in,
00039     double sample_freq_out);
00040
00041 /*!
00042 * \brief This class implements a direct resampler conditioner for std::complex<signed char>
00043 *
00044 * Direct resampling without interpolation
00045 */
00046 class direct_resampler_conditioner_cb : public gr::block
00047 {
00048 public:
00049     ~direct_resampler_conditioner_cb() = default;
00050
00051     inline unsigned int sample_freq_in() const
00052     {
00053         return d_sample_freq_in;
00054     }
00055
00056     inline unsigned int sample_freq_out() const
00057     {
00058         return d_sample_freq_out;
00059     }
00060
00061     void forecast(int noutput_items, gr_vector_int &ninput_items_required);
00062
00063     int general_work(int noutput_items, gr_vector_int &ninput_items,
00064         gr_vector_const_void_star &input_items,
00065         gr_vector_void_star &output_items);
00066
00067 private:
00068     friend direct_resampler_conditioner_cb_sptr direct_resampler_make_conditioner_cb(
00069         double sample_freq_in,
00070         double sample_freq_out);
00071
00072     direct_resampler_conditioner_cb(
00073         double sample_freq_in,
00074         double sample_freq_out);
00075
00076     double d_sample_freq_in;
00077     double d_sample_freq_out;
00078     uint32_t d_phase;
00079     uint32_t d_lphase;
00080     uint32_t d_phase_step;
00081 };
00082
00083
00084 /** \} */
00085 /** \} */
00086 #endif // GNSS_SDR_DIRECT_RESAMPLER_CONDITIONER_CB_H

```

## 11.292 direct\_resampler\_conditioner\_cc.h File Reference

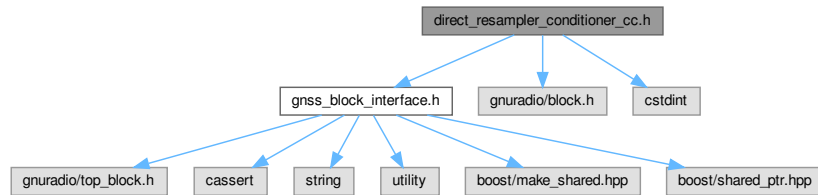
Nearest neighborhood resampler with gr\_complex input and gr\_complex output.

```

#include "gnss_block_interface.h"
#include <gnuradio/block.h>
#include <stdint>

```

Include dependency graph for direct\_resampler\_conditioner\_cc.h:



## Classes

- class [direct\\_resampler\\_conditioner\\_cc](#)  
*This class implements a direct resampler conditioner for complex data.*

## Typedefs

- using [direct\\_resampler\\_conditioner\\_cc\\_sptr](#) = gnss\_shared\_ptr<[direct\\_resampler\\_conditioner\\_cc](#)>

## Functions

- [direct\\_resampler\\_conditioner\\_cc\\_sptr](#) **direct\_resampler\_make\_conditioner\_cc** (double sample\_freq\_in, double sample\_freq\_out)

### 11.292.1 Detailed Description

Nearest neighborhood resampler with gr\_complex input and gr\_complex output.

#### Author

Luis Esteve, 2011. [luis\(at\)epsilon-formacion.com](mailto:luis(at)epsilon-formacion.com)

This block takes in a signal stream and performs direct resampling. The theory behind this block can be found in Chapter 7.5 of the following book: R. Lyons, Understanding Digital Signal Processing, 3rd ed., Pearson Education, 2010.

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [direct\\_resampler\\_conditioner\\_cc.h](#).

## 11.293 direct\_resampler\_conditioner\_cc.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file direct_resampler_conditioner_cc.h
00003  * 
00004  * \brief Nearest neighborhood resampler with
00005  *  gr_complex input and gr_complex output
00006  * \author Luis Esteve, 2011. luis\(at\)epsilon-formacion.com
00007  * 
00008  * This block takes in a signal stream and performs direct resampling.
00009  * The theory behind this block can be found in Chapter 7.5 of the following
00010  * book:
00011  * R. Lyons, Understanding Digital Signal Processing, 3rd ed., Pearson Education,
00012  * 2010.
00013  * 
00014  * -----
00015  * 
00016  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00017  * This file is part of GNSS-SDR.
00018  * 
00019  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00020  * SPDX-License-Identifier: GPL-3.0-or-later
00021  * 

```

```

00022  * -----
00023  */
00024
00025 #ifndef GNSS_SDR_DIRECT_RESAMPLER_CONDITIONER_CC_H
00026 #define GNSS_SDR_DIRECT_RESAMPLER_CONDITIONER_CC_H
00027
00028 #include "gnss_block_interface.h"
00029 #include <gnuradio/block.h>
00030 #include <stdint>
00031
00032 /** \addtogroup Resampler
00033  * \{ */
00034 /** \addtogroup Resampler_gnuradio_blocks resampler_gr_blocks
00035  * \{ */
00036
00037
00038 class direct_resampler_conditioner_cc;
00039
00040 using direct_resampler_conditioner_cc_sptr = gnss_shared_ptr<direct_resampler_conditioner_cc>;
00041
00042 direct_resampler_conditioner_cc_sptr direct_resampler_make_conditioner_cc(
00043     double sample_freq_in,
00044     double sample_freq_out);
00045
00046 /*!
00047  * \brief This class implements a direct resampler conditioner for complex data
00048  *
00049  * Direct resampling without interpolation
00050  */
00051 class direct_resampler_conditioner_cc : public gr::block
00052 {
00053 public:
00054     ~direct_resampler_conditioner_cc() = default;
00055     inline unsigned int sample_freq_in() const
00056     {
00057         return d_sample_freq_in;
00058     }
00059
00060     inline unsigned int sample_freq_out() const
00061     {
00062         return d_sample_freq_out;
00063     }
00064
00065     void forecast(int noutput_items, gr_vector_int &ninput_items_required);
00066
00067     int general_work(int noutput_items, gr_vector_int &ninput_items,
00068         gr_vector_const_void_star &input_items,
00069         gr_vector_void_star &output_items);
00070
00071 private:
00072     friend direct_resampler_conditioner_cc_sptr direct_resampler_make_conditioner_cc(
00073         double sample_freq_in,
00074         double sample_freq_out);
00075
00076     direct_resampler_conditioner_cc(
00077         double sample_freq_in,
00078         double sample_freq_out);
00079
00080     double d_sample_freq_in; // Sampling frequency of the input signal
00081     double d_sample_freq_out; // Sampling frequency of the output signal
00082     uint32_t d_phase;
00083     uint32_t d_lphase;
00084     uint32_t d_phase_step;
00085 };
00086
00087
00088 /** \} */
00089 /** \} */
00090 #endif // GNSS_SDR_DIRECT_RESAMPLER_CONDITIONER_CC_H

```

## 11.294 direct\_resampler\_conditioner\_cs.h File Reference

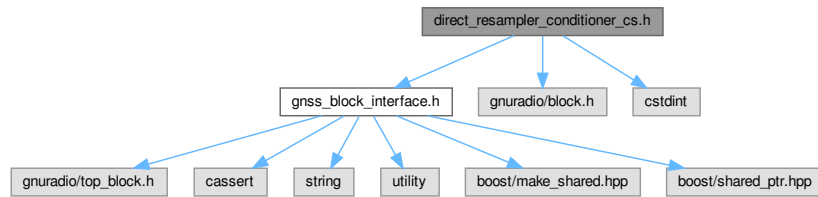
Nearest neighborhood resampler with std::complex<short> input and std::complex<short> output.

```

#include "gnss_block_interface.h"
#include <gnuradio/block.h>
#include <stdint>

```

Include dependency graph for direct\_resampler\_conditioner\_cs.h:



## Classes

- class [direct\\_resampler\\_conditioner\\_cs](#)  
*This class implements a direct resampler conditioner for `std::complex<short>`*

## Typedefs

- using [direct\\_resampler\\_conditioner\\_cs\\_sptr](#) = `gnss_shared_ptr<direct\_resampler\_conditioner\_cs>`

## Functions

- `direct_resampler_conditioner_cs_sptr` **direct\_resampler\_make\_conditioner\_cs** (double sample\_freq\_in, double sample\_freq\_out)

### 11.294.1 Detailed Description

Nearest neighborhood resampler with `std::complex<short>` input and `std::complex<short>` output.

#### Author

Luis Esteve, 2011. [luis\(at\)epsilon-formacion.com](mailto:luis(at)epsilon-formacion.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [direct\\_resampler\\_conditioner\\_cs.h](#).

---

## 11.295 direct\_resampler\_conditioner\_cs.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file direct_resampler_conditioner_cs.h
00003  * \brief Nearest neighborhood resampler with
00004  *        std::complex<short> input and std::complex<short> output
00005  * \author Luis Esteve, 2011. luis\(at\)epsilon-formacion.com
00006  *
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019 #ifndef GNSS_SDR_DIRECT_RESAMPLER_CONDITIONER_CS_H
00020 #define GNSS_SDR_DIRECT_RESAMPLER_CONDITIONER_CS_H
00021
00022 #include "gnss_block_interface.h"
00023 #include <gnuradio/block.h>
00024 #include <cstdint>
00025

```

```

00026 /** \addtogroup Resampler
00027 * \{ */
00028 /** \addtogroup Resampler_gnuradio_blocks
00029 * \{ */
00030
00031
00032 class direct_resampler_conditioner_cs;
00033
00034 using direct_resampler_conditioner_cs_sptr = gnss_shared_ptr<direct_resampler_conditioner_cs>;
00035
00036 direct_resampler_conditioner_cs_sptr direct_resampler_make_conditioner_cs(
00037     double sample_freq_in,
00038     double sample_freq_out);
00039
00040 /*!
00041 * \brief This class implements a direct resampler conditioner for std::complex<short>
00042 *
00043 * Direct resampling without interpolation
00044 */
00045 class direct_resampler_conditioner_cs : public gr::block
00046 {
00047 public:
00048     ~direct_resampler_conditioner_cs() = default;
00049
00050     inline unsigned int sample_freq_in() const
00051     {
00052         return d_sample_freq_in;
00053     }
00054
00055     inline unsigned int sample_freq_out() const
00056     {
00057         return d_sample_freq_out;
00058     }
00059
00060     void forecast(int noutput_items, gr_vector_int &ninput_items_required);
00061
00062     int general_work(int noutput_items, gr_vector_int &ninput_items,
00063         gr_vector_const_void_star &input_items,
00064         gr_vector_void_star &output_items);
00065
00066 private:
00067     friend direct_resampler_conditioner_cs_sptr direct_resampler_make_conditioner_cs(
00068         double sample_freq_in,
00069         double sample_freq_out);
00070
00071     direct_resampler_conditioner_cs(
00072         double sample_freq_in,
00073         double sample_freq_out);
00074
00075     double d_sample_freq_in;
00076     double d_sample_freq_out;
00077     uint32_t d_phase;
00078     uint32_t d_lphase;
00079     uint32_t d_phase_step;
00080 };
00081
00082
00083 /** \} */
00084 /** \} */
00085 #endif // GNSS_SDR_DIRECT_RESAMPLER_CONDITIONER_CS_H

```

## 11.296 signal\_generator.h File Reference

Adapter of a class that generates synthesized GNSS signal.

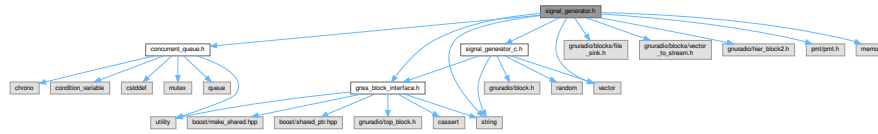
```

#include "concurrent_queue.h"
#include "gnss_block_interface.h"
#include "signal_generator_c.h"
#include <gnuradio/blocks/file_sink.h>
#include <gnuradio/blocks/vector_to_stream.h>
#include <gnuradio/hier_block2.h>
#include <pmt/pmt.h>
#include <memory>
#include <string>
#include <vector>

```



Include dependency graph for signal\_generator.h:



## Classes

- class [SignalGenerator](#)

*This class generates synthesized GNSS signal.*

### 11.296.1 Detailed Description

Adapter of a class that generates synthesized GNSS signal.

#### Author

Marc Molina, 2013. [marc.molina.pena@gmail.com](mailto:marc.molina.pena@gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [signal\\_generator.h](#).

---

## 11.297 signal\_generator.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file signal_generator.h
00003  * \brief Adapter of a class that generates synthesized GNSS signal.
00004  * \author Marc Molina, 2013. marc.molina.pena@gmail.com
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018
00019 #ifndef GNSS_SDR_SIGNAL_GENERATOR_H
00020 #define GNSS_SDR_SIGNAL_GENERATOR_H
00021
00022 #include "concurrent_queue.h"
00023 #include "gnss_block_interface.h"
00024 #include "signal_generator_c.h"
00025 #include <gnuradio/blocks/file_sink.h>
00026 #include <gnuradio/blocks/vector_to_stream.h>
00027 #include <gnuradio/hier_block2.h>
00028 #include <pmt/pmt.h>
00029 #include <memory>
00030 #include <string>
00031 #include <vector>
00032
00033 class ConfigurationInterface;
00034
00035 /*!
00036 * \brief This class generates synthesized GNSS signal.
00037 *
00038 */
00039 class SignalGenerator : public GNSSBlockInterface
00040 {
00041 public:
00042     SignalGenerator(const ConfigurationInterface* configuration,
00043                   const std::string& role, unsigned int in_stream,
00044                   unsigned int out_stream, Concurrent_Queue<pmt::pmt_t>* queue);

```

```

00045
00046     ~SignalGenerator() = default;
00047
00048     inline std::string role() override
00049     {
00050         return role_;
00051     }
00052
00053     /*!
00054     * \brief Returns "GNSSSignalGenerator".
00055     */
00056     inline std::string implementation() override
00057     {
00058         return "GNSSSignalGenerator";
00059     }
00060
00061     inline size_t item_size() override
00062     {
00063         return item_size_;
00064     }
00065
00066     void connect(gr::top_block_sptr top_block) override;
00067     void disconnect(gr::top_block_sptr top_block) override;
00068     gr::basic_block_sptr get_left_block() override;
00069     gr::basic_block_sptr get_right_block() override;
00070
00071 private:
00072     gnss_shared_ptr<gr::block> gen_source_;
00073     gr::blocks::vector_to_stream::sptr vector_to_stream_;
00074     gr::blocks::file_sink::sptr file_sink_;
00075     std::string role_;
00076     std::string item_type_;
00077     std::string dump_filename_;
00078     size_t item_size_;
00079     unsigned int in_stream_;
00080     unsigned int out_stream_;
00081     bool dump_;
00082 };
00083
00084 #endif // GNSS_SDR_SIGNAL_GENERATOR_H

```

## 11.298 signal\_generator\_c.h File Reference

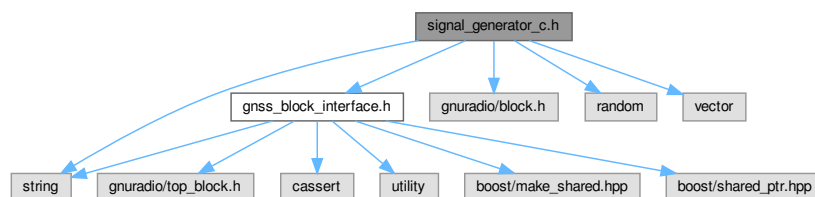
GNU Radio source block that generates synthesized GNSS signal.

```

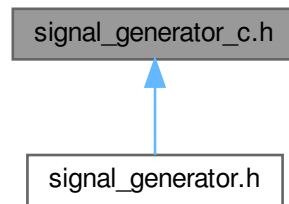
#include "gnss_block_interface.h"
#include <gnuradio/block.h>
#include <random>
#include <string>
#include <vector>

```

Include dependency graph for signal\_generator\_c.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [signal\\_generator\\_c](#)  
*This class generates synthesized GNSS signal.*

## Typedefs

- using [signal\\_generator\\_c\\_sptr](#) = `gnss_shared_ptr<signal\_generator\_c>`

## Functions

- `signal_generator_c_sptr signal\_make\_generator\_c (const std::vector< std::string > &signal1, const std::vector< std::string > &system, const std::vector< unsigned int > &PRN, const std::vector< float > &CN0_dB, const std::vector< float > &doppler_Hz, const std::vector< unsigned int > &delay_chips, const std::vector< unsigned int > &delay_sec, bool data_flag, bool noise_flag, unsigned int fs_in, unsigned int vector_length, float BW_BB)`  
*Return a shared\_ptr to a new instance of gen\_source.*

## 11.298.1 Detailed Description

GNU Radio source block that generates synthesized GNSS signal.

### Author

Marc Molina, 2013. [marc.molina.pena@gmail.com](mailto:marc.molina.pena@gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [signal\\_generator\\_c.h](#).

---

## 11.298.2 Typedef Documentation

### 11.298.2.1 signal\_generator\_c\_sptr

`using signal_generator_c_sptr = gnss_shared_ptr<signal\_generator\_c>`  
Definition at line 29 of file [signal\\_generator\\_c.h](#).

## 11.298.3 Function Documentation

### 11.298.3.1 signal\_make\_generator\_c()

```

signal_generator_c_sptr signal_make_generator_c (
    const std::vector< std::string > & signal1,
    const std::vector< std::string > & system,

```

```

const std::vector< unsigned int > & PRN,
const std::vector< float > & CNO_dB,
const std::vector< float > & doppler_Hz,
const std::vector< unsigned int > & delay_chips,
const std::vector< unsigned int > & delay_sec,
bool data_flag,
bool noise_flag,
unsigned int fs_in,
unsigned int vector_length,
float BW_BB)

```

Return a shared\_ptr to a new instance of gen\_source.

To avoid accidental use of raw pointers, gen\_source's constructor is private. signal\_make\_generator\_c is the public interface for creating new instances.

## 11.299 signal\_generator\_c.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file signal_generator_c.h
00003  * \brief GNU Radio source block that generates synthesized GNSS signal.
00004  * \author Marc Molina, 2013. marc.molina.pena@gmail.com
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_SIGNAL_GENERATOR_C_H
00018 #define GNSS_SDR_SIGNAL_GENERATOR_C_H
00019
00020 #include "gnss_block_interface.h"
00021 #include <gnuradio/block.h>
00022 #include <random>
00023 #include <string>
00024 #include <vector>
00025
00026
00027 class signal_generator_c;
00028
00029 using signal_generator_c_sptr = gnss_shared_ptr<signal_generator_c>;
00030
00031 /*!
00032  * \brief Return a shared_ptr to a new instance of gen_source.
00033  *
00034  * To avoid accidental use of raw pointers, gen_source's
00035  * constructor is private. signal_make_generator_c is the public
00036  * interface for creating new instances.
00037  */
00038 signal_generator_c_sptr signal_make_generator_c(
00039     const std::vector<std::string> &signall,
00040     const std::vector<std::string> &system,
00041     const std::vector<unsigned int> &PRN,
00042     const std::vector<float> &CNO_dB,
00043     const std::vector<float> &doppler_Hz,
00044     const std::vector<unsigned int> &delay_chips,
00045     const std::vector<unsigned int> &delay_sec,
00046     bool data_flag,
00047     bool noise_flag,
00048     unsigned int fs_in,
00049     unsigned int vector_length,
00050     float BW_BB);
00051
00052 /*!
00053  * \brief This class generates synthesized GNSS signal.
00054  * \ingroup block
00055  *
00056  * \sa gen_source for a version that subclasses gr_block.
00057  */
00058 class signal_generator_c : public gr::block
00059 {
00060 public:
00061     ~signal_generator_c() = default; // public destructor

```

```

00062
00063 // Where all the action really happens
00064 int general_work(int noutput_items,
00065                 gr_vector_int &ninput_items,
00066                 gr_vector_const_void_star &input_items,
00067                 gr_vector_void_star &output_items);
00068
00069 private:
00070 friend signal_generator_c_sptr signal_make_generator_c(
00071     const std::vector<std::string> &signal1,
00072     const std::vector<std::string> &system,
00073     const std::vector<unsigned int> &PRN,
00074     const std::vector<float> &CN0_dB,
00075     const std::vector<float> &doppler_Hz,
00076     const std::vector<unsigned int> &delay_chips,
00077     const std::vector<unsigned int> &delay_sec,
00078     bool data_flag,
00079     bool noise_flag,
00080     unsigned int fs_in,
00081     unsigned int vector_length,
00082     float BW_BB);
00083
00084 signal_generator_c(
00085     std::vector<std::string> signal1,
00086     std::vector<std::string> system,
00087     const std::vector<unsigned int> &PRN,
00088     std::vector<float> CN0_dB,
00089     std::vector<float> doppler_Hz,
00090     std::vector<unsigned int> delay_chips,
00091     std::vector<unsigned int> delay_sec,
00092     bool data_flag,
00093     bool noise_flag,
00094     unsigned int fs_in,
00095     unsigned int vector_length,
00096     float BW_BB);
00097
00098 void init();
00099
00100 void generate_codes();
00101
00102 std::random_device r;
00103 std::uniform_int_distribution<int> uniform_dist;
00104 std::normal_distribution<float> normal_dist;
00105 std::vector<std::string> signal_;
00106 std::vector<std::string> system_;
00107 std::vector<std::vector<gr_complex>> sampled_code_data_;
00108 std::vector<std::vector<gr_complex>> sampled_code_pilot_;
00109 std::vector<gr_complex> current_data_bits_;
00110 std::vector<gr_complex> complex_phase_;
00111 std::vector<float> CN0_dB_;
00112 std::vector<float> doppler_Hz_;
00113 std::vector<float> start_phase_rad_;
00114 std::vector<unsigned int> PRN_;
00115 std::vector<unsigned int> delay_chips_;
00116 std::vector<unsigned int> delay_sec_;
00117 std::vector<unsigned int> samples_per_code_;
00118 std::vector<unsigned int> num_of_codes_per_vector_;
00119 std::vector<unsigned int> data_bit_duration_ms_;
00120 std::vector<unsigned int> ms_counter_;
00121 std::vector<signed int> current_data_bit_int_;
00122 std::vector<signed int> data_modulation_;
00123 std::vector<signed int> pilot_modulation_;
00124 float BW_BB_;
00125 unsigned int work_counter_{};
00126 unsigned int fs_in_;
00127 unsigned int num_sats_;
00128 unsigned int vector_length_;
00129 bool data_flag_;
00130 bool noise_flag_;
00131 };
00132
00133 #endif // GNSS_SDR_SIGNAL_GENERATOR_C_H

```

## 11.300 ad936x\_custom\_signal\_source.h File Reference

A direct IIO custom front-end gnss-sdr signal source for the AD936x AD front-end family with special FPGA custom functionalities.

```

#include "ad936x_iio_source.h"
#include "concurrent_queue.h"
#include "conjugate_cc.h"
#include "signal_source_base.h"

```

```
#include "unpack_byte_2bit_cpx_samples.h"
#include "unpack_byte_4bit_samples.h"
#include "unpack_short_byte_samples.h"
#include <gnuradio/blocks/char_to_short.h>
#include <gnuradio/blocks/file_sink.h>
#include <gnuradio/blocks/delay.h>
#include <gnuradio/blocks/interleaved_short_to_complex.h>
#include <pmt/pmt.h>
#include <cstdint>
#include <memory>
#include <stdexcept>
#include <string>
#include <vector>
```

Include dependency graph for ad936x\_custom\_signal\_source.h:



## Classes

- class [Ad936xCustomSignalSource](#)

*This class instantiates the Ad936xCustom gnuradio signal source. It has support also for a customized Ad936x Custom firmware and signal source to support PPS samplestamp reading.*

### 11.300.1 Detailed Description

A direct IIO custom front-end gnss-sdr signal source for the AD936x AD front-end family with special FPGA custom functionalities.

#### Author

Javier Arribas, jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [ad936x\\_custom\\_signal\\_source.h](#).

---

## 11.301 ad936x\_custom\_signal\_source.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file ad936x_custom_signal_source.h
00003  * \brief A direct IIO custom front-end gnss-sdr signal source for the AD936x AD front-end family with
00004  * special FPGA custom functionalities.
00005  * \author Javier Arribas, jarribas(at)cttc.es
00006  *
00007  * -----
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_Ad936xCUSTOM_SIGNAL_SOURCE_H
00018 #define GNSS_SDR_Ad936xCUSTOM_SIGNAL_SOURCE_H
00019
00020 #include "ad936x_iio_source.h"
00021 #include "concurrent_queue.h"
00022 #include "conjugate_cc.h"
00023 #include "signal_source_base.h"
00024 #include "unpack_byte_2bit_cpx_samples.h"
```

```

00025 #include "unpack_byte_4bit_samples.h"
00026 #include "unpack_short_byte_samples.h"
00027 #include <gnuradio/blocks/char_to_short.h>
00028 #include <gnuradio/blocks/file_sink.h>
00029 // #include <gnuradio/blocks/interleaved_char_to_complex.h>
00030 #include <gnuradio/blocks/delay.h>
00031 #include <gnuradio/blocks/interleaved_short_to_complex.h>
00032 #include <pmt/pmt.h>
00033 #include <stdint>
00034 #include <memory>
00035 #include <stdexcept>
00036 #include <string>
00037 #include <vector>
00038
00039 /** \addtogroup Signal_Source
00040  * \{ */
00041 /** \addtogroup Signal_Source_adapters
00042  * \{ */
00043
00044
00045 class ConfigurationInterface;
00046
00047 /*!
00048  * \brief This class instantiates the Ad936xCustom gnuradio signal source.
00049  * It has support also for a customized Ad936xCustom firmware and signal source to support PPS
00050  * samplestamp reading.
00051 */
00051 class Ad936xCustomSignalSource : public SignalSourceBase
00052 {
00053 public:
00054     Ad936xCustomSignalSource(const ConfigurationInterface* configuration,
00055                             const std::string& role, unsigned int in_stream,
00056                             unsigned int out_stream, Concurrent_Queue<pmt::pmt_t>* queue);
00057
00058     ~Ad936xCustomSignalSource() = default;
00059
00060     inline size_t item_size() override
00061     {
00062         return item_size_;
00063     }
00064
00065     void connect(gr::top_block_sptr top_block) override;
00066     void disconnect(gr::top_block_sptr top_block) override;
00067     gr::basic_block_sptr get_left_block() override;
00068     gr::basic_block_sptr get_right_block() override;
00069     gr::basic_block_sptr get_right_block(int RF_channel) override;
00070
00071 private:
00072     unsigned int in_stream_;
00073     unsigned int out_stream_;
00074     gr::block_sptr ad936x_iio_source;
00075     std::vector<gr::blocks::file_sink::sptr> sink_;
00076     std::vector<std::string> filename_vec_;
00077
00078     gr::blocks::delay::sptr gr_delay;
00079     std::vector<gr::blocks::char_to_short::sptr> gr_char_to_short_;
00080     std::vector<gr::blocks::interleaved_short_to_complex::sptr> gr_interleaved_short_to_complex_;
00081     // std::vector<gr::blocks::interleaved_char_to_complex::sptr> gr_interleaved_char_to_complex_;
00082     std::vector<unpack_short_byte_samples_sptr> unpack_short_byte_;
00083     std::vector<unpack_byte_4bit_samples_sptr> unpack_byte_fourbits;
00084     std::vector<unpack_byte_2bit_cpx_samples_sptr> unpack_byte_twobits;
00085
00086     std::string item_type_;
00087     size_t item_size_;
00088     int64_t samples_;
00089     bool dump_;
00090     std::string dump_filename_;
00091
00092     // Front-end settings
00093     std::string pluto_uri_;
00094     std::string board_type_;
00095     long long sample_rate_;
00096     long long bandwidth_;
00097     long long freq_;
00098     long long freq_2ch;
00099     std::string rf_port_select_;
00100     std::string rf_filter;
00101     std::string gain_mode_rx0_;
00102     std::string gain_mode_rxl_;
00103     double rf_gain_rx0_;
00104     double rf_gain_rxl_;
00105     bool enable_ch0;
00106     bool enable_ch1;
00107     bool PPS_mode_;
00108     std::string fe_ip_;
00109     int fe_ctlport_;
00110     int ssize_;

```

```

00111     int bshift_;
00112     bool spattern_;
00113     bool inverted_spectrum_ch0_;
00114     bool inverted_spectrum_ch1_;
00115     double lo_attenuation_db_;
00116     bool high_side_lo_;
00117     int tx_lo_channel_;
00118     double rx0_to_rx1_delay_ns_;
00119     bool delay_enabled;
00120     bool apply_delay_on_rx0;
00121
00122     std::vector<bool> inverted_spectrum_vec;
00123     int n_channels;
00124 };
00125
00126
00127 /** \} */
00128 /** \} */
00129 #endif // GNSS_SDR_Ad936xCustom_SIGNAL_SOURCE_H

```

## 11.302 adrv9361\_z7035\_signal\_source\_fpga.h File Reference

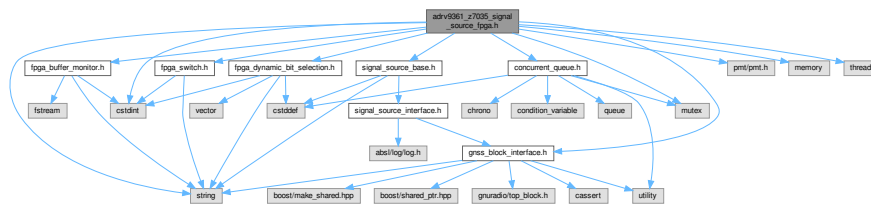
Signal source for the Analog Devices ADRV9361-Z7035 evaluation board directly connected to the FPGA accelerators. This source implements only the AD9361 control. It is NOT compatible with conventional SDR acquisition and tracking blocks. Please use the fmcomms2 source if conventional SDR acquisition and tracking is selected in the configuration file.

```

#include "concurrent_queue.h"
#include "fpga_buffer_monitor.h"
#include "fpga_dynamic_bit_selection.h"
#include "fpga_switch.h"
#include "gnss_block_interface.h"
#include "signal_source_base.h"
#include <pmt/pmt.h>
#include <stdint>
#include <memory>
#include <mutex>
#include <string>
#include <thread>

```

Include dependency graph for `adrv9361_z7035_signal_source_fpga.h`:



### Classes

- class [Adrv9361z7035SignalSourceFPGA](#)

### 11.302.1 Detailed Description

Signal source for the Analog Devices ADRV9361-Z7035 evaluation board directly connected to the FPGA accelerators. This source implements only the AD9361 control. It is NOT compatible with conventional SDR acquisition and tracking blocks. Please use the fmcomms2 source if conventional SDR acquisition and tracking is selected in the configuration file.

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [adrv9361\\_z7035\\_signal\\_source\\_fpga.h](#).



## 11.303 adrv9361\_z7035\_signal\_source\_fpga.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file adrv9361_z7035_signal_source_fpga.h
00003  * \brief Signal source for the Analog Devices ADRV9361-Z7035 evaluation board
00004  * directly connected to the FPGA accelerators.
00005  * This source implements only the AD9361 control. It is NOT compatible with
00006  * conventional SDR acquisition and tracking blocks.
00007  * Please use the fmcomms2 source if conventional SDR acquisition and tracking
00008  * is selected in the configuration file.
00009  *
00010  * -----
00011  *
00012  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00013  * This file is part of GNSS-SDR.
00014  *
00015  * Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors)
00016  * SPDX-License-Identifier: GPL-3.0-or-later
00017  *
00018  * -----
00019  */
00020
00021 #ifndef GNSS_SDR_ADRV9361_Z7035_SIGNAL_SOURCE_FPGA_H
00022 #define GNSS_SDR_ADRV9361_Z7035_SIGNAL_SOURCE_FPGA_H
00023
00024 #include "concurrent_queue.h"
00025 #include "fpga_buffer_monitor.h"
00026 #include "fpga_dynamic_bit_selection.h"
00027 #include "fpga_switch.h"
00028 #include "gnss_block_interface.h"
00029 #include "signal_source_base.h"
00030 #include <pmt/pmt.h>
00031 #include <cstdint>
00032 #include <memory>
00033 #include <mutex>
00034 #include <string>
00035 #include <thread>
00036
00037
00038 /** \addtogroup Signal_Source
00039  * \{ */
00040 /** \addtogroup Signal_Source_adapters
00041  * \{ */
00042
00043
00044 class ConfigurationInterface;
00045
00046 class ADRV9361Z7035SignalSourceFPGA : public SignalSourceBase
00047 {
00048 public:
00049     ADRV9361Z7035SignalSourceFPGA(const ConfigurationInterface *configuration,
00050         const std::string &role, unsigned int in_stream,
00051         unsigned int out_stream, Concurrent_Queue<pmt::pmt_t> *queue);
00052
00053     ~ADRV9361Z7035SignalSourceFPGA();
00054
00055     inline size_t item_size() override
00056     {
00057         return item_size_;
00058     }
00059
00060     void connect(gr::top_block_sptr top_block) override;
00061     void disconnect(gr::top_block_sptr top_block) override;
00062     gr::basic_block_sptr get_left_block() override;
00063     gr::basic_block_sptr get_right_block() override;
00064
00065 private:
00066     const std::string default_dump_filename = std::string("FPGA_buffer_monitor_dump.dat");
00067     const std::string default_rf_port_select = std::string("A_BALANCED");
00068     const std::string default_gain_mode = std::string("slow_attack");
00069     const double default_tx_attenuation_db = -10.0;
00070     const double default_manual_gain_rx1 = 64.0;
00071     const double default_manual_gain_rx2 = 64.0;
00072     const uint64_t default_bandwidth = 12500000;
00073
00074     // perform dynamic bit selection every 500 ms by default
00075     const uint32_t Gain_control_period_ms = 500;
00076     // check buffer overflow and perform buffer monitoring every 1s by default
00077     const uint32_t buffer_monitor_period_ms = 1000;
00078     // buffer overflow and buffer monitoring initial delay
00079     const uint32_t buffer_monitoring_initial_delay_ms = 2000;
00080     const int32_t switch_to_real_time_mode = 2;
00081
00082     void run_dynamic_bit_selection_process();
00083     void run_buffer_monitor_process();

```

```

00084
00085     mutable std::mutex dynamic_bit_selection_mutex;
00086     mutable std::mutex buffer_monitor_mutex;
00087
00088     std::thread thread_dynamic_bit_selection;
00089     std::thread thread_buffer_monitor;
00090
00091     std::shared_ptr<Fpga_Switch> switch_fpga;
00092     std::shared_ptr<Fpga_dynamic_bit_selection> dynamic_bit_selection_fpga;
00093     std::shared_ptr<Fpga_buffer_monitor> buffer_monitor_fpga;
00094
00095     std::string gain_mode_rx1_;
00096     std::string gain_mode_rx2_;
00097     std::string rf_port_select_;
00098     std::string filter_file_;
00099     std::string filter_source_;
00100     std::string filter_filename_;
00101
00102     double rf_gain_rx1_;
00103     double rf_gain_rx2_;
00104     double scale_dds_dbfs_;
00105     double phase_dds_deg_;
00106     double tx_attenuation_db_;
00107
00108     uint64_t freq0_; // frequency of local oscillator for ADRV9361-A 0
00109     uint64_t sample_rate_;
00110     uint64_t bandwidth_;
00111     uint64_t freq_dds_tx_hz_;
00112     uint64_t freq_rf_tx_hz_;
00113     uint64_t tx_bandwidth_;
00114
00115     float Fpass_;
00116     float Fstop_;
00117     uint32_t in_stream_;
00118     uint32_t out_stream_;
00119
00120     size_t item_size_;
00121
00122     bool enable_dds_lo_;
00123     bool quadrature_;
00124     bool rf_dc_;
00125     bool bb_dc_;
00126     bool rx1_enable_;
00127     bool rx2_enable_;
00128     bool enable_dynamic_bit_selection_;
00129     bool enable_ovf_check_buffer_monitor_active_;
00130     bool dump_;
00131     bool rf_shutdown_;
00132 };
00133
00134
00135 /** \} */
00136 /** \} */
00137 #endif // GNSS_SDR_ADRV9361_Z7035_SIGNAL_SOURCE_FPGA_H

```

## 11.304 custom\_udp\_signal\_source.h File Reference

Receives ip frames containing samples in UDP frame encapsulation using a high performance packet capture library (libpcap)

```

#include "concurrent_queue.h"
#include "gr_complex_ip_packet_source.h"
#include "signal_source_base.h"
#include <gnuradio/blocks/file_sink.h>
#include <gnuradio/blocks/null_sink.h>
#include <pmt/pmt.h>
#include <stdexcept>
#include <string>
#include <vector>

```

[illegible]

- class CustomUDPSignalSource

### 11.304.1 Detailed Description

**Author**

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [custom\\_udp\\_signal\\_source.h](#).

[Go to the documentation of this file.](#)

Generated by Doxygen

```

00041  * I/Q samples over a network.
00042  */
00043  class CustomUDPSignalSource : public SignalSourceBase
00044  {
00045  public:
00046      CustomUDPSignalSource(const ConfigurationInterface* configuration,
00047                          const std::string& role, unsigned int in_stream,
00048                          unsigned int out_stream, Concurrent_Queue<pmt::pmt_t*> queue);
00049
00050      ~CustomUDPSignalSource() = default;
00051
00052      inline size_t item_size() override
00053      {
00054          return item_size_;
00055      }
00056
00057      void connect(gr::top_block_sptr top_block) override;
00058      void disconnect(gr::top_block_sptr top_block) override;
00059      gr::basic_block_sptr get_left_block() override;
00060      gr::basic_block_sptr get_right_block() override;
00061      gr::basic_block_sptr get_right_block(int RF_channel) override;
00062
00063  private:
00064      Gr_Complex_Ip_Packet_Source::sptr udp_gnss_rx_source_;
00065      std::vector<gnss_shared_ptr<gr::block>> null_sinks_;
00066      std::vector<gnss_shared_ptr<gr::block>> file_sink_;
00067
00068      std::string item_type_;
00069      std::string dump_filename_;
00070
00071      size_t item_size_;
00072
00073      int RF_channels_;
00074      int channels_in_udp_;
00075      unsigned int in_stream_;
00076      unsigned int out_stream_;
00077      bool IQ_swap_;
00078      bool dump_;
00079  };
00080
00081
00082  /** \} */
00083  /** \} */
00084  #endif // GNSS_SDR_CUSTOM_UDP_SIGNAL_SOURCE_H

```

## 11.306 dma\_signal\_source\_fpga.h File Reference

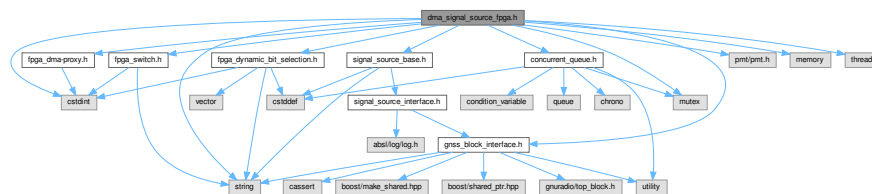
signal source for a DMA connected directly to FPGA accelerators. This source implements only the DMA control. It is NOT compatible with conventional SDR acquisition and tracking blocks.

```

#include "concurrent_queue.h"
#include "fpga_dma-proxy.h"
#include "fpga_dynamic_bit_selection.h"
#include "fpga_switch.h"
#include "gnss_block_interface.h"
#include "signal_source_base.h"
#include <pmt/pmt.h>
#include <stdint>
#include <memory>
#include <mutex>
#include <string>
#include <thread>

```

Include dependency graph for dma\_signal\_source\_fpga.h:



**Classes**

- class [DMASignalSourceFPGA](#)

**11.306.1 Detailed Description**

signal source for a DMA connected directly to FPGA accelerators. This source implements only the DMA control. It is NOT compatible with conventional SDR acquisition and tracking blocks.

**Author**

Marc Majoral, mmajoral(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [dma\\_signal\\_source\\_fpga.h](#).

---

**11.307 dma\_signal\_source\_fpga.h**

[Go to the documentation of this file.](#)

```

00001  /*!
00002   * \file dma_signal_source_fpga.h
00003   * \brief signal source for a DMA connected directly to FPGA accelerators.
00004   * This source implements only the DMA control. It is NOT compatible with
00005   * conventional SDR acquisition and tracking blocks.
00006   * \author Marc Majoral, mmajoral(at)cttc.es
00007   *
00008   * -----
00009   *
00010   * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011   * This file is part of GNSS-SDR.
00012   *
00013   * Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors)
00014   * SPDX-License-Identifier: GPL-3.0-or-later
00015   *
00016   * -----
00017   */
00018
00019 #ifndef GNSS_SDR_DMA_SIGNAL_SOURCE_FPGA_H
00020 #define GNSS_SDR_DMA_SIGNAL_SOURCE_FPGA_H
00021
00022 #include "concurrent_queue.h"
00023 #include "fpga_dma-proxy.h"
00024 #include "fpga_dynamic_bit_selection.h"
00025 #include "fpga_switch.h"
00026 #include "gnss_block_interface.h"
00027 #include "signal_source_base.h"
00028 #include <pmt/pmt.h>
00029 #include <cstdint>
00030 #include <memory>
00031 #include <mutex>
00032 #include <string>
00033 #include <thread>
00034
00035
00036 /** \addtogroup Signal_Source
00037  * \{ */
00038 /** \addtogroup Signal_Source_adapters
00039  * \{ */
00040
00041
00042 class ConfigurationInterface;
00043
00044 class DMASignalSourceFPGA : public SignalSourceBase
00045 {
00046 public:
00047     DMASignalSourceFPGA(const ConfigurationInterface *configuration,
00048         const std::string &role, unsigned int in_stream,
00049         unsigned int out_stream, Concurrent_Queue<pmt::pmt_t> *queue);
00050
00051     ~DMASignalSourceFPGA();
00052
00053     void start() override;
00054
00055     inline size_t item_size() override
00056     {
00057         return item_size_;
00058     }
00059

```

```

00060     void connect(gr::top_block_sptr top_block) override;
00061     void disconnect(gr::top_block_sptr top_block) override;
00062     gr::basic_block_sptr get_left_block() override;
00063     gr::basic_block_sptr get_right_block() override;
00064
00065 private:
00066     const std::string dyn_bit_sel_device_name = std::string("dynamic_bits_selector"); // Switch
00067     dhnamc bit selector device name
00067     const std::string empty_string;
00068     const uint64_t default_bandwidth = 12500000;
00069     // perform dynamic bit selection every 500 ms by default
00070     const uint32_t Gain_control_period_ms = 500;
00071     // sample block size when running in post-processing mode
00072     const int sample_block_size = 16384;
00073     const int32_t switch_to_DMA = 0;
00074
00075     void run_DMA_process(const std::string &filename0,
00076                         const std::string &filename1,
00077                         uint64_t &samples_to_skip,
00078                         size_t &item_size,
00079                         int64_t &samples,
00080                         bool &repeat,
00081                         uint32_t &dma_buff_offset_pos,
00082                         Concurrent_Queue<pmt::pmt_t> *queue);
00083
00084     void run_dynamic_bit_selection_process();
00085
00086     std::thread thread_file_to_dma;
00087     std::thread thread_dynamic_bit_selection;
00088
00089     std::shared_ptr<Fpga_Switch> switch_fpga;
00090     std::shared_ptr<Fpga_dynamic_bit_selection> dynamic_bit_selection_fpga;
00091     std::shared_ptr<Fpga_DMA> dma_fpga;
00092
00093     std::mutex dma_mutex;
00094     std::mutex dynamic_bit_selection_mutex;
00095
00096     Concurrent_Queue<pmt::pmt_t> *queue_;
00097
00098     std::string filename0_;
00099     std::string filename1_;
00100
00101     uint64_t sample_rate_;
00102     uint64_t samples_to_skip_;
00103     int64_t samples_;
00104     uint32_t num_input_files_;
00105     uint32_t dma_buff_offset_pos_;
00106     uint32_t in_stream_;
00107     uint32_t out_stream_;
00108     size_t item_size_;
00109
00110     bool enable_DMA_;
00111     bool enable_dynamic_bit_selection_;
00112     bool repeat_;
00113 };
00114
00115
00116 /** \} */
00117 /** \} */
00118 #endif // GNSS_SDR_DMA_SIGNAL_SOURCE_FPGA_H

```

## 11.308 fifo\_signal\_source.h File Reference

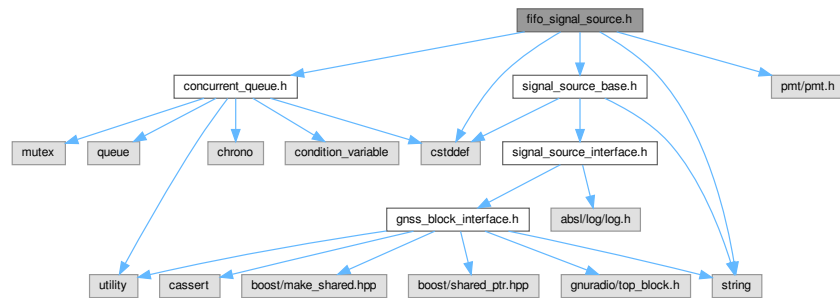
Header file of the class for retrieving samples through a Unix FIFO.

```

#include "concurrent_queue.h"
#include "signal_source_base.h"
#include <pmt/pmt.h>
#include <cstdint>
#include <string>

```

Include dependency graph for fifo\_signal\_source.h:



## Classes

- class [FifoSignalSource](#)

*Class that reads a sample stream from a Unix FIFO.*

### 11.308.1 Detailed Description

Header file of the class for retrieving samples through a Unix FIFO.

#### Author

Malte Lenhart, 2021. [malte.lenhart\(at\)mailbox.org](mailto:malte.lenhart(at)mailbox.org)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [fifo\\_signal\\_source.h](#).

---

## 11.309 fifo\_signal\_source.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file fifo_signal_source.h
00003  *
00004  * \brief Header file of the class for retrieving samples through a Unix FIFO
00005  * \author Malte Lenhart, 2021. malte.lenhart(at)mailbox.org
00006  *
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019 #ifndef GNSS_SDR_FIFO_SIGNAL_SOURCE_H
00020 #define GNSS_SDR_FIFO_SIGNAL_SOURCE_H
00021
00022 #include "concurrent_queue.h"
00023 #include "signal_source_base.h"
00024 #include <pmt/pmt.h>
00025 #include <cstdint>
00026 #include <string>
00027
00028 /** \addtogroup Signal_Source
00029  * \{ */
00030 /** \addtogroup Signal_Source_adapters
00031  * \{ */
00032
00033 // forward declaration to avoid include in header
00034 class ConfigurationInterface;

```

```

00035
00036 /// \brief Class that reads a sample stream from a Unix FIFO.
00037 ///
00038 /// This class supports the following properties:
00039 ///
00040 /// .filename - the path to the input file
00041 ///           - may be overridden by the -signal_source or -s command-line arguments
00042 ///
00043 /// .sample_type - data type read out from the FIFO. default is short ;
00044 ///               - note: not output format. that is always gr_complex
00045 ///
00046 /// .dump - whether to archive input data
00047 ///
00048 /// .dump_filename - if dumping, path to file for output
00049 ///
00050 class FifoSignalSource : public SignalSourceBase
00051 {
00052 public:
00053     FifoSignalSource(const ConfigurationInterface* configuration, const std::string& role,
00054                      unsigned int in_streams, unsigned int out_streams,
00055                      Concurrent_Queue<pmt::pmt_t>* queue);
00056 
00057     ~FifoSignalSource() = default;
00058 
00059     /// override methods from GNSSBlockInterface
00060     void connect(gr::top_block_sptr top_block) override;
00061     void disconnect(gr::top_block_sptr top_block) override;
00062     size_t item_size() override;
00063     gr::basic_block_sptr get_left_block() override;
00064     gr::basic_block_sptr get_right_block() override;
00065 
00066 protected:
00067 private:
00068     /// output size - always gr_complex
00069     const size_t item_size_;
00070     /// internal fifo_reader_ class acts as signal source
00071     const gnss_shared_ptr<gr::block> fifo_reader_;
00072 
00073     gnss_shared_ptr<gr::block> file_sink_;
00074     const bool dump_;
00075     const std::string dump_filename_;
00076 };
00077 
00078 /** \} */
00079 /** \} */
00080 #endif // GNSS_SDR_FIFO_SIGNAL_SOURCE_H

```

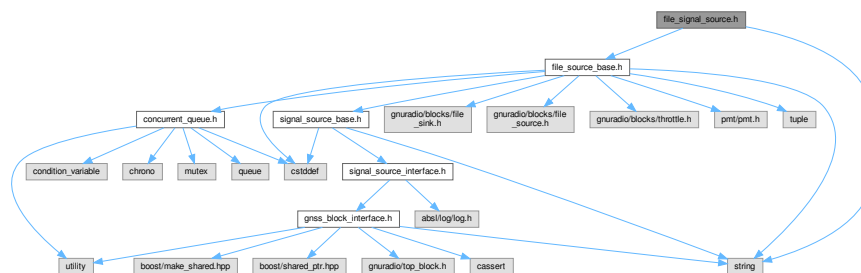
## 11.310 file\_signal\_source.h File Reference

Interface of a class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).

```
#include "file_source_base.h"
```

```
#include <string>
```

Include dependency graph for file\_signal\_source.h:



### Classes

- class [FileSignalSource](#)

*Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).*



### 11.310.1 Detailed Description

Interface of a class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).

#### Author

Carlos Aviles, 2010. carlos.avilesr(at)gmail.com

This class represents a file signal source. Internally it uses a GNU Radio's gr\_file\_source as a connector to the data.

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [file\\_signal\\_source.h](#).

## 11.311 file\_signal\_source.h

[Go to the documentation of this file.](#)

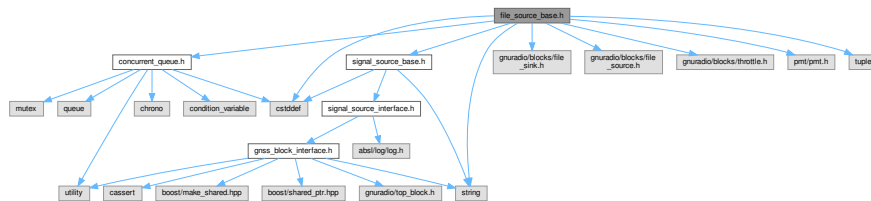
```
00001 /*!
00002  * \file file_signal_source.h
00003  * \brief Interface of a class that reads signals samples from a file
00004  * and adapts it to a SignalSourceInterface
00005  * \author Carlos Aviles, 2010. carlos.avilesr(at)gmail.com
00006  *
00007  * This class represents a file signal source. Internally it uses a GNU Radio's
00008  * gr_file_source as a connector to the data.
00009  *
00010  * -----
00011  *
00012  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00013  * This file is part of GNSS-SDR.
00014  *
00015  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00016  * SPDX-License-Identifier: GPL-3.0-or-later
00017  *
00018  * -----
00019  */
00020
00021 #ifndef GNSS_SDR_FILE_SIGNAL_SOURCE_H
00022 #define GNSS_SDR_FILE_SIGNAL_SOURCE_H
00023
00024 #include "file_source_base.h"
00025 #include <string>
00026
00027 /** \addtogroup Signal_Source Signal Source
00028  * Classes for Signal Source management.
00029  * \{ */
00030 /** \addtogroup Signal_Source_adapters signal_source_adapters
00031  * Classes that wrap GNU Radio signal sources with a GNSSBlockInterface
00032  * \{ */
00033
00034
00035 class ConfigurationInterface;
00036
00037 /*!
00038  * \brief Class that reads signals samples from a file
00039  * and adapts it to a SignalSourceInterface
00040  */
00041 class FileSignalSource : public FileSourceBase
00042 {
00043 public:
00044     FileSignalSource(ConfigurationInterface const* configuration, std::string const& role,
00045                     unsigned int in_streams, unsigned int out_streams,
00046                     Concurrent_Queue<pmt::pmt_t>* queue);
00047     ~FileSignalSource() = default;
00048
00049 private:
00050 };
00051
00052
00053
00054 /** \} */
00055 /** \} */
00056 #endif // GNSS_SDR_FILE_SIGNAL_SOURCE_H
```

## 11.312 file\_source\_base.h File Reference

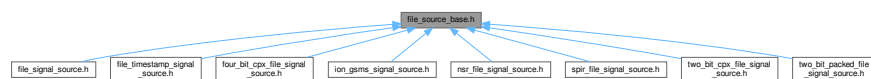
Header file of the base class to file-oriented signal\_source GNSS blocks.

```
#include "concurrent_queue.h"
#include "signal_source_base.h"
#include <gnuradio/blocks/file_sink.h>
#include <gnuradio/blocks/file_source.h>
#include <gnuradio/blocks/throttle.h>
#include <pmt/pmt.h>
#include <cstdint>
#include <string>
#include <tuple>
```

Include dependency graph for file\_source\_base.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [FileSourceBase](#)  
Base class to file-oriented [SignalSourceBase](#) GNSS blocks.

### 11.312.1 Detailed Description

Header file of the base class to file-oriented signal\_source GNSS blocks.

#### Author

Jim Melton, 2021. [jim.melton@snrcorp.com](mailto:jim.melton@snrcorp.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [file\\_source\\_base.h](#).

---

### 11.313 file\_source\_base.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file file_source_base.h
00003  * \brief Header file of the base class to file-oriented signal_source GNSS blocks.
00004  * \author Jim Melton, 2021. jim.melton@snrcorp.com
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
```

```

00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_FILE_SOURCE_BASE_H
00019 #define GNSS_SDR_FILE_SOURCE_BASE_H
00020
00021 #include "concurrent_queue.h"
00022 #include "signal_source_base.h"
00023 #include <gnuradio/blocks/file_sink.h> // for dump
00024 #include <gnuradio/blocks/file_source.h>
00025 #include <gnuradio/blocks/throttle.h>
00026 #include <pmt/pmt.h>
00027 #include <cstdint>
00028 #include <string>
00029 #include <tuple>
00030
00031 /** \addtogroup Signal_Source
00032  * \{ */
00033 /** \addtogroup Signal_Source_adapters
00034  * \{ */
00035
00036 class ConfigurationInterface;
00037
00038
00039 /// \brief Base class to file-oriented SignalSourceBase GNSS blocks.
00040 ///
00041 /// This class supports the following properties:
00042 ///
00043 /// .filename - the path to the input file
00044 ///             - may be overridden by the -signal_source or -s command-line arguments
00045 ///
00046 /// .samples - number of samples to process (default 0)
00047 ///             - if not specified or 0, read the entire file; otherwise stop after that many samples
00048 ///
00049 /// .sampling_frequency - the frequency of the sampled data (samples/second)
00050 ///
00051 /// .item_type - data type of the samples (default "short")
00052 ///
00053 /// .header_size - the size of a prefixed header to skip in "samples" (default 0)
00054 ///
00055 /// .seconds_to_skip - number of seconds of lead-in data to skip over (default 0)
00056 ///
00057 /// .enable_throttle_control - whether to stop reading if the upstream buffer is full (default
00058 ///                             false)
00059 ///
00060 /// .repeat - whether to rewind and continue at end of file (default false)
00061 ///
00062 /// (probably abstracted to the base class)
00063 ///
00064 /// .dump - whether to archive input data
00065 ///
00066 /// .dump_filename - if dumping, path to file for output
00067 class FileSourceBase : public SignalSourceBase
00068 {
00069 public:
00070     // Virtual overrides
00071     void connect(gr::top_block_sptr top_block) override;
00072     void disconnect(gr::top_block_sptr top_block) override;
00073     gr::basic_block_sptr get_left_block() override;
00074     gr::basic_block_sptr get_right_block() override;
00075
00076     /// The file to read
00077     std::string filename() const;
00078
00079     /// The item type
00080     std::string item_type() const;
00081
00082     /// The configured size of each item
00083     size_t item_size() override;
00084     virtual size_t item_size() const; // what the interface **should** have declared
00085
00086     /// Whether to repeat reading after end-of-file
00087     bool repeat() const;
00088
00089     /// The sampling frequency of the source file
00090     int64_t sampling_frequency() const;
00091
00092     /// The number of samples in the file
00093     uint64_t samples() const;
00094
00095 protected:
00096     /// \brief Constructor
00097     ///
00098     /// Subclasses may want to assert default item types that are appropriate to the specific file
00099     /// type supported. Rather than require the item type to be specified in the config file, allow
00100     /// sub-classes to impose their will
00101     FileSourceBase(ConfigurationInterface const* configuration, std::string const& role, std::string

```

```

impl,
00101     Concurrent_Queue<pmt::pmt_t>* queue,
00102     std::string default_item_type = "short");
00103
00104     ///! Perform post-construction initialization
00105     void init();
00106
00107     ///! Compute the item size, from the item_type(). Subclasses may constrain types that don't make
00108     ///! sense. The return of this method is a tuple of item_size and is_complex
00109     virtual std::tuple<size_t, bool> itemTypeToSize();
00110
00111     ///! The number of (possibly unpacked) samples in a (raw) file sample (default=1)
00112     virtual double packetsPerSample() const;
00113
00114     ///! Compute the number of samples to skip
00115     virtual size_t samplesToSkip() const;
00116
00117     ///! Compute the number of samples in the file
00118     size_t computeSamplesInFile() const;
00119
00120     ///! Abstracted front-end source. Sub-classes may override if they create specialized chains to
00121     ///! decode source files into a usable format
00122     virtual gnss_shared_ptr<gr::block> source() const;
00123
00124     ///! For complex source chains, the size of the file item may not be the same as the size of the
00125     ///! "source" (decoded) item. This method allows subclasses to handle these differences
00126     virtual size_t source_item_size() const;
00127     bool is_complex() const;
00128
00129     /// Generic access to created objects
00130     gnss_shared_ptr<gr::block> file_source() const;
00131     gnss_shared_ptr<gr::block> valve() const;
00132     gnss_shared_ptr<gr::block> throttle() const;
00133     gnss_shared_ptr<gr::block> sink() const;
00134
00135     /// The methods create the various blocks, if enabled, and return access to them. The created
00136     /// object is also held in this class
00137     gr::blocks::file_source::sptr create_file_source();
00138     gr::blocks::throttle::sptr create_throttle();
00139     gnss_shared_ptr<gr::block> create_valve();
00140     gr::blocks::file_sink::sptr create_sink();
00141
00142     /// Subclass hooks to augment created objects, as required
00143     virtual void create_file_source_hook();
00144     virtual void create_throttle_hook();
00145     virtual void create_valve_hook();
00146     virtual void create_sink_hook();
00147
00148     /// Subclass hooks for connection/disconnection
00149     virtual void pre_connect_hook(gr::top_block_sptr top_block);
00150     virtual void post_connect_hook(gr::top_block_sptr top_block);
00151     virtual void pre_disconnect_hook(gr::top_block_sptr top_block);
00152     virtual void post_disconnect_hook(gr::top_block_sptr top_block);
00153
00154 private:
00155     gr::blocks::file_source::sptr file_source_;
00156     gr::blocks::throttle::sptr throttle_;
00157     gr::blocks::file_sink::sptr sink_;
00158
00159     /// The valve allows only the configured number of samples through, then it closes.
00160
00161     /// The framework passes the queue as a naked pointer, rather than a shared pointer, so this
00162     /// class has two choices: construct the valve in the ctor, or hold onto the pointer, possibly
00163     /// beyond its lifetime. Fortunately, the queue is only used to create the valve, so the
00164     /// likelihood of holding a stale pointer is mitigated
00165     gnss_shared_ptr<gr::block> valve_;
00166     Concurrent_Queue<pmt::pmt_t>* queue_;
00167
00168     std::string role_;
00169     std::string filename_;
00170     std::string dump_filename_;
00171     std::string item_type_;
00172     size_t item_size_;
00173     size_t header_size_; // length (in samples) of the header (if any)
00174     uint64_t samples_;
00175     int64_t sampling_frequency_; // why is this signed
00176     double minimum_tail_s_;
00177     double seconds_to_skip_;
00178     bool is_complex_; // a misnomer; if I/Q are interleaved as integer values
00179     bool repeat_;
00180     bool enable_throttle_control_;
00181     bool dump_;
00182 };
00183
00184 /** @} */
00185 /** @} */
00186 #endif // GNSS_SDR_FILE_SOURCE_BASE_H

```





### 11.316.1 Detailed Description

Signal Source adapter for the Teleorbit Flexiband front-end device. This adapter requires a Flexiband GNU Radio driver installed (not included with GNSS-SDR)

#### Author

Javier Arribas, jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [flexiband\\_signal\\_source.h](#).

---

## 11.317 flexiband\_signal\_source.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file flexiband_signal_source.h
00003  * \brief Signal Source adapter for the Teleorbit Flexiband front-end device.
00004  * This adapter requires a Flexiband GNU Radio driver
00005  * installed (not included with GNSS-SDR)
00006  * \author Javier Arribas, jarribas(at)cttc.es
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019
00020 #ifndef GNSS_SDR_FLEXIBAND_SIGNAL_SOURCE_H
00021 #define GNSS_SDR_FLEXIBAND_SIGNAL_SOURCE_H
00022
00023 #include "concurrent_queue.h"
00024 #include "signal_source_base.h"
00025 #include <gnuradio/blocks/char_to_float.h>
00026 #include <gnuradio/blocks/file_sink.h>
00027 #include <gnuradio/blocks/float_to_complex.h>
00028 #include <gnuradio/blocks/null_sink.h>
00029 #include <gnuradio/hier_block2.h>
00030 #include <pmt/pmt.h>
00031 #include <memory>
00032 #include <string>
00033 #include <vector>
00034
00035
00036 /** \addtogroup Signal_Source
00037  * \{ */
00038 /** \addtogroup Signal_Source_adapters
00039  * \{ */
00040
00041
00042 class ConfigurationInterface;
00043
00044 /*!
00045  * \brief This class configures and reads samples from Teleorbit Flexiband front-end.
00046  * This software requires a Flexiband GNU Radio driver installed (not included with GNSS-SDR).
00047  */
00048 class FlexibandSignalSource : public SignalSourceBase
00049 {
00050 public:
00051     FlexibandSignalSource(const ConfigurationInterface* configuration,
00052                          const std::string& role, unsigned int in_stream,
00053                          unsigned int out_stream, Concurrent_Queue<pmt::pmt_t>* queue);
00054
00055     ~FlexibandSignalSource() = default;
00056
00057     inline size_t item_size() override
00058     {
00059         return item_size_;
00060     }
00061
00062     void connect(gr::top_block_sptr top_block) override;
00063     void disconnect(gr::top_block_sptr top_block) override;
00064     gr::basic_block_sptr get_left_block() override;
00065     gr::basic_block_sptr get_right_block() override;

```

```

00066     gr::basic_block_sptr get_right_block(int RF_channel) override;
00067
00068 private:
00069     gnss_shared_ptr<gr::block> flexiband_source_;
00070
00071     std::vector<gnss_shared_ptr<gr::block>> char_to_float_;
00072     std::vector<gnss_shared_ptr<gr::block>> float_to_complex_;
00073     std::vector<gr::blocks::null_sink::sptr> null_sinks_;
00074
00075     std::string item_type_;
00076     std::string firmware_filename_;
00077     std::string signal_file_;
00078
00079     size_t item_size_;
00080     unsigned int in_stream_;
00081     unsigned int out_stream_;
00082
00083     int gain1_;
00084     int gain2_;
00085     int gain3_;
00086     int usb_packet_buffer_size_;
00087     int n_channels_;
00088     int sel_ch_;
00089
00090     bool AGC_;
00091     bool flag_read_file_;
00092 };
00093
00094
00095 /** \} */
00096 /** \} */
00097 #endif // GNSS_SDR_FLEXIBAND_SIGNAL_SOURCE_H

```

## 11.318 fmcomms2\_signal\_source.h File Reference

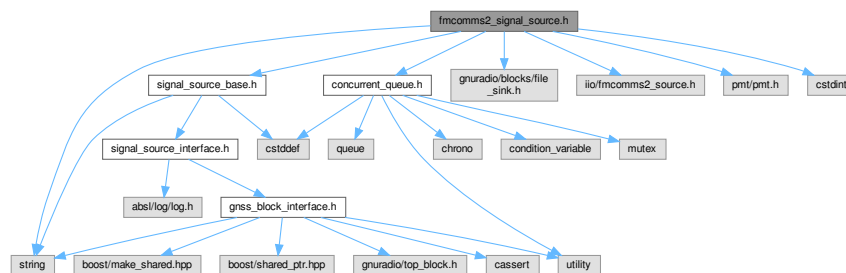
Interface to use SDR hardware based in FCOMMS2 driver from analog devices, for example FCOMMS4 and ADALM-PLUTO (PlutoSdr)

```

#include "signal_source_base.h"
#include <gnuradio/blocks/file_sink.h>
#include <iio/fmcomms2_source.h>
#include "concurrent_queue.h"
#include <pmt/pmt.h>
#include <cstdint>
#include <string>

```

Include dependency graph for fmcomms2\_signal\_source.h:



### Classes

- class [Fmcomms2SignalSource](#)

### 11.318.1 Detailed Description

Interface to use SDR hardware based in FCOMMS2 driver from analog devices, for example FCOMMS4 and ADALM-PLUTO (PlutoSdr)



## Author

Rodrigo Muñoz, 2017. rmunozl(at)inacap.cl, rodrigo.munoz(at)proteinlab.cl

### 11.318.1.1 This class represent a fmcomms2 signal source. It use the gr\_iio block

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [fmcomms2\\_signal\\_source.h](#).

## 11.319 fmcomms2\_signal\_source.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file fmcomms2_signal_source.h
00003  * \brief Interface to use SDR hardware based in FMCOMMS2 driver from analog
00004  * devices, for example FMCOMMS4 and ADALM-PLUTO (PlutoSdr)
00005  * \author Rodrigo Muñoz, 2017. rmunozl(at)inacap.cl, rodrigo.munoz(at)proteinlab.cl
00006  *
00007  *
00008  * This class represent a fmcomms2 signal source. It use the gr_iio block
00009  * -----
00010  *
00011  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00012  * This file is part of GNSS-SDR.
00013  *
00014  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00015  * SPDX-License-Identifier: GPL-3.0-or-later
00016  *
00017  * -----
00018  */
00019
00020 #ifndef GNSS_SDR_FCOMMS2_SIGNAL_SOURCE_H
00021 #define GNSS_SDR_FCOMMS2_SIGNAL_SOURCE_H
00022
00023 #include "signal_source_base.h"
00024 #include <gnuradio/blocks/file_sink.h>
00025 #if GRIIO_INCLUDE_HAS_GNURADIO
00026 #include <gnuradio/iio/fmcomms2_source.h>
00027 #else
00028 #include <iio/fmcomms2_source.h>
00029 #endif
00030 #include "concurrent_queue.h"
00031 #include <pmt/pmt.h>
00032 #include <cstdint>
00033 #include <string>
00034
00035
00036 /** \addtogroup Signal_Source
00037  * \{ */
00038 /** \addtogroup Signal_Source_adapters
00039  * \{ */
00040
00041
00042 class ConfigurationInterface;
00043
00044 class Fmcomms2SignalSource : public SignalSourceBase
00045 {
00046 public:
00047     Fmcomms2SignalSource(const ConfigurationInterface* configuration,
00048         const std::string& role, unsigned int in_stream,
00049         unsigned int out_stream, Concurrent_Queue<pmt::pmt_t>* queue);
00050
00051     ~Fmcomms2SignalSource();
00052
00053     inline size_t item_size() override
00054     {
00055         return item_size_;
00056     }
00057
00058     void connect(gr::top_block_sptr top_block) override;
00059     void disconnect(gr::top_block_sptr top_block) override;
00060     gr::basic_block_sptr get_left_block() override;
00061     gr::basic_block_sptr get_right_block() override;
00062
00063 private:
00064     const std::string default_gain_mode = std::string("slow_attack");
00065     const double default_tx_attenuation_db = -10.0;
00066
00067 #if GNURADIO_API_IIO
00068 #if GR_IIO_TEMPLATIZED_API

```

```

00069     gr::io::fmcomms2_source<gr_complex>::sptr fmcomms2_source_f32c_;
00070 #else
00071     gr::io::fmcomms2_source::sptr fmcomms2_source_f32c_;
00072 #endif
00073 #else
00074     gr::io::fmcomms2_source_f32c::sptr fmcomms2_source_f32c_;
00075 #endif
00076     gnss_shared_ptr<gr::block> valve_;
00077     gr::blocks::file_sink::sptr file_sink_;
00078
00079     std::string item_type_;
00080     std::string dump_filename_;
00081
00082     // Front-end settings
00083     std::string uri_; // device direction
00084     std::string gain_mode_rx1_;
00085     std::string gain_mode_rx2_;
00086     std::string rf_port_select_;
00087     std::string filter_file_;
00088     std::string filter_source_;
00089     std::string filter_filename_;
00090
00091     int64_t samples_;
00092     size_t item_size_;
00093
00094     double rf_gain_rx1_;
00095     double rf_gain_rx2_;
00096     uint64_t freq_; // frequency of local oscillator
00097     uint64_t sample_rate_;
00098     uint64_t bandwidth_;
00099     uint64_t buffer_size_; // reception buffer
00100     float Fpass_;
00101     float Fstop_;
00102     unsigned int in_stream_;
00103     unsigned int out_stream_;
00104     int RF_channels_;
00105
00106     // DDS configuration for LO generation for external mixer
00107     double scale_dds_dbfs_;
00108     double phase_dds_deg_;
00109     double tx_attenuation_db_;
00110     uint64_t freq_dds_tx_hz_;
00111     uint64_t freq_rf_tx_hz_;
00112     uint64_t tx_bandwidth_;
00113     bool enable_dds_lo_;
00114
00115     bool rx1_en_;
00116     bool rx2_en_;
00117     bool quadrature_;
00118     bool rf_dc_;
00119     bool bb_dc_;
00120     bool filter_auto_;
00121     bool rf_shutdown_;
00122     bool dump_;
00123 };
00124
00125
00126 /** \} */
00127 /** \} */
00128 #endif // GNSS_SDR_FMCOMMS2_SIGNAL_SOURCE_H

```

## 11.320 fmcomms5\_signal\_source\_fpga.h File Reference

Signal source for the Analog Devices FMCOMMS5 directly connected to the FPGA accelerators. This source implements only the AD9361 control. It is NOT compatible with conventional SDR acquisition and tracking blocks. Please use the fmcomms2 source if conventional SDR acquisition and tracking is selected in the configuration file.

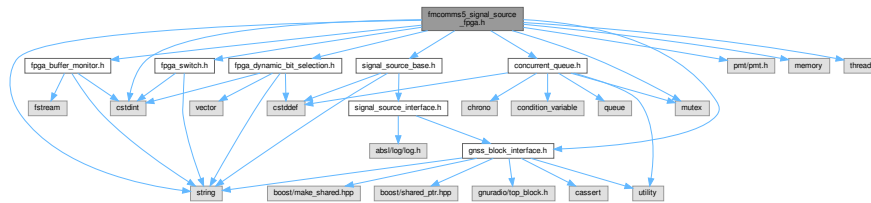
```

#include "concurrent_queue.h"
#include "fpga_buffer_monitor.h"
#include "fpga_dynamic_bit_selection.h"
#include "fpga_switch.h"
#include "gnss_block_interface.h"
#include "signal_source_base.h"
#include <pmt/pmt.h>
#include <cstdint>
#include <memory>
#include <mutex>
#include <string>

```

```
#include <thread>
```

Include dependency graph for fmcomms5\_signal\_source\_fpga.h:



## Classes

- class [Fmcomms5SignalSourceFPGA](#)

### 11.320.1 Detailed Description

Signal source for the Analog Devices FMCOMMS5 directly connected to the FPGA accelerators. This source implements only the AD9361 control. It is NOT compatible with conventional SDR acquisition and tracking blocks. Please use the fmcomms2 source if conventional SDR acquisition and tracking is selected in the configuration file. GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [fmcomms5\\_signal\\_source\\_fpga.h](#).

## 11.321 fmcomms5\_signal\_source\_fpga.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file fmcomms5_signal_source_fpga.h
00003  * \brief Signal source for the Analog Devices FMCOMMS5 directly connected
00004  * to the FPGA accelerators.
00005  * This source implements only the AD9361 control. It is NOT compatible with
00006  * conventional SDR acquisition and tracking blocks.
00007  * Please use the fmcomms2 source if conventional SDR acquisition and tracking
00008  * is selected in the configuration file.
00009  *
00010  * -----
00011  *
00012  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00013  * This file is part of GNSS-SDR.
00014  *
00015  * Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors)
00016  * SPDX-License-Identifier: GPL-3.0-or-later
00017  *
00018  * -----
00019  */
00020
00021 #ifndef GNSS_SDR_FMCOMMS5_SIGNAL_SOURCE_FPGA_H
00022 #define GNSS_SDR_FMCOMMS5_SIGNAL_SOURCE_FPGA_H
00023
00024 #include "concurrent_queue.h"
00025 #include "fpga_buffer_monitor.h"
00026 #include "fpga_dynamic_bit_selection.h"
00027 #include "fpga_switch.h"
00028 #include "gnss_block_interface.h"
00029 #include "signal_source_base.h"
00030 #include <pmt/pmt.h>
00031 #include <cstdint>
00032 #include <memory>
00033 #include <mutex>
00034 #include <string>
00035 #include <thread>
00036
00037
00038 /** \addtogroup Signal_Source
00039  * \{ */
00040 /** \addtogroup Signal_Source_adapters
00041  * \{ */
00042
00043
```

```

00044 class ConfigurationInterface;
00045
00046 class Fmcomms5SignalSourceFPGA : public SignalSourceBase
00047 {
00048 public:
00049     Fmcomms5SignalSourceFPGA(const ConfigurationInterface *configuration,
00050         const std::string &role, unsigned int in_stream,
00051         unsigned int out_stream, Concurrent_Queue<pmt::pmt_t> *queue);
00052
00053     ~Fmcomms5SignalSourceFPGA();
00054
00055     inline size_t item_size() override
00056     {
00057         return item_size_;
00058     }
00059
00060     void connect(gr::top_block_sptr top_block) override;
00061     void disconnect(gr::top_block_sptr top_block) override;
00062     gr::basic_block_sptr get_left_block() override;
00063     gr::basic_block_sptr get_right_block() override;
00064
00065 private:
00066     const std::string default_dump_filename = std::string("FPGA_buffer_monitor_dump.dat");
00067     const std::string default_rf_port_select = std::string("A_BALANCED");
00068     const std::string default_gain_mode = std::string("slow_attack");
00069     const double default_manual_gain_rx1 = 64.0;
00070     const double default_manual_gain_rx2 = 64.0;
00071     const uint64_t default_bandwidth = 12500000;
00072
00073     // perform dynamic bit selection every 500 ms by default
00074     const uint32_t Gain_control_period_ms = 500;
00075     // check buffer overflow and perform buffer monitoring every 1s by default
00076     const uint32_t buffer_monitor_period_ms = 1000;
00077     // buffer overflow and buffer monitoring initial delay
00078     const uint32_t buffer_monitoring_initial_delay_ms = 2000;
00079     const int32_t switch_to_real_time_mode = 2;
00080
00081     void run_dynamic_bit_selection_process();
00082     void run_buffer_monitor_process();
00083
00084     mutable std::mutex dynamic_bit_selection_mutex;
00085     mutable std::mutex buffer_monitor_mutex;
00086
00087     std::thread thread_dynamic_bit_selection;
00088     std::thread thread_buffer_monitor;
00089
00090     std::shared_ptr<Fpga_Switch> switch_fpga;
00091     std::shared_ptr<Fpga_dynamic_bit_selection> dynamic_bit_selection_fpga;
00092     std::shared_ptr<Fpga_buffer_monitor> buffer_monitor_fpga;
00093
00094     std::string gain_mode_rx1_;
00095     std::string gain_mode_rx2_;
00096     std::string rf_port_select_;
00097     std::string filter_file_;
00098     std::string filter_source_;
00099     std::string filter_filename_;
00100
00101     double rf_gain_rx1_;
00102     double rf_gain_rx2_;
00103
00104     uint64_t freq0_; // frequency of local oscillator for ADRV9361-A
00105     uint64_t freq1_; // frequency of local oscillator for ADRV9361-B
00106     uint64_t sample_rate_;
00107     uint64_t bandwidth_;
00108
00109     float Fpass_;
00110     float Fstop_;
00111     uint32_t in_stream_;
00112     uint32_t out_stream_;
00113
00114     size_t item_size_;
00115
00116     bool quadrature_;
00117     bool rf_dc_;
00118     bool bb_dc_;
00119     bool rx1_enable_;
00120     bool rx2_enable_;
00121     bool enable_dynamic_bit_selection_;
00122     bool enable_ovf_check_buffer_monitor_active_;
00123     bool dump_;
00124     bool rf_shutdown_;
00125 };
00126
00127
00128 /** \} */
00129 /** \} */
00130 #endif // GNSS_SDR_FCOMMS5_SIGNAL_SOURCE_FPGA_H

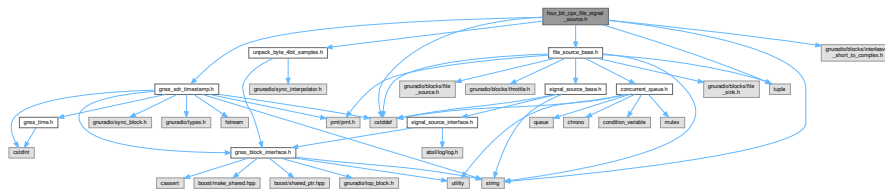
```

## 11.322 four\_bit\_cpx\_file\_signal\_source.h File Reference

Interface of a class that reads signals samples from a 2 bit complex sampler front-end file and adapts it to a [SignalSourceInterface](#).

```
#include "file_source_base.h"
#include "gnss_sdr_timestamp.h"
#include "unpack_byte_4bit_samples.h"
#include <gnuradio/blocks/interleaved_short_to_complex.h>
#include <cstring>
#include <string>
#include <tuple>
```

Include dependency graph for four\_bit\_cpx\_file\_signal\_source.h:



### Classes

- class [FourBitCpxFileSignalSource](#)

*Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).*

### 11.322.1 Detailed Description

Interface of a class that reads signals samples from a 2 bit complex sampler front-end file and adapts it to a [SignalSourceInterface](#).

#### Author

Javier Arribas, 2015 jarribas(at)cttc.es

This class represents a file signal source.

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [four\\_bit\\_cpx\\_file\\_signal\\_source.h](#).

## 11.323 four\_bit\_cpx\_file\_signal\_source.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file four_bit_cpx_file_signal_source.h
00003  * \brief Interface of a class that reads signals samples from a 2 bit complex sampler front-end file
00004  * and adapts it to a SignalSourceInterface.
00005  * \author Javier Arribas, 2015 jarribas(at)cttc.es
00006  *
00007  * This class represents a file signal source.
00008  *
00009  * -----
00010  *
00011  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00012  * This file is part of GNSS-SDR.
00013  *
00014  * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)
00015  * SPDX-License-Identifier: GPL-3.0-or-later
00016  *
00017  * -----
00018  */
00019
00020 #ifndef GNSS_SDR_FOUR_BIT_CPX_FILE_SIGNAL_SOURCE_H
00021 #define GNSS_SDR_FOUR_BIT_CPX_FILE_SIGNAL_SOURCE_H
00022
```

```

00023 #include "file_source_base.h"
00024 #include "gnss_sdr_timestamp.h"
00025 #include "unpack_byte_4bit_samples.h"
00026 #include <gnuradio/blocks/interleaved_short_to_complex.h>
00027 #include <cstdlib>
00028 #include <string>
00029 #include <tuple>
00030
00031 /** \addtogroup Signal_Source
00032  * \{ */
00033 /** \addtogroup Signal_Source_adapters
00034  * \{ */
00035
00036
00037 class ConfigurationInterface;
00038
00039 /*!
00040  * \brief Class that reads signals samples from a file
00041  * and adapts it to a SignalSourceInterface
00042  */
00043 class FourBitCpxFileSignalSource : public FileSourceBase
00044 {
00045 public:
00046     FourBitCpxFileSignalSource(const ConfigurationInterface* configuration,
00047                               const std::string& role,
00048                               unsigned int in_streams,
00049                               unsigned int out_streams,
00050                               Concurrent_Queue<pmt::pmt_t>* queue);
00051
00052     ~FourBitCpxFileSignalSource() = default;
00053
00054 protected:
00055     std::tuple<size_t, bool> itemTypeToSize() override;
00056     double packetsPerSample() const override;
00057     gnss_shared_ptr<gr::block> source() const override;
00058     void create_file_source_hook() override;
00059     void pre_connect_hook(gr::top_block_sptr top_block) override;
00060     void pre_disconnect_hook(gr::top_block_sptr top_block) override;
00061
00062 private:
00063     unpack_byte_4bit_samples_sptr unpack_byte_;
00064     gr::blocks::interleaved_short_to_complex::sptr inter_shorts_to_cpx_;
00065     gnss_shared_ptr<Gnss_Sdr_Timestamp> timestamp_block_;
00066     std::string sample_type_;
00067     std::string timestamp_file_;
00068     double timestamp_clock_offset_ms_;
00069     bool reverse_interleaving_;
00070 };
00071
00072
00073 /** \} */
00074 /** \} */
00075 #endif // GNSS_SDR_FOUR_BIT_CPX_FILE_SIGNAL_SOURCE_H

```

## 11.324 gen\_signal\_source.h File Reference

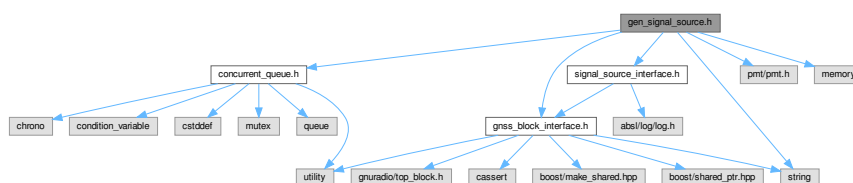
It wraps blocks that generates synthesized GNSS signal and filters it.

```

#include "concurrent_queue.h"
#include "gnss_block_interface.h"
#include "signal_source_interface.h"
#include <pmt/pmt.h>
#include <memory>
#include <string>

```

Include dependency graph for gen\_signal\_source.h:



**Classes**

- class [GenSignalSource](#)

*This class wraps blocks that generates synthesized GNSS signal and filters the signal.*

**11.324.1 Detailed Description**

It wraps blocks that generates synthesized GNSS signal and filters it.

**Author**

Marc Molina, 2013. [marc.molina.pena@gmail.com](mailto:marc.molina.pena@gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [gen\\_signal\\_source.h](#).

---

**11.325 gen\_signal\_source.h**

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file gen_signal_source.h
00003  * \brief It wraps blocks that generates synthesized GNSS signal and filters
00004  * it.
00005  * \author Marc Molina, 2013. marc.molina.pena@gmail.com
00006  *
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019 #ifndef GNSS_SDR_GEN_SIGNAL_SOURCE_H
00020 #define GNSS_SDR_GEN_SIGNAL_SOURCE_H
00021
00022
00023 #include "concurrent_queue.h"
00024 #include "gnss_block_interface.h"
00025 #include "signal_source_interface.h"
00026 #include <pmt/pmt.h>
00027 #include <memory>
00028 #include <string>
00029
00030
00031 /** \addtogroup Signal_Source
00032  * \{ */
00033 /** \addtogroup Signal_Source_adapters
00034  * \{ */
00035
00036
00037 /*!
00038 * \brief This class wraps blocks that generates synthesized GNSS signal and
00039 * filters the signal.
00040 */
00041 class GenSignalSource : public SignalSourceInterface
00042 {
00043 public:
00044     /** Constructor
00045     GenSignalSource(std::shared_ptr<GNSSBlockInterface> signal_generator,
00046         std::shared_ptr<GNSSBlockInterface> filter,
00047         std::string role, Concurrent_Queue<pmt::pmt_t> *queue);
00048
00049     void connect(gr::top_block_sptr top_block) override;
00050     void disconnect(gr::top_block_sptr top_block) override;
00051     gr::basic_block_sptr get_left_block() override;
00052     gr::basic_block_sptr get_right_block() override;
00053
00054     inline std::string role() override { return role_; }
00055     /** Returns "Signal Source"
00056     inline std::string implementation() override { return "Signal Source"; }
00057     inline size_t item_size() override { return 0; }
00058     inline size_t getRFChannels() const final { return 0; }

```

```

00059     inline std::shared_ptr<GNSSBlockInterface> signal_generator() const { return signal_generator_; }
00060
00061 private:
00062     std::shared_ptr<GNSSBlockInterface> signal_generator_;
00063     std::shared_ptr<GNSSBlockInterface> filter_;
00064     std::string role_;
00065     std::string implementation_;
00066     bool connected_;
00067 };
00068
00069
00070 /** \} */
00071 /** \} */
00072 #endif // GNSS_SDR_GEN_SIGNAL_SOURCE_H

```

## 11.326 ion\_gsms\_signal\_source.h File Reference

GNSS-SDR Signal Source that reads sample streams following ION's GNSS-SDR metadata standard.

```

#include "configuration_interface.h"
#include "file_source_base.h"
#include "gnss_sdr_timestamp.h"
#include "ion_gsms.h"
#include <GnssMetadata.h>
#include <cstdint>
#include <memory>
#include <string>
#include <vector>

```

Include dependency graph for ion\_gsms\_signal\_source.h:



### Classes

- class [IONGSMSSignalSource](#)

*Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).*

### 11.326.1 Detailed Description

GNSS-SDR Signal Source that reads sample streams following ION's GNSS-SDR metadata standard.

#### Author

Víctor Castillo Agüero, 2024. [victorcastilloaguero\(at\)gmail.com](mailto:victorcastilloaguero(at)gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [ion\\_gsms\\_signal\\_source.h](#).

---

## 11.327 ion\_gsms\_signal\_source.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file ion_gsms_signal_source.h
00003  * \brief GNSS-SDR Signal Source that reads sample streams following ION's GNSS-SDR metadata standard
00004  * \author Víctor Castillo Agüero, 2024. victorcastilloaguero(at)gmail.com
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors)

```



```

00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_ION_GSMS_SIGNAL_SOURCE_H
00019 #define GNSS_SDR_ION_GSMS_SIGNAL_SOURCE_H
00020
00021 #include "configuration_interface.h"
00022 #include "file_source_base.h"
00023 #include "gnss_sdr_timestamp.h"
00024 #include "ion_gsms.h"
00025 #include <GnssMetadata.h>
00026 #include <cstdint>
00027 #include <memory>
00028 #include <string>
00029 #include <vector>
00030
00031 /** \addtogroup Signal_Source
00032  * \{ */
00033 /** \addtogroup Signal_Source_adapters
00034  * \{ */
00035
00036 /*!
00037  * \brief Class that reads signals samples from a file
00038  * and adapts it to a SignalSourceInterface
00039  */
00040 class IONGSMSSignalSource : public SignalSourceBase
00041 {
00042 public:
00043     IONGSMSSignalSource(const ConfigurationInterface* configuration, const std::string& role,
00044         unsigned int in_streams, unsigned int out_streams,
00045         Concurrent_Queue<pmt::pmt_t*> queue);
00046
00047     ~IONGSMSSignalSource() override = default;
00048
00049 protected:
00050     void connect(gr::top_block_sptr top_block) override;
00051     void disconnect(gr::top_block_sptr top_block) override;
00052
00053     gr::basic_block_sptr get_left_block() override;
00054     gr::basic_block_sptr get_right_block() override;
00055     gr::basic_block_sptr get_right_block(int RF_channel) override;
00056
00057     inline size_t item_size() override
00058     {
00059         return (*sources_.begin())->output_stream_item_size(0);
00060     }
00061
00062 private:
00063     std::vector<IONGSMSSignalSource::sptr> make_stream_sources(const std::vector<std::string>&
00064         stream_ids) const;
00065
00066     void load_metadata();
00067
00068     std::vector<std::string> stream_ids_;
00069     std::vector<IONGSMSSignalSource::sptr> sources_;
00070     std::vector<gnss_shared_ptr<gr::block>> copy_blocks_;
00071     std::vector<gnss_shared_ptr<gr::block>> valves_;
00072
00073     std::string metadata_filepath_;
00074     std::shared_ptr<GnssMetadata::Metadata> metadata_;
00075
00076     gnss_shared_ptr<Gnss_Sdr_Timestamp> timestamp_block_;
00077     std::string timestamp_file_;
00078
00079     uint32_t in_streams_;
00080     uint32_t out_streams_;
00081 };
00082
00083 /** \} */
00084 /** \} */
00085 #endif // GNSS_SDR_ION_GSMS_SIGNAL_SOURCE_H

```

## 11.328 labsat\_signal\_source.h File Reference

LabSat version 2, 3, and 3 Wideband format reader.

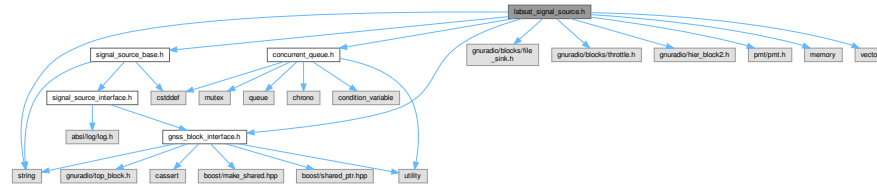
```

#include "concurrent_queue.h"
#include "gnss_block_interface.h"
#include "signal_source_base.h"

```

```
#include <gnuradio/blocks/file_sink.h>
#include <gnuradio/blocks/throttle.h>
#include <gnuradio/hier_block2.h>
#include <pmt/pmt.h>
#include <memory>
#include <string>
#include <vector>
```

Include dependency graph for labsat\_signal\_source.h:



## Classes

- class [LabsatSignalSource](#)

*This class reads samples stored in LabSat version 2, 3, and 3 Wideband format.*

## 11.328.1 Detailed Description

LabSat version 2, 3, and 3 Wideband format reader.

### Author

Javier Arribas, jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [labsat\\_signal\\_source.h](#).

---

## 11.329 labsat\_signal\_source.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file labsat_signal_source.h
00003  * \brief LabSat version 2, 3, and 3 Wideband format reader
00004  * \author Javier Arribas, jarribas(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_LABSAT_SIGNAL_SOURCE_H
00019 #define GNSS_SDR_LABSAT_SIGNAL_SOURCE_H
00020
00021 #include "concurrent_queue.h"
00022 #include "gnss_block_interface.h"
00023 #include "signal_source_base.h"
00024 #include <gnuradio/blocks/file_sink.h>
00025 #include <gnuradio/blocks/throttle.h>
00026 #include <gnuradio/hier_block2.h>
00027 #include <pmt/pmt.h>
00028 #include <memory>
00029 #include <string>
00030 #include <vector>
00031
```

```

00032 /** \addtogroup Signal_Source
00033 * \{ */
00034 /** \addtogroup Signal_Source_adapters
00035 * \{ */
00036
00037
00038 class ConfigurationInterface;
00039
00040 /*!
00041 * \brief This class reads samples stored in LabSat version 2, 3, and 3 Wideband
00042 * format.
00043 */
00044 class LabsatSignalSource : public SignalSourceBase
00045 {
00046 public:
00047     LabsatSignalSource(const ConfigurationInterface* configuration,
00048         const std::string& role, unsigned int in_stream,
00049         unsigned int out_stream, Concurrent_Queue<pmt::pmt_t>* queue);
00050
00051     ~LabsatSignalSource() = default;
00052
00053     inline size_t item_size() override
00054     {
00055         return item_size_;
00056     }
00057
00058     size_t getRfChannels() const override;
00059     void connect(gr::top_block_sptr top_block) override;
00060     void disconnect(gr::top_block_sptr top_block) override;
00061     gr::basic_block_sptr get_left_block() override;
00062     gr::basic_block_sptr get_right_block() override;
00063     gr::basic_block_sptr get_right_block(int i) override;
00064
00065 private:
00066     gr::block_sptr labsat23_source_;
00067     std::vector<gr::blocks::file_sink::sptr> file_sink_;
00068     std::vector<gr::blocks::throttle::sptr> throttle_;
00069     std::vector<int> channels_selector_vec_;
00070
00071     std::string item_type_;
00072     std::string filename_;
00073     std::string dump_filename_;
00074
00075     size_t item_size_;
00076
00077     unsigned int in_stream_;
00078     unsigned int out_stream_;
00079
00080     bool enable_throttle_control_;
00081     bool dump_;
00082 };
00083
00084 /** \} */
00085 /** \} */
00086 #endif // GNSS_SDR_LABSAT_SIGNAL_SOURCE_H

```

## 11.330 limesdr\_signal\_source.h

```

00001 /*!
00002 * \file limesdr_signal_source.cc
00003 * \brief Signal source for LimeSDR front-end
00004 * \author Javier Arribas, 2021. jarribas(at)cttc.es
00005 *
00006 * -----
00007 *
00008 * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009 * This file is part of GNSS-SDR.
00010 *
00011 * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)
00012 * SPDX-License-Identifier: GPL-3.0-or-later
00013 *
00014 * -----
00015 */
00016
00017 #ifndef GNSS_SDR_LIMESDR_SIGNAL_SOURCE_H
00018 #define GNSS_SDR_LIMESDR_SIGNAL_SOURCE_H
00019
00020 #include "concurrent_queue.h"
00021 #include "signal_source_base.h"
00022 #include <gnuradio/blocks/file_sink.h>
00023 #include <pmt/pmt.h>
00024 #include <cstdint>
00025 #include <limesdr/source.h>
00026 #include <memory>

```

```

00027 #include <stdexcept>
00028 #include <string>
00029
00030 /** \addtogroup Signal_Source
00031  * \{ */
00032 /** \addtogroup Signal_Source_adapters
00033  * \{ */
00034
00035
00036 class ConfigurationInterface;
00037
00038 /*!
00039  * \brief This class instantiates the LimeSDR gnuradio signal source.
00040  * It has support also for a customized LimeSDR firmware and signal source to support PPS samplestamp
00041  * reading.
00042 */
00042 class LimesdrSignalSource : public SignalSourceBase
00043 {
00044 public:
00045     LimesdrSignalSource(const ConfigurationInterface* configuration,
00046         const std::string& role, unsigned int in_stream,
00047         unsigned int out_stream, Concurrent_Queue<pmt::pmt_t>* queue);
00048
00049     ~LimesdrSignalSource() = default;
00050
00051     inline size_t item_size() override
00052     {
00053         return item_size_;
00054     }
00055
00056     void connect(gr::top_block_sptr top_block) override;
00057     void disconnect(gr::top_block_sptr top_block) override;
00058     gr::basic_block_sptr get_left_block() override;
00059     gr::basic_block_sptr get_right_block() override;
00060
00061 private:
00062     gr::limesdr::source::sptr limesdr_source_;
00063     gnss_shared_ptr<gr::block> valve_;
00064     gr::blocks::file_sink::sptr file_sink_;
00065
00066     std::string item_type_;
00067     std::string dump_filename_;
00068     std::string limesdr_serial_;
00069     std::string limesdr_file_;
00070
00071     // Front-end settings
00072     double sample_rate_;
00073     double freq_;
00074     double gain_;
00075     double analog_bw_hz_;
00076     double digital_bw_hz_;
00077     double ext_clock_MHz_;
00078     size_t item_size_;
00079     int64_t samples_;
00080
00081     unsigned int in_stream_;
00082     unsigned int out_stream_;
00083
00084     int limechannel_mode_;
00085     int antenna_;
00086     int channel_;
00087
00088     bool PPS_mode_;
00089     bool dump_;
00090 };
00091
00092
00093 /** \} */
00094 /** \} */
00095 #endif // GNSS_SDR_LIMESDR_SIGNAL_SOURCE_H

```

## 11.331 max2771\_evkit\_signal\_source\_fpga.h File Reference

Signal source for the MAX2771EVKIT evaluation board connected directly to FPGA accelerators. This source implements only the MAX2771 control. It is NOT compatible with conventional SDR acquisition and tracking blocks.

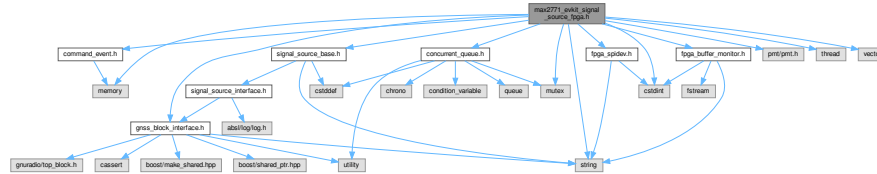
```

#include "command_event.h"
#include "concurrent_queue.h"
#include "fpga_buffer_monitor.h"
#include "fpga_spidev.h"
#include "gnss_block_interface.h"
#include "signal_source_base.h"

```

```
#include <pmt/pmt.h>
#include <cstdint>
#include <memory>
#include <mutex>
#include <string>
#include <thread>
#include <vector>
```

Include dependency graph for max2771\_evkit\_signal\_source\_fpga.h:



## Classes

- class [MAX2771EVKITSignalSourceFPGA](#)

### 11.331.1 Detailed Description

Signal source for the MAX2771EVKIT evaluation board connected directly to FPGA accelerators. This source implements only the MAX2771 control. It is NOT compatible with conventional SDR acquisition and tracking blocks. GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [max2771\\_evkit\\_signal\\_source\\_fpga.h](#).

## 11.332 max2771\_evkit\_signal\_source\_fpga.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file max2771_evkit_signal_source_fpga.h
00003  * \brief Signal source for the MAX2771EVKIT evaluation board connected directly
00004  * to FPGA accelerators.
00005  * This source implements only the MAX2771 control. It is NOT compatible with
00006  * conventional SDR acquisition and tracking blocks.
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019 #ifndef GNSS_SDR_MAX2771_EVKIT_SIGNAL_SOURCE_FPGA_H
00020 #define GNSS_SDR_MAX2771_EVKIT_SIGNAL_SOURCE_FPGA_H
00021
00022 #include "command_event.h"
00023 #include "concurrent_queue.h"
00024 #include "fpga_buffer_monitor.h"
00025 #include "fpga_spidev.h"
00026 #include "gnss_block_interface.h"
00027 #include "signal_source_base.h"
00028 #include <pmt/pmt.h> // for pmt::pmt_t
00029 #include <cstdint> // for fixed-width integer types
00030 #include <memory> // for smart pointers
00031 #include <mutex> // for mutex
00032 #include <string> // for strings
00033 #include <thread> // for threads
00034 #include <vector> // for std::vector
00035
00036
00037 /** \addtogroup Signal_Source
```

```

00038  * \{ */
00039  /** \addtogroup Signal_Source_adapters
00040  * \{ */
00041
00042
00043  class ConfigurationInterface;
00044
00045  class MAX2771EVKITSignalSourceFPGA : public SignalSourceBase
00046  {
00047  public:
00048      MAX2771EVKITSignalSourceFPGA(const ConfigurationInterface *configuration,
00049          const std::string &role, unsigned int in_stream,
00050          unsigned int out_stream, Concurrent_Queue<pmt::pmt_t> *queue);
00051
00052      ~MAX2771EVKITSignalSourceFPGA();
00053
00054      std::vector<uint32_t> setup_regs(void);
00055
00056      inline size_t item_size() override
00057      {
00058          return item_size_;
00059      }
00060
00061      void connect(gr::top_block_sptr top_block) override;
00062      void disconnect(gr::top_block_sptr top_block) override;
00063      gr::basic_block_sptr get_left_block() override;
00064      gr::basic_block_sptr get_right_block() override;
00065
00066  private:
00067      const std::string default_dump_filename = std::string("FPGA_buffer_monitor_dump.dat");
00068      const uint64_t default_bandwidth = 2500000;
00069      const uint32_t default_filter_order = 5;
00070      const uint64_t default_sampling_rate = 4092000;
00071      const uint32_t default_PGA_gain_value = 0x3A; // default PGA gain when AGC is off
00072      // max PGA gain value
00073      const uint32_t max_PGA_gain_value = 0x3F;
00074      // check buffer overflow and perform buffer monitoring every 1s by default
00075      const uint32_t buffer_monitor_period_ms = 1000;
00076      // buffer overflow and buffer monitoring initial delay
00077      const uint32_t buffer_monitoring_initial_delay_ms = 2000;
00078      // MAX2771 number of configuration registers
00079      const uint32_t MAX2771_NUM_REGS = 11;
00080      // MAX2771 configuration register fields
00081      const uint32_t NUM_FREQ_BANDS = 1;
00082      const uint32_t IDLE = 0x0; // Idle mode disabled
00083      const uint32_t MIXPOLE = 0x0; // set the passive filter pole at mixer output at 13 MHz.
00084      const uint32_t MIXERMODE = 0x0; // L1 band enabled
00085      const uint32_t FCEN = 0x58; // Center frequency not used when in low-pass filter mode.
00086      // Set to default value.
00087      const uint32_t FCENX = 0x0; // POLYphase filter selection set to Lowpass filter
00088      const uint32_t ANAIMON = 0x0; // analog monitor disabled
00089      const uint32_t IQEN = 0x1; // I and Q channels enable
00090      const uint32_t GAINREF = 0xAA; // AGC Gain ref
00091      const uint32_t SPI_SDIO_CONFIG = 0x0; // SPI SDIO config when tri-stated: nothing applied
00092      const uint32_t FORMAT = 0x1; // sign and magnitude
00093      const uint32_t BITS = 0x2; // number of bits in the ADC = 2
00094      const uint32_t DRVCFG = 0x0; // output driver configuration = CMOS Logic
00095      const uint32_t DIEID = 0x0; // identifies version of IC
00096      const uint32_t HILOADEN = 0x0; // disable output driver for high loads
00097      const uint32_t FHIPEN = 0x1; // enable highpass coupling between filter and PGA.
00098      const uint32_t PGAEN = 0x1; // I-Channel PGA Enable
00099      const uint32_t PGAQEN = 0x1; // Q-Channel PGA Enable
00100      const uint32_t STRMEN = 0x0; // disable DSP interface for serial streaming of data
00101      const uint32_t STRMSTART = 0x0; // the rising edge of this bit enables data streaming to
00102      // the output, clock, data, sync and frame sync outputs.
00103      const uint32_t STRMSTOP = 0x0; // the rising edge of this bit disables data streaming to
00104      // the output, clock, data sync and frame sync outputs.
00105      const uint32_t STRMBITS = 0x1; // number of bits to be streamed: I MSB, I LSB
00106      const uint32_t STAMPEN = 0x1; // enable frame number insertion
00107      const uint32_t TIMESYNCEN = 0x1; // enable the output of the time sync pulses at all times
00108      // when streaming is enabled.
00109      const uint32_t DATASYNCEN = 0x0; // disable the sync pulses at the DATASYNC output
00110      const uint32_t STRMRST = 0x0; // counter reset not active
00111      const uint32_t LOBAND = 0x0; // L1 band
00112      const uint32_t REFOUTEN = 0x1; // Output clock buffer enable
00113      const uint32_t IXTAL = 0x1; // XTAL oscillator/buffer set to normal current
00114      const uint32_t ICP = 0x0; // charge pump current selection set to 0.5 mA
00115      const uint32_t INT_PLL = 0x1; // PLL mode set to integer-N PLL
00116      const uint32_t PWRSAV = 0x0; // PLL power save mode disabled
00117      const uint32_t RDIV = 0x10; // Set the PLL reference division ratio such that the L1
00118      // band is tuned to 1575.42 Mhz
00119      const uint32_t FDIV = 0x80000; // PLL fractional division ratio not used. Set to default
00120      // value
00121      const uint32_t EXTADCCLK = 0x0; // use internally generated clock
00122      const uint32_t REFCLK_L_CNT = 0x100; // set the L counter of the reference clock configuration
00123      // to its default value
00124      const uint32_t REFCLK_M_CNT = 0x61B; // set the M counter of the reference clock configuration

```

```

        to its default value
00118     const uint32_t FCLKIN = 0x0;           // fractional clock divider set to default value
00119     const uint32_t ADCCLK = 0x0;           // ADC clock selection set to reference clock
        divider/multiplier
00120     const uint32_t MODE = 0x0;             // DSP interface mode selection
00121     const uint32_t ADCCLK_L_CNT = 0x100;    // set the L counter of the ADC clock configuration to its
        default value
00122     const uint32_t ADCCLK_M_CNT = 0x61B;    // set the M counter of the ADC clock configuration to its
        default value
00123     const uint32_t PRE_FRACDIV_SEL = 0x0;   // bypass fractional clock divider
00124     const uint32_t CLKOUT_SEL = 0x1;        // CLKOUT selection set to ADC clock
00125     // MAX2771 configuration register registers
00126     const uint32_t TEST_MODE_1_REG_VAL = 0x01E0F401; // reserved
00127     const uint32_t TEST_MODE_2_REG_VAL = 0x00000002;
00128
00129     bool configure(std::vector<uint32_t> register_values);
00130     void run_buffer_monitor_process();
00131
00132     mutable std::mutex buffer_monitor_mutex;
00133
00134     std::thread thread_buffer_monitor;
00135
00136     std::shared_ptr<Fpga_buffer_monitor> buffer_monitor_fpga;
00137     std::shared_ptr<Fpga_spidev> spidev_fpga;
00138
00139     uint64_t freq; // frequency of local oscillator
00140     uint64_t sample_rate;
00141
00142     uint32_t in_stream;
00143     uint32_t out_stream;
00144     uint32_t bandwidth; // 2500000, 4200000, 8700000, 16400000, 23400000, 36000000
00145     uint32_t filter_order; // 3, 5
00146     uint32_t gain_in; // 0 to 0x3F
00147
00148     size_t item_size; // 1
00149
00150     bool chipen; // chip enable
00151     bool if_filter_gain; // true, false
00152     bool LNA_active; // true, false
00153     bool enable_agc; // true, false
00154     bool enable_ovf_check_buffer_monitor_active;
00155     bool dump;
00156     bool rf_shutdown;
00157 };
00158
00159
00160 /** \} */
00161 /** \} */
00162 #endif // GNSS_SDR_MAX2771_EVKIT_SIGNAL_SOURCE_FPGA_H

```

## 11.333 multichannel\_file\_signal\_source.h File Reference

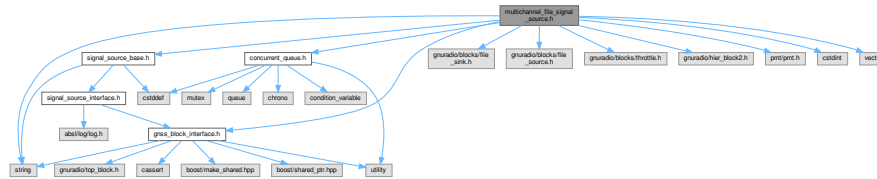
Implementation of a class that reads signals samples from files at different frequency band and adapts it to a [SignalSourceInterface](#).

```

#include "concurrent_queue.h"
#include "gnss_block_interface.h"
#include "signal_source_base.h"
#include <gnuradio/blocks/file_sink.h>
#include <gnuradio/blocks/file_source.h>
#include <gnuradio/blocks/throttle.h>
#include <gnuradio/hier_block2.h>
#include <pmt/pmt.h>
#include <stdint>
#include <string>
#include <vector>

```

Include dependency graph for `multichannel_file_signal_source.h`:



## Classes

- class [MultichannelFileSignalSource](#)

*Class that reads signals samples from files at different frequency bands and adapts it to a [SignalSourceInterface](#).*

### 11.333.1 Detailed Description

Implementation of a class that reads signals samples from files at different frequency band and adapts it to a [SignalSourceInterface](#).

#### Author

Javier Arribas, 2019 jarribas(at)cttc.es

This class represents a file signal source. Internally it uses a GNU Radio's `gr_file_source` as a connector to the data.

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [multichannel\\_file\\_signal\\_source.h](#).

## 11.334 multichannel\_file\_signal\_source.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file multichannel_file_signal_source.h
00003  * \brief Implementation of a class that reads signals samples from files at
00004  * different frequency band and adapts it to a SignalSourceInterface
00005  * \author Javier Arribas, 2019 jarribas(at)cttc.es
00006  *
00007  * This class represents a file signal source. Internally it uses a GNU Radio's
00008  * gr_file_source as a connector to the data.
00009  *
00010  * -----
00011  *
00012  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00013  * This file is part of GNSS-SDR.
00014  *
00015  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00016  * SPDX-License-Identifier: GPL-3.0-or-later
00017  *
00018  * -----
00019  */
00020
00021 #ifndef GNSS_SDR_MULTICHANNEL_FILE_SIGNAL_SOURCE_H
00022 #define GNSS_SDR_MULTICHANNEL_FILE_SIGNAL_SOURCE_H
00023
00024 #include "concurrent_queue.h"
00025 #include "gnss_block_interface.h"
00026 #include "signal_source_base.h"
00027 #include <gnuradio/blocks/file_sink.h>
00028 #include <gnuradio/blocks/file_source.h>
00029 #include <gnuradio/blocks/throttle.h>
00030 #include <gnuradio/hier_block2.h>
00031 #include <pmt/pmt.h>
00032 #include <cstdint>
00033 #include <string>
00034 #include <vector>
00035
00036
00037 /** \addtogroup Signal_Source
00038  * \{ */
```



```

00039 /** \addtogroup Signal_Source_adapters
00040 * \{ */
00041
00042
00043 class ConfigurationInterface;
00044
00045 /*!
00046 * \brief Class that reads signals samples from files at different frequency bands
00047 * and adapts it to a SignalSourceInterface
00048 */
00049 class MultichannelFileSignalSource : public SignalSourceBase
00050 {
00051 public:
00052     MultichannelFileSignalSource(const ConfigurationInterface* configuration, const std::string& role,
00053         unsigned int in_streams, unsigned int out_streams,
00054         Concurrent_Queue<pmt::pmt_t>* queue);
00055
00056     ~MultichannelFileSignalSource() = default;
00057
00058     inline size_t item_size() override
00059     {
00060         return item_size_;
00061     }
00062
00063     void connect(gr::top_block_sptr top_block) override;
00064     void disconnect(gr::top_block_sptr top_block) override;
00065     gr::basic_block_sptr get_left_block() override;
00066     gr::basic_block_sptr get_right_block() override;
00067
00068     inline std::string filename() const
00069     {
00070         return filename_vec_.at(0);
00071     }
00072
00073     inline std::string item_type() const
00074     {
00075         return item_type_;
00076     }
00077
00078     inline bool repeat() const
00079     {
00080         return repeat_;
00081     }
00082
00083     inline int64_t sampling_frequency() const
00084     {
00085         return sampling_frequency_;
00086     }
00087
00088     inline uint64_t samples() const
00089     {
00090         return samples_;
00091     }
00092
00093 private:
00094     std::vector<gr::blocks::file_source::sptr> file_source_vec_;
00095     gnss_shared_ptr<gr::block> valve_;
00096     gr::blocks::file_sink::sptr sink_;
00097     std::vector<gr::blocks::throttle::sptr> throttle_vec_;
00098     std::vector<std::string> filename_vec_;
00099     std::string item_type_;
00100     uint64_t samples_;
00101     int64_t sampling_frequency_;
00102     size_t item_size_;
00103     int32_t n_channels_;
00104     uint32_t in_streams_;
00105     uint32_t out_streams_;
00106     bool repeat_;
00107     // Throttle control
00108     bool enable_throttle_control_;
00109 };
00110
00111
00112 /** \} */
00113 /** \} */
00114 #endif // GNSS_SDR_MULTICHANNEL_FILE_SIGNAL_SOURCE_H

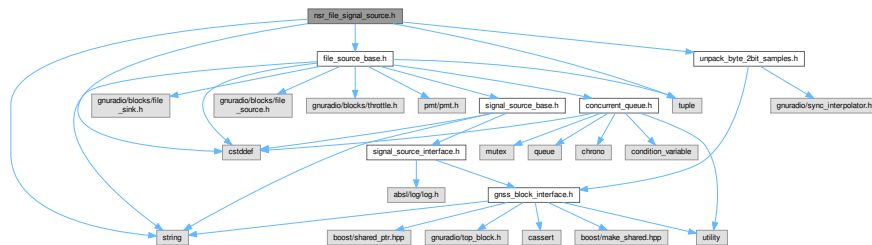
```

## 11.335 nsr\_file\_signal\_source.h File Reference

Implementation of a class that reads signals samples from a NSR 2 bits sampler front-end file and adapts it to a [SignalSourceInterface](#). More information about the front-end here <http://www.ifen.com/products/sx-scientific-gnss-solutions/nsr-software-receiver.html>.

```
#include "file_source_base.h"
#include "unpack_byte_2bit_samples.h"
#include <cstdio>
#include <string>
#include <tuple>
```

Include dependency graph for nsr\_file\_signal\_source.h:



## Classes

- class [NsrFileSignalSource](#)

*Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).*

### 11.335.1 Detailed Description

Implementation of a class that reads signals samples from a NSR 2 bits sampler front-end file and adapts it to a [SignalSourceInterface](#). More information about the front-end here <http://www.ifen.com/products/sx-scientific-gnss-solutions/nsr-software-receiver.html>.

#### Author

Javier Arribas, 2013 jarribas(at)cttc.es

This class represents a file signal source.

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [nsr\\_file\\_signal\\_source.h](#).

## 11.336 nsr\_file\_signal\_source.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file nsr_file_signal_source.h
00003  * \brief Implementation of a class that reads signals samples from a NSR 2 bits sampler front-end
00004  * file
00005  * and adapts it to a SignalSourceInterface. More information about the front-end here
00006  * http://www.ifen.com/products/sx-scientific-gnss-solutions/nsr-software-receiver.html
00007  * \author Javier Arribas, 2013 jarribas(at)cttc.es
00008  *
00009  * This class represents a file signal source.
00010  * -----
00011  *
00012  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00013  * This file is part of GNSS-SDR.
00014  *
00015  * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)
00016  * SPDX-License-Identifier: GPL-3.0-or-later
00017  *
00018  * -----
00019  */
00020
00021 #ifndef GNSS_SDR_NSR_FILE_SIGNAL_SOURCE_H
00022 #define GNSS_SDR_NSR_FILE_SIGNAL_SOURCE_H
00023
00024 #include "file_source_base.h"
00025 #include "unpack_byte_2bit_samples.h"
```

```

00026 #include <cstdlib>
00027 #include <string>
00028 #include <tuple>
00029
00030 /** \addtogroup Signal_Source
00031  * \{ */
00032 /** \addtogroup Signal_Source_adapters
00033  * \{ */
00034
00035 class ConfigurationInterface;
00036
00037 /*!
00038  * \brief Class that reads signals samples from a file
00039  * and adapts it to a SignalSourceInterface
00040  */
00041 class NsrFileSignalSource : public FileSourceBase
00042 {
00043 public:
00044     NsrFileSignalSource(const ConfigurationInterface* configuration, const std::string& role,
00045         unsigned int in_streams, unsigned int out_streams,
00046         Concurrent_Queue<pmt::pmt_t*> queue);
00047
00048     ~NsrFileSignalSource() = default;
00049
00050 protected:
00051     std::tuple<size_t, bool> itemTypeToSize() override;
00052     double packetsPerSample() const override;
00053     gnss_shared_ptr<gr::block> source() const override;
00054     void create_file_source_hook() override;
00055     void pre_connect_hook(gr::top_block_sptr top_block) override;
00056     void pre_disconnect_hook(gr::top_block_sptr top_block) override;
00057 private:
00058     unpack_byte_2bit_samples_sptr unpack_byte_;
00059 };
00060
00061
00062
00063 /** \} */
00064 /** \} */
00065 #endif // GNSS_SDR_NSR_FILE_SIGNAL_SOURCE_H

```

## 11.337 osmosdr\_signal\_source.h File Reference

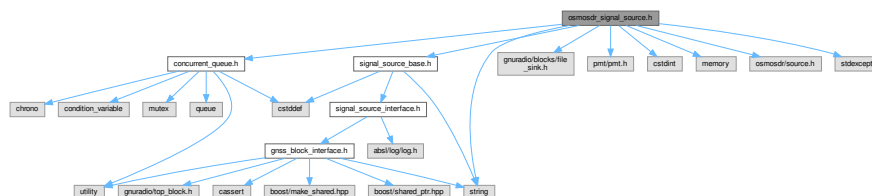
Signal source wrapper for OsmoSDR-compatible front-ends, such as HackRF or Realtek's RTL2832U-based USB dongle DVB-T receivers (see <https://osmocom.org/projects/rtl-sdr/wiki> for more information)

```

#include "concurrent_queue.h"
#include "signal_source_base.h"
#include <gnuradio/blocks/file_sink.h>
#include <pmt/pmt.h>
#include <stdint>
#include <memory>
#include <osmosdr/source.h>
#include <stdexcept>
#include <string>

```

Include dependency graph for osmosdr\_signal\_source.h:



### Classes

- class [OsmosdrSignalSource](#)

*This class reads samples OsmoSDR-compatible front-ends, such as HackRF or Realtek's RTL2832U-based USB dongle DVB-T receivers (see <https://osmocom.org/projects/rtl-sdr/wiki>)*

### 11.337.1 Detailed Description

Signal source wrapper for OsmoSDR-compatible front-ends, such as HackRF or Realtek's RTL2832U-based USB dongle DVB-T receivers (see <https://osmocom.org/projects/rtl-sdr/wiki> for more information)

#### Author

Javier Arribas, 2012. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [osmosdr\\_signal\\_source.h](#).

---

## 11.338 osmosdr\_signal\_source.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file osmosdr_signal_source.h
00003  * \brief Signal source wrapper for OsmoSDR-compatible front-ends, such as
00004  * HackRF or Realtek's RTL2832U-based USB dongle DVB-T receivers
00005  * (see https://osmocom.org/projects/rtl-sdr/wiki for more information)
00006  * \author Javier Arribas, 2012. jarribas(at)cttc.es
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019 #ifndef GNSS_SDR_OSMOSDR_SIGNAL_SOURCE_H
00020 #define GNSS_SDR_OSMOSDR_SIGNAL_SOURCE_H
00021
00022 #include "concurrent_queue.h"
00023 #include "signal_source_base.h"
00024 #include <gnuradio/blocks/file_sink.h>
00025 #include <pmt/pmt.h>
00026 #include <stdint>
00027 #include <memory>
00028 #include <osmosdr/source.h>
00029 #include <stdexcept>
00030 #include <string>
00031
00032 /** \addtogroup Signal_Source
00033  * \{ */
00034 /** \addtogroup Signal_Source_adapters
00035  * \{ */
00036
00037
00038 class ConfigurationInterface;
00039
00040 /*!
00041  * \brief This class reads samples OsmoSDR-compatible front-ends, such as
00042  * HackRF or Realtek's RTL2832U-based USB dongle DVB-T receivers
00043  * (see https://osmocom.org/projects/rtl-sdr/wiki)
00044  */
00045 class OsmosdrSignalSource : public SignalSourceBase
00046 {
00047 public:
00048     OsmosdrSignalSource(const ConfigurationInterface* configuration,
00049         const std::string& role, unsigned int in_stream,
00050         unsigned int out_stream, Concurrent_Queue<pmt::pmt_t>* queue);
00051
00052     ~OsmosdrSignalSource() = default;
00053
00054     inline size_t item_size() override
00055     {
00056         return item_size_;
00057     }
00058
00059     void connect(gr::top_block_sptr top_block) override;
00060     void disconnect(gr::top_block_sptr top_block) override;
00061     gr::basic_block_sptr get_left_block() override;
00062     gr::basic_block_sptr get_right_block() override;
00063
00064 private:
00065     void driver_instance();
```

```

00066
00067     osmosdr::source::sptr osmosdr_source_;
00068     gnss_shared_ptr<gr::block> valve_;
00069     gr::blocks::file_sink::sptr file_sink_;
00070
00071     std::string item_type_;
00072     std::string dump_filename_;
00073     std::string osmosdr_args_;
00074     std::string antenna_;
00075
00076     // Front-end settings
00077     double sample_rate_;
00078     double freq_;
00079     double gain_;
00080     double if_gain_;
00081     double rf_gain_;
00082     double if_bw_;
00083
00084     size_t item_size_;
00085     int64_t samples_;
00086
00087     unsigned int in_stream_;
00088     unsigned int out_stream_;
00089
00090     bool AGC_enabled_;
00091     bool dump_;
00092 };
00093
00094
00095 /** \} */
00096 /** \} */
00097 #endif // GNSS_SDR_OSMOSDR_SIGNAL_SOURCE_H

```

## 11.339 plutosdr\_signal\_source.h File Reference

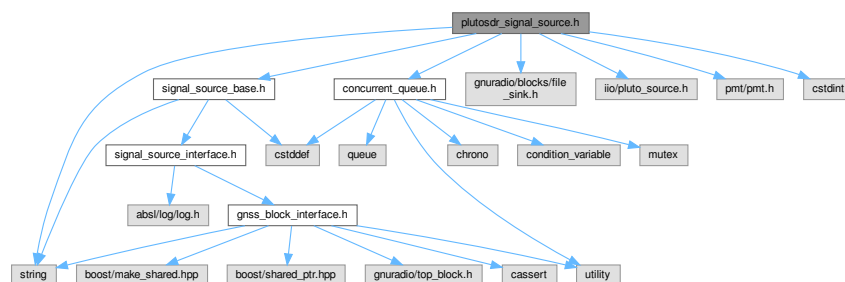
Signal source for PlutoSDR.

```

#include "signal_source_base.h"
#include <gnuradio/blocks/file_sink.h>
#include <iio/pluto_source.h>
#include "concurrent_queue.h"
#include <pmt/pmt.h>
#include <cstdint>
#include <string>

```

Include dependency graph for plutosdr\_signal\_source.h:



### Classes

- class [PlutosdrSignalSource](#)

### 11.339.1 Detailed Description

Signal source for PlutoSDR.

## Author

Rodrigo Muñoz, 2017, [rmunozl\(at\)inacap.cl](mailto:rmunozl@inacap.cl), [rodrigo.munoz\(at\)proteinlab.cl](mailto:rodrigo.munoz(at)proteinlab.cl)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [plutosdr\\_signal\\_source.h](#).

---

## 11.340 plutosdr\_signal\_source.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file plutosdr_signal_source.h
00003  * \brief Signal source for PlutoSDR
00004  * \author Rodrigo Muñoz, 2017, rmunozl\(at\)inacap.cl, rodrigo.munoz\(at\)proteinlab.cl
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018
00019 #ifndef GNSS_SDR_PLUTOSDR_SIGNAL_SOURCE_H
00020 #define GNSS_SDR_PLUTOSDR_SIGNAL_SOURCE_H
00021
00022 #include "signal_source_base.h"
00023 #include <gnuradio/blocks/file_sink.h>
00024 #if GRIIO_INCLUDE_HAS_GNURADIO
00025 #if GR_IIO_TEMPLATIZED_API
00026 #include <gnuradio/iio/fmcomms2_source.h>
00027 #else
00028 #include <gnuradio/iio/pluto_source.h>
00029 #endif
00030 #else
00031 #include <iio/pluto_source.h>
00032 #endif
00033 #include "concurrent_queue.h"
00034 #include <pmt/pmt.h>
00035 #include <cstdint>
00036 #include <string>
00037
00038
00039 /** \addtogroup Signal_Source
00040  * \{ */
00041 /** \addtogroup Signal_Source_adapters
00042  * \{ */
00043
00044
00045 class ConfigurationInterface;
00046
00047 /*!
00048 */
00049 class PlutosdrSignalSource : public SignalSourceBase
00050 {
00051 public:
00052     PlutosdrSignalSource(const ConfigurationInterface* configuration,
00053                         const std::string& role, unsigned int in_stream,
00054                         unsigned int out_stream, Concurrent_Queue<pmt::pmt_t>* queue);
00055
00056     ~PlutosdrSignalSource() = default;
00057
00058     size_t item_size() override
00059     {
00060         return item_size_;
00061     }
00062
00063     void connect(gr::top_block_sptr top_block) override;
00064     void disconnect(gr::top_block_sptr top_block) override;
00065     gr::basic_block_sptr get_left_block() override;
00066     gr::basic_block_sptr get_right_block() override;
00067
00068 private:
00069     const std::string default_gain_mode = std::string("slow_attack");
00070 #if GR_IIO_TEMPLATIZED_API
00071     gr::iio::fmcomms2_source<gr_complex>::sptr plutosdr_source_;
00072 #else

```

```

00073     gr::io::pluto_source::sptr plutosdr_source_;
00074 #endif
00075
00076     gnss_shared_ptr<gr::block> valve_;
00077     gr::blocks::file_sink::sptr file_sink_;
00078
00079     std::string dump_filename_;
00080
00081     // Front-end settings
00082     std::string uri_; // device direction
00083     std::string gain_mode_;
00084     std::string filter_file_;
00085     std::string filter_source_;
00086     std::string filter_filename_;
00087     std::string item_type_;
00088     double rf_gain_;
00089     int64_t samples_;
00090     uint64_t freq_; // frequency of local oscillator
00091     uint64_t sample_rate_;
00092     uint64_t bandwidth_;
00093     uint64_t buffer_size_; // reception buffer
00094     size_t item_size_;
00095     float Fpass_;
00096     float Fstop_;
00097     unsigned int in_stream_;
00098     unsigned int out_stream_;
00099
00100     bool quadrature_;
00101     bool rf_dc_;
00102     bool bb_dc_;
00103     bool filter_auto_;
00104     bool dump_;
00105 };
00106
00107
00108 /** \} */
00109 /** \} */
00110 #endif // GNSS_SDR_PLUTOSDR_SIGNAL_SOURCE_H

```

## 11.341 raw\_array\_signal\_source.h File Reference

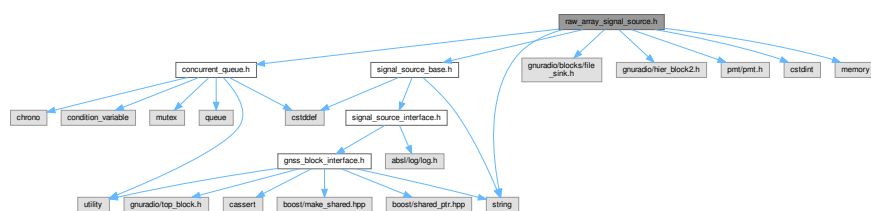
CTTC Experimental GNSS 8 channels array signal source.

```

#include "concurrent_queue.h"
#include "signal_source_base.h"
#include <gnuradio/blocks/file_sink.h>
#include <gnuradio/hier_block2.h>
#include <pmt/pmt.h>
#include <cstdlib>
#include <memory>
#include <string>

```

Include dependency graph for raw\_array\_signal\_source.h:



### Classes

- class [RawArraySignalSource](#)

*This class reads samples from an antenna array RF front-end signal sampler.*

### 11.341.1 Detailed Description

CTTC Experimental GNSS 8 channels array signal source.

## Author

Javier Arribas, jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [raw\\_array\\_signal\\_source.h](#).

---

## 11.342 raw\_array\_signal\_source.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file raw_array_signal_source.h
00003  * \brief CTTC Experimental GNSS 8 channels array signal source
00004  * \author Javier Arribas, jarribas(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_RAW_ARRAY_SIGNAL_SOURCE_H
00019 #define GNSS_SDR_RAW_ARRAY_SIGNAL_SOURCE_H
00020
00021 #include "concurrent_queue.h"
00022 #include "signal_source_base.h"
00023 #include <gnuradio/blocks/file_sink.h>
00024 #include <gnuradio/hier_block2.h>
00025 #include <pmt/pmt.h>
00026 #include <cstdint>
00027 #include <memory>
00028 #include <string>
00029
00030
00031 /** \addtogroup Signal_Source
00032  * \{ */
00033 /** \addtogroup Signal_Source_adapters
00034  * \{ */
00035
00036
00037 class ConfigurationInterface;
00038
00039 /*!
00040 * \brief This class reads samples from an antenna array RF front-end signal sampler
00041 */
00042 class RawArraySignalSource : public SignalSourceBase
00043 {
00044 public:
00045     RawArraySignalSource(const ConfigurationInterface* configuration,
00046                          std::string role, unsigned int in_stream,
00047                          unsigned int out_stream, Concurrent_Queue<pmt::pmt_t>* queue);
00048
00049     ~RawArraySignalSource() = default;
00050
00051     inline size_t item_size() override
00052     {
00053         return item_size_;
00054     }
00055
00056     void connect(gr::top_block_sptr top_block) override;
00057     void disconnect(gr::top_block_sptr top_block) override;
00058     gr::basic_block_sptr get_left_block() override;
00059     gr::basic_block_sptr get_right_block() override;
00060
00061 private:
00062     gr::block_sptr raw_array_source_;
00063     gr::blocks::file_sink::sptr file_sink_;
00064     std::string item_type_;
00065     std::string dump_filename_;
00066     std::string eth_device_;
00067     size_t item_size_;
00068     [[maybe_unused]] int64_t samples_;
00069     unsigned int in_stream_;
00070     unsigned int out_stream_;
00071     bool dump_;
00072 };

```



```

00073
00074
00075 /** \} */
00076 /** \} */
00077 #endif // GNSS_SDR_RAW_ARRAY_SIGNAL_SOURCE_H

```

## 11.343 rtl\_tcp\_signal\_source.h File Reference

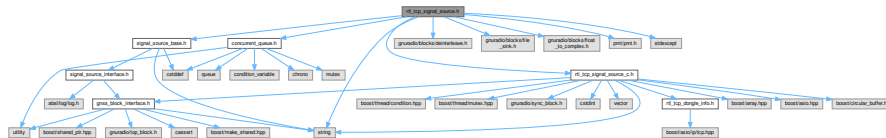
Signal source which reads from rtl\_tcp. (see <https://osmocom.org/projects/rtl-sdr/wiki> for more information)

```

#include "concurrent_queue.h"
#include "rtl_tcp_signal_source_c.h"
#include "signal_source_base.h"
#include <gnuradio/blocks/deinterleave.h>
#include <gnuradio/blocks/file_sink.h>
#include <gnuradio/blocks/float_to_complex.h>
#include <pmt/pmt.h>
#include <stdexcept>
#include <string>

```

Include dependency graph for rtl\_tcp\_signal\_source.h:



### Classes

- class [RtlTcpSignalSource](#)

*This class reads from rtl\_tcp, which streams interleaved I/Q samples over TCP. (see <https://osmocom.org/projects/rtl-sdr/wiki>)*

### 11.343.1 Detailed Description

Signal source which reads from rtl\_tcp. (see <https://osmocom.org/projects/rtl-sdr/wiki> for more information)

#### Author

Anthony Arnold, 2015. [anthony.arnold@uqconnect.edu.au](mailto:anthony.arnold@uqconnect.edu.au)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [rtl\\_tcp\\_signal\\_source.h](#).

---

## 11.344 rtl\_tcp\_signal\_source.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002 * \file rtl_tcp_signal_source.h
00003 * \brief Signal source which reads from rtl_tcp.
00004 * (see https://osmocom.org/projects/rtl-sdr/wiki for more information)
00005 * \author Anthony Arnold, 2015. anthony.arnold@uqconnect.edu.au
00006 *
00007 * -----
00008 *
00009 * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010 * This file is part of GNSS-SDR.
00011 *
00012 * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013 * SPDX-License-Identifier: GPL-3.0-or-later
00014 *

```

```

00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_RTL_TCP_SIGNAL_SOURCE_H
00019 #define GNSS_SDR_RTL_TCP_SIGNAL_SOURCE_H
00020
00021 #include "concurrent_queue.h"
00022 #include "rtl_tcp_signal_source_c.h"
00023 #include "signal_source_base.h"
00024 #include <gnuradio/blocks/deinterleave.h>
00025 #include <gnuradio/blocks/file_sink.h>
00026 #include <gnuradio/blocks/float_to_complex.h>
00027 #include <pmt/pmt.h>
00028 #include <stdexcept>
00029 #include <string>
00030
00031
00032 /** \addtogroup Signal_Source
00033  * \{ */
00034 /** \addtogroup Signal_Source_adapters
00035  * \{ */
00036
00037
00038 class ConfigurationInterface;
00039
00040 /*!
00041 * \brief This class reads from rtl_tcp, which streams interleaved
00042 * I/Q samples over TCP.
00043 * (see https://osmocom.org/projects/rtl-sdr/wiki)
00044 */
00045 class RtlTcpSignalSource : public SignalSourceBase
00046 {
00047 public:
00048     RtlTcpSignalSource(const ConfigurationInterface* configuration,
00049         const std::string& role,
00050         unsigned int in_stream,
00051         unsigned int out_stream,
00052         Concurrent_Queue<pmt::pmt_t>* queue);
00053
00054     ~RtlTcpSignalSource() = default;
00055
00056     inline size_t item_size() override
00057     {
00058         return item_size_;
00059     }
00060
00061     void connect(gr::top_block_sptr top_block) override;
00062     void disconnect(gr::top_block_sptr top_block) override;
00063     gr::basic_block_sptr get_left_block() override;
00064     gr::basic_block_sptr get_right_block() override;
00065
00066 private:
00067     void MakeBlock();
00068
00069     rtl_tcp_signal_source_c_sptr signal_source_;
00070
00071     gnss_shared_ptr<gr::block> valve_;
00072     gr::blocks::file_sink::sptr file_sink_;
00073
00074     std::string item_type_;
00075     std::string dump_filename_;
00076
00077     // rtl_tcp settings
00078     std::string address_;
00079     size_t item_size_;
00080     uint64_t samples_;
00081     double rf_gain_;
00082     int sample_rate_;
00083     int freq_;
00084     int gain_;
00085     int if_gain_;
00086     unsigned int in_stream_;
00087     unsigned int out_stream_;
00088     int16_t port_;
00089     bool AGC_enabled_;
00090     bool flip_iq_;
00091     bool dump_;
00092 };
00093
00094
00095 /** \} */
00096 /** \} */
00097 #endif // GNSS_SDR_RTL_TCP_SIGNAL_SOURCE_H

```

## 11.345 signal\_source\_base.h File Reference

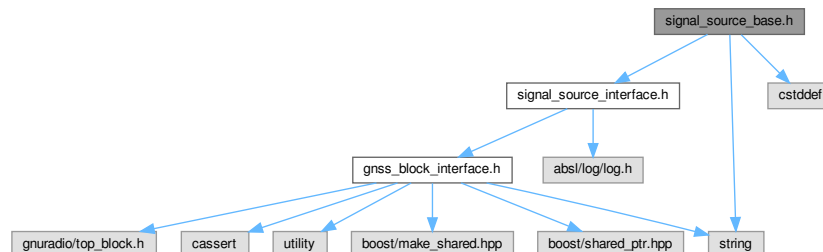
Header file of the base class to signal\_source GNSS blocks.

```
#include "signal_source_interface.h"
```

```
#include <cstdint>
```

```
#include <string>
```

Include dependency graph for signal\_source\_base.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [SignalSourceBase](#)

### 11.345.1 Detailed Description

Header file of the base class to signal\_source GNSS blocks.

#### Author

Jim Melton, 2020. [jim.melton@snrcorp.com](mailto:jim.melton@snrcorp.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [signal\\_source\\_base.h](#).

---

## 11.346 signal\_source\_base.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file signal_source_base.h
00003  * \brief Header file of the base class to signal_source GNSS blocks.
00004  * \author Jim Melton, 2020. jim.melton@snrcorp.com
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_SIGNAL_SOURCE_BASE_H
00019 #define GNSS_SDR_SIGNAL_SOURCE_BASE_H

```

```

00020
00021 #include "signal_source_interface.h"
00022 #include <cstdlib>
00023 #include <string>
00024
00025
00026 class ConfigurationInterface;
00027
00028 class SignalSourceBase : public SignalSourceInterface
00029 {
00030 public:
00031     std::string role() final;
00032     std::string implementation() final;
00033
00034     size_t getRfChannels() const override;
00035     gr::basic_block_sptr get_left_block() override; // non-sensical; implement once
00036
00037 protected:
00038     /// Constructor
00039     SignalSourceBase(ConfigurationInterface const* configuration, std::string role, std::string impl);
00040
00041     /// utility for decoding passed ".item_type" values
00042     /// @param[in] item_type - user provided string, should be one of the known types
00043     /// @param[out] is_interleaved - if non-null, the pointed to memory is updated with
00044     ///                               whether the data is interleaved I/Q (e.g., ishort)
00045     /// @param[in] throw_on_error - if true, throw an exception if the string does not
00046     ///                               represent a known type
00047     /// @return the size in bytes of the passed type
00048     size_t decode_item_type(std::string const& item_type, bool* is_interleaved = nullptr, bool
        throw_on_error = false);
00049
00050 private:
00051     std::string const role_;
00052     std::string const implementation_;
00053     size_t rfChannels_;
00054 };
00055
00056
00057 #endif

```

## 11.347 spir\_file\_signal\_source.h File Reference

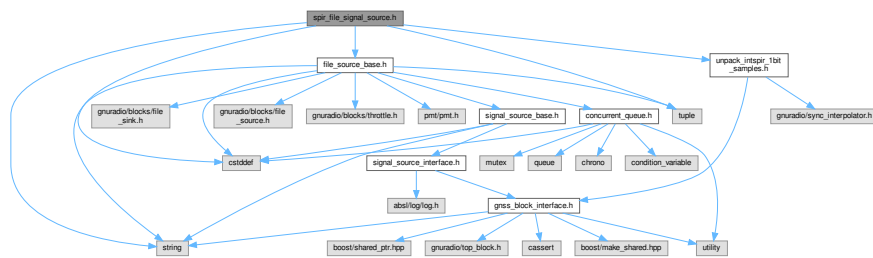
Implementation of a class that reads signals samples from a SPIR file and adapts it to a [SignalSourceInterface](#).

```

#include "file_source_base.h"
#include "unpack_intspir_1bit_samples.h"
#include <cstdlib>
#include <string>
#include <tuple>

```

Include dependency graph for spir\_file\_signal\_source.h:



### Classes

- class [SpirFileSignalSource](#)

*Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).*

### 11.347.1 Detailed Description

Implementation of a class that reads signals samples from a SPIR file and adapts it to a [SignalSourceInterface](#).

## Author

Fran Fabra, 2014 fabra(at)ice.csic.es

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [spir\\_file\\_signal\\_source.h](#).

## 11.348 spir\_file\_signal\_source.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file spir_file_signal_source.h
00003  * \brief Implementation of a class that reads signals samples from a SPIR file
00004  * and adapts it to a SignalSourceInterface.
00005  * \author Fran Fabra, 2014 fabra(at)ice.csic.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_SPIR_FILE_SIGNAL_SOURCE_H
00019 #define GNSS_SDR_SPIR_FILE_SIGNAL_SOURCE_H
00020
00021 #include "file_source_base.h"
00022 #include "unpack_intspir_lbit_samples.h"
00023 #include <cstdlib>
00024 #include <string>
00025 #include <tuple>
00026
00027
00028 /** \addtogroup Signal_Source
00029  * \{ */
00030 /** \addtogroup Signal_Source_adapters
00031  * \{ */
00032
00033
00034 class ConfigurationInterface;
00035
00036 /*!
00037  * \brief Class that reads signals samples from a file
00038  * and adapts it to a SignalSourceInterface
00039  */
00040 class SpirFileSignalSource : public FileSourceBase
00041 {
00042 public:
00043     SpirFileSignalSource(const ConfigurationInterface* configuration, const std::string& role,
00044         unsigned int in_streams, unsigned int out_streams,
00045         Concurrent_Queue<pmt::pmt_t>* queue);
00046
00047     ~SpirFileSignalSource() = default;
00048
00049 protected:
00050     std::tuple<size_t, bool> itemTypeToSize() override;
00051     gnss_shared_ptr<gr::block> source() const override;
00052     void create_file_source_hook() override;
00053     void pre_connect_hook(gr::top_block_sptr top_block) override;
00054     void post_disconnect_hook(gr::top_block_sptr top_block) override;
00055
00056 private:
00057     unpack_intspir_lbit_samples_sptr unpack_intspir_;
00058 };
00059
00060
00061
00062 /** \} */
00063 /** \} */
00064 #endif // GNSS_SDR_SPIR_FILE_SIGNAL_SOURCE_H
```

## 11.349 spir\_gss6450\_file\_signal\_source.h File Reference

Implementation of a class that reads signals samples from a SPIR file and adapts it to a [SignalSourceInterface](#).

```
#include "concurrent_queue.h"
#include "gnss_sdr_valve.h"
#include "signal_source_base.h"
#include "unpack_spir_gss6450_samples.h"
#include <gnuradio/blocks/deinterleave.h>
#include <gnuradio/blocks/endian_swap.h>
#include <gnuradio/blocks/file_sink.h>
#include <gnuradio/blocks/file_source.h>
#include <gnuradio/blocks/null_sink.h>
#include <gnuradio/blocks/throttle.h>
#include <gnuradio/hier_block2.h>
#include <pmt/pmt.h>
#include <cstdint>
#include <string>
#include <vector>
```

Include dependency graph for spir\_gss6450\_file\_signal\_source.h:



## Classes

- class [SpirGSS6450FileSignalSource](#)

*Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).*

## 11.349.1 Detailed Description

Implementation of a class that reads signals samples from a SPIR file and adapts it to a [SignalSourceInterface](#).

### Author

Antonio Ramos, 2017 antonio.ramos(at)cttc.es

---

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)

GNSS-SDR is a software defined Global Navigation Satellite Systems receiver

This file is not part of GNSS-SDR.

SPDX-License-Identifier: GPL-3.0-or-later

---

Definition in file [spir\\_gss6450\\_file\\_signal\\_source.h](#).

## 11.350 spir\_gss6450\_file\_signal\_source.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file spir_gss6450_file_signal_source.h
00003  * \brief Implementation of a class that reads signals samples from a SPIR file
00004  * and adapts it to a SignalSourceInterface.
00005  * \author Antonio Ramos, 2017 antonio.ramos(at)cttc.es
00006  *
00007  * -----
00008  *
00009  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00010  *
00011  * GNSS-SDR is a software defined Global Navigation
00012  * Satellite Systems receiver
00013  *
00014  * This file is not part of GNSS-SDR.
00015  *
00016  * SPDX-License-Identifier: GPL-3.0-or-later
00017  *
00018  * -----
00019  */
00020
```

```

00021 #ifndef GNSS_SDR_SPIR_GSS6450_FILE_SIGNAL_SOURCE_H
00022 #define GNSS_SDR_SPIR_GSS6450_FILE_SIGNAL_SOURCE_H
00023
00024 #include "concurrent_queue.h"
00025 #include "gnss_sdr_valve.h"
00026 #include "signal_source_base.h"
00027 #include "unpack_spir_gss6450_samples.h"
00028 #include <gnuradio/blocks/deinterleave.h>
00029 #include <gnuradio/blocks/endian_swap.h>
00030 #include <gnuradio/blocks/file_sink.h>
00031 #include <gnuradio/blocks/file_source.h>
00032 #include <gnuradio/blocks/null_sink.h>
00033 #include <gnuradio/blocks/throttle.h>
00034 #include <gnuradio/hier_block2.h>
00035 #include <pmt/pmt.h>
00036 #include <stdint>
00037 #include <string>
00038 #include <vector>
00039
00040
00041 /** \addtogroup Signal_Source
00042  * \{ */
00043 /** \addtogroup Signal_Source_adapters
00044  * \{ */
00045
00046
00047 class ConfigurationInterface;
00048
00049 /*!
00050  * \brief Class that reads signals samples from a file
00051  * and adapts it to a SignalSourceInterface
00052  */
00053 class SpirGSS6450FileSignalSource : public SignalSourceBase
00054 {
00055 public:
00056     SpirGSS6450FileSignalSource(const ConfigurationInterface* configuration, const std::string& role,
00057         uint32_t in_streams, uint32_t out_streams, Concurrent_Queue<pmt::pmt_t*> queue);
00058
00059     inline size_t item_size() override
00060     {
00061         return item_size_;
00062     }
00063
00064     void connect(gr::top_block_sptr top_block) override;
00065     void disconnect(gr::top_block_sptr top_block) override;
00066     gr::basic_block_sptr get_left_block() override;
00067     gr::basic_block_sptr get_right_block(int RF_channel) override;
00068     gr::basic_block_sptr get_right_block() override;
00069
00070     inline std::string filename() const
00071     {
00072         return filename_;
00073     }
00074
00075     inline std::string item_type() const
00076     {
00077         return item_type_;
00078     }
00079
00080     inline bool repeat() const
00081     {
00082         return repeat_;
00083     }
00084
00085     inline int64_t sampling_frequency() const
00086     {
00087         return sampling_frequency_;
00088     }
00089
00090     inline uint64_t samples() const
00091     {
00092         return samples_;
00093     }
00094
00095 private:
00096     gr::blocks::file_source::sptr file_source_;
00097     gr::blocks::deinterleave::sptr deint_;
00098     std::vector<gnss_shared_ptr<gr::block>> valve_vec_;
00099     std::vector<gr::blocks::endian_swap::sptr> endian_vec_;
00100     std::vector<gr::blocks::null_sink::sptr> null_sinks_;
00101     std::vector<unpack_spir_gss6450_samples_sptr> unpack_spir_vec_;
00102     std::vector<gr::blocks::file_sink::sptr> sink_vec_;
00103     std::vector<gr::blocks::throttle::sptr> throttle_vec_;
00104     std::string filename_;
00105     std::string dump_filename_;
00106     std::string item_type_;
00107     uint64_t samples_;

```

```

00108     int64_t sampling_frequency_;
00109     size_t item_size_;
00110     uint32_t in_streams_;
00111     uint32_t out_streams_;
00112     uint32_t adc_bits_;
00113     int32_t n_channels_;
00114     int32_t sel_ch_;
00115     bool repeat_;
00116     bool dump_; // Enables dumping the gr_complex sample output
00117     bool enable_throttle_control_;
00118     bool endian_swap_;
00119 };
00120
00121
00122 /** \} */
00123 /** \} */
00124 #endif // GNSS_SDR_SPIR_GSS6450_FILE_SIGNAL_SOURCE_H

```

## 11.351 two\_bit\_cpx\_file\_signal\_source.h File Reference

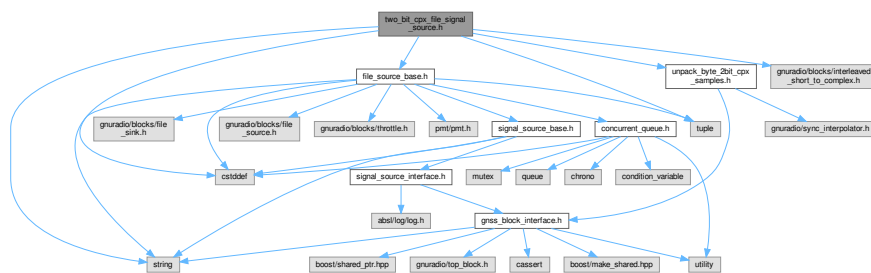
Interface of a class that reads signals samples from a 2 bit complex sampler front-end file and adapts it to a [SignalSourceInterface](#).

```

#include "file_source_base.h"
#include "unpack_byte_2bit_cpx_samples.h"
#include <gnuradio/blocks/interleaved_short_to_complex.h>
#include <csddef>
#include <string>
#include <tuple>

```

Include dependency graph for two\_bit\_cpx\_file\_signal\_source.h:



### Classes

- class [TwoBitCpxFileSignalSource](#)

*Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).*

### 11.351.1 Detailed Description

Interface of a class that reads signals samples from a 2 bit complex sampler front-end file and adapts it to a [SignalSourceInterface](#).

#### Author

Javier Arribas, 2015 jarribas(at)cttc.es

This class represents a file signal source.

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [two\\_bit\\_cpx\\_file\\_signal\\_source.h](#).

## 11.352 two\_bit\_cpx\_file\_signal\_source.h

[Go to the documentation of this file.](#)



```

00001  /*!
00002  * \file two_bit_cpx_file_signal_source.h
00003  * \brief Interface of a class that reads signals samples from a 2 bit complex sampler front-end file
00004  * and adapts it to a SignalSourceInterface.
00005  * \author Javier Arribas, 2015 jarribas(at)cttc.es
00006  *
00007  * This class represents a file signal source.
00008  *
00009  * -----
00010  *
00011  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00012  * This file is part of GNSS-SDR.
00013  *
00014  * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)
00015  * SPDX-License-Identifier: GPL-3.0-or-later
00016  *
00017  * -----
00018  */
00019
00020 #ifndef GNSS_SDR_TWO_BIT_CPX_FILE_SIGNAL_SOURCE_H
00021 #define GNSS_SDR_TWO_BIT_CPX_FILE_SIGNAL_SOURCE_H
00022
00023 #include "file_source_base.h"
00024 #include "unpack_byte_2bit_cpx_samples.h"
00025 #include <gnuradio/blocks/interleaved_short_to_complex.h>
00026 #include <cstdint>
00027 #include <string>
00028 #include <tuple>
00029
00030 /** \addtogroup Signal_Source
00031  * \{ */
00032 /** \addtogroup Signal_Source_adapters
00033  * \{ */
00034
00035
00036 class ConfigurationInterface;
00037
00038 /*!
00039 * \brief Class that reads signals samples from a file
00040 * and adapts it to a SignalSourceInterface
00041 */
00042 class TwoBitCpxFileSignalSource : public FileSourceBase
00043 {
00044 public:
00045     TwoBitCpxFileSignalSource(const ConfigurationInterface* configuration,
00046                             const std::string& role,
00047                             unsigned int in_streams,
00048                             unsigned int out_streams,
00049                             Concurrent_Queue<pmt::pmt_t>* queue);
00050
00051     ~TwoBitCpxFileSignalSource() = default;
00052
00053 protected:
00054     std::tuple<size_t, bool> itemTypeToSize() override;
00055     double packetsPerSample() const override;
00056     gnss_shared_ptr<gr::block> source() const override;
00057     void create_file_source_hook() override;
00058     void pre_connect_hook(gr::top_block_sptr top_block) override;
00059     void pre_disconnect_hook(gr::top_block_sptr top_block) override;
00060
00061 private:
00062     unpack_byte_2bit_cpx_samples_sptr unpack_byte_;
00063     gr::blocks::interleaved_short_to_complex::sptr inter_shorts_to_cpx_;
00064 };
00065
00066
00067 /** \} */
00068 /** \} */
00069 #endif // GNSS_SDR_TWO_BIT_CPX_FILE_SIGNAL_SOURCE_H

```

## 11.353 two\_bit\_packed\_file\_signal\_source.h File Reference

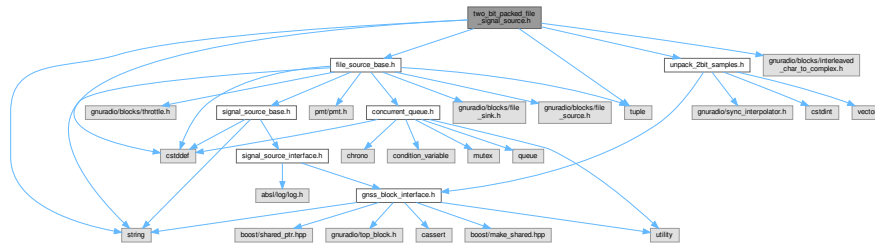
Interface of a class that reads signals samples from a file. Each sample is two bits, which are packed into bytes or shorts.

```

#include "file_source_base.h"
#include "unpack_2bit_samples.h"
#include <gnuradio/blocks/interleaved_char_to_complex.h>
#include <cstdint>
#include <string>
#include <tuple>

```

Include dependency graph for `two_bit_packed_file_signal_source.h`:



## Classes

- class [TwoBitPackedFileSignalSource](#)

*Class that reads signals samples from a file and adapts it to a [SignalSourceInterface](#).*

### 11.353.1 Detailed Description

Interface of a class that reads signals samples from a file. Each sample is two bits, which are packed into bytes or shorts.

#### Author

Cillian O'Driscoll, 2015 cillian.odriscoll (at) gmail.com

This class represents a file signal source.

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [two\\_bit\\_packed\\_file\\_signal\\_source.h](#).

## 11.354 two\_bit\_packed\_file\_signal\_source.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file two_bit_packed_file_signal_source.h
00003  * \brief Interface of a class that reads signals samples from a file. Each
00004  * sample is two bits, which are packed into bytes or shorts.
00005  *
00006  * \author Cillian O'Driscoll, 2015 cillian.odriscoll (at) gmail.com
00007  *
00008  * This class represents a file signal source.
00009  *
00010  * -----
00011  *
00012  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00013  * This file is part of GNSS-SDR.
00014  *
00015  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00016  * SPDX-License-Identifier: GPL-3.0-or-later
00017  *
00018  * -----
00019  */
00020
00021 #ifndef GNSS_SDR_TWO_BIT_PACKED_FILE_SIGNAL_SOURCE_H
00022 #define GNSS_SDR_TWO_BIT_PACKED_FILE_SIGNAL_SOURCE_H
00023
00024 #include "file_source_base.h"
00025 #include "unpack_2bit_samples.h"
00026 #include <gnuradio/blocks/interleaved_char_to_complex.h>
00027 #include <cstdlib>
00028 #include <string>
00029 #include <tuple>
00030
00031
00032 /** \addtogroup Signal_Source
00033  * \{ */
00034 /** \addtogroup Signal_Source_adapters
00035  * \{ */
00036
```

```

00037
00038 class ConfigurationInterface;
00039
00040 /*!
00041 * \brief Class that reads signals samples from a file
00042 * and adapts it to a SignalSourceInterface
00043 */
00044 class TwoBitPackedFileSignalSource : public FileSourceBase
00045 {
00046 public:
00047     TwoBitPackedFileSignalSource(const ConfigurationInterface* configuration, const std::string& role,
00048         unsigned int in_streams, unsigned int out_streams,
00049         Concurrent_Queue<pmt::pmt_t*> queue);
00050
00051     ~TwoBitPackedFileSignalSource() = default;
00052
00053 private:
00054     inline bool big_endian_items() const
00055     {
00056         return big_endian_items_;
00057     }
00058
00059     inline bool big_endian_bytes() const
00060     {
00061         return big_endian_bytes_;
00062     }
00063
00064
00065     inline bool reverse_interleaving() const
00066     {
00067         return reverse_interleaving_;
00068     }
00069
00070 protected:
00071     std::tuple<size_t, bool> itemTypeToSize() override;
00072     double packetsPerSample() const override;
00073     gnss_shared_ptr<gr::block> source() const override;
00074     void create_file_source_hook() override;
00075     void pre_connect_hook(gr::top_block_sptr top_block) override;
00076     void pre_disconnect_hook(gr::top_block_sptr top_block) override;
00077
00078 private:
00079     std::string sample_type_;
00080     bool big_endian_items_;
00081     bool big_endian_bytes_;
00082     bool reverse_interleaving_;
00083     unpack_2bit_samples_sptr unpack_samples_;
00084     gnss_shared_ptr<gr::block> char_to_float_;
00085 };
00086
00087
00088 /** \} */
00089 /** \} */
00090 #endif // GNSS_SDR_TWO_BIT_CPX_FILE_SIGNAL_SOURCE_H

```

## 11.355 uhd\_signal\_source.h File Reference

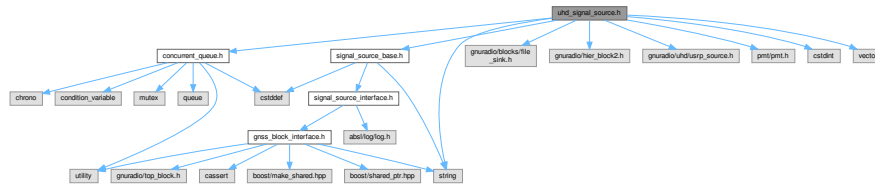
Interface for the Universal Hardware Driver signal source.

```

#include "concurrent_queue.h"
#include "signal_source_base.h"
#include <gnuradio/blocks/file_sink.h>
#include <gnuradio/hier_block2.h>
#include <gnuradio/uhd/usrp_source.h>
#include <pmt/pmt.h>
#include <cstdint>
#include <string>
#include <vector>

```

Include dependency graph for uhd\_signal\_source.h:



## Classes

- class [UhdSignalSource](#)

*This class reads samples from a UHD device (see <http://code.ettus.com/redmine/ettus/projects/uhd/wiki>)*

### 11.355.1 Detailed Description

Interface for the Universal Hardware Driver signal source.

#### Author

Javier Arribas, 2012. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [uhd\\_signal\\_source.h](#).

---

## 11.356 uhd\_signal\_source.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file uhd_signal_source.h
00003  * \brief Interface for the Universal Hardware Driver signal source
00004  * \author Javier Arribas, 2012. jarribas(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_UHD_SIGNAL_SOURCE_H
00018 #define GNSS_SDR_UHD_SIGNAL_SOURCE_H
00019
00020 #include "concurrent_queue.h"
00021 #include "signal_source_base.h"
00022 #include <gnuradio/blocks/file_sink.h>
00023 #include <gnuradio/hier_block2.h>
00024 #include <gnuradio/uhd/usrp_source.h>
00025 #include <pmt/pmt.h>
00026 #include <csdint>
00027 #include <string>
00028 #include <vector>
00029
00030
00031 /** \addtogroup Signal_Source
00032  * \{ */
00033 /** \addtogroup Signal_Source_adapters
00034  * \{ */
00035
00036 class ConfigurationInterface;
00037
00038 /*!
00039 * \brief This class reads samples from a UHD device (see
00040 * http://code.ettus.com/redmine/ettus/projects/uhd/wiki)
00041 */

```

```

00041 class UhdSignalSource : public SignalSourceBase
00042 {
00043 public:
00044     UhdSignalSource(const ConfigurationInterface* configuration,
00045                     const std::string& role, unsigned int in_stream,
00046                     unsigned int out_stream, Concurrent_Queue<pmt::pmt_t*> queue);
00047
00048     ~UhdSignalSource() = default;
00049
00050     inline size_t item_size() override
00051     {
00052         return item_size_;
00053     }
00054
00055     void connect(gr::top_block_sptr top_block) override;
00056     void disconnect(gr::top_block_sptr top_block) override;
00057     gr::basic_block_sptr get_left_block() override;
00058     gr::basic_block_sptr get_right_block() override;
00059     gr::basic_block_sptr get_right_block(int RF_channel) override;
00060
00061 private:
00062     gr::uhd::usrp_source::sptr uhd_source_;
00063
00064     std::vector<gnss_shared_ptr<gr::block>> valve_;
00065     std::vector<gr::blocks::file_sink::sptr> file_sink_;
00066     std::vector<double> freq_;
00067     std::vector<double> gain_;
00068     std::vector<double> IF_bandwidth_hz_;
00069     std::vector<uint64_t> samples_;
00070     std::vector<std::string> dump_filename_;
00071     std::vector<bool> dump_;
00072
00073     uhd::stream_args_t uhd_stream_args_; // UHD SETTINGS
00074
00075     std::string device_address_;
00076     std::string item_type_;
00077     std::string subdevice_;
00078     std::string clock_source_;
00079
00080     // * The OTW format is a string that describes the format over-the-wire.
00081     // * The following over-the-wire formats have been implemented:
00082     // * - sc16 - Q16 I16
00083     // * - sc8 - Q8_1 I8_1 Q8_0 I8_0
00084     // * - sc12 (Only some devices)
00085     std::string otw_format_;
00086
00087     double sample_rate_;
00088     size_t item_size_;
00089     int RF_channels_;
00090     unsigned int in_stream_;
00091     unsigned int out_stream_;
00092 };
00093
00094
00095 /** \} */
00096 /** \} */
00097 #endif // GNSS_SDR_UHD_SIGNAL_SOURCE_H

```

## 11.357 zmq\_signal\_source.h File Reference

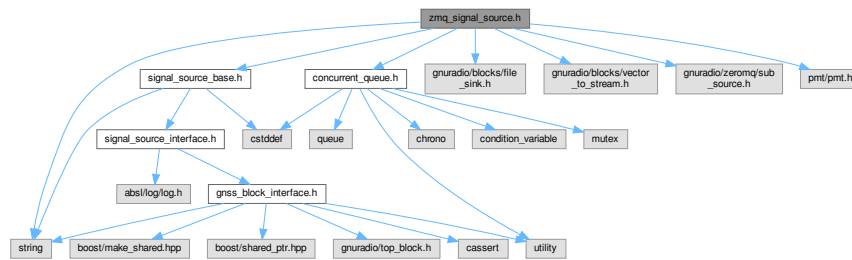
Signal source which reads from ZeroMQ.

```

#include "signal_source_base.h"
#include "concurrent_queue.h"
#include <gnuradio/blocks/file_sink.h>
#include <gnuradio/blocks/vector_to_stream.h>
#include <gnuradio/zeromq/sub_source.h>
#include <pmt/pmt.h>
#include <string>

```

Include dependency graph for `zmq_signal_source.h`:



## Classes

- class [ZmqSignalSource](#)

*This class supports the following properties:*

### 11.357.1 Detailed Description

Signal source which reads from ZeroMQ.

#### Author

Jim Melton, 2022. [jim.melton@snrcorp.com](mailto:jim.melton@snrcorp.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [zmq\\_signal\\_source.h](#).

---

## 11.358 zmq\_signal\_source.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file zmq_signal_source.h
00003  * \brief Signal source which reads from ZeroMQ.
00004  * \author Jim Melton, 2022. jim.melton@snrcorp.com
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_ZMQ_SIGNAL_SOURCE_H
00018 #define GNSS_SDR_ZMQ_SIGNAL_SOURCE_H
00019
00020 #include "signal_source_base.h"
00021 //
00022 #include "concurrent_queue.h"
00023 #include <gnuradio/blocks/file_sink.h> // for dump
00024 #include <gnuradio/blocks/vector_to_stream.h>
00025 #include <gnuradio/zeromq/sub_source.h>
00026 #include <pmt/pmt.h>
00027 #include <string>
00028
00029 /** \addtogroup Signal_Source
00030  * \{ */
00031 /** \addtogroup Signal_Source_adapters
00032  * \{ */
00033
00034 //! This class supports the following properties:
00035 //!
00036 //! .endpoint - the ZMQ endpoint to be connected to
00037 //! .vlen - vector length of the input items (default 1, one item)
  
```

```

00038 //!           this must match the size of the publisher!
00039 //!   .pass_tags - boolean flag if tags should be propagated (default false)
00040 //!   .timeout_ms - receive timeout, in milliseconds (default 100)
00041 //!   .hwm       - ZMQ high water mark (default -1, ZMQ default)
00042 //!
00043 //!   .item_type - data type of the samples (default "gr_complex")
00044 //!
00045 //! (probably should be abstracted to the base class)
00046 //!
00047 //!   .dump       - whether to archive input data
00048 //!   .dump_filename - if dumping, path to file for output
00049 //!
00050
00051 class ZmqSignalSource : public SignalSourceBase
00052 {
00053 public:
00054     ZmqSignalSource(const ConfigurationInterface* configuration,
00055         const std::string& role,
00056         unsigned int in_stream,
00057         unsigned int out_stream,
00058         Concurrent_Queue<pmt::pmt_t>* queue);
00059
00060     ~ZmqSignalSource() = default;
00061
00062     size_t item_size() override;
00063
00064     auto connect(gr::top_block_sptr top_block) -> void override;
00065     auto disconnect(gr::top_block_sptr top_block) -> void override;
00066     auto get_right_block() -> gr::basic_block_sptr override;
00067
00068 private:
00069     gr::zeromq::sub_source::sptr d_source_block;
00070     gr::blocks::vector_to_stream::sptr d_vec_block;
00071     gr::blocks::file_sink::sptr d_dump_sink;
00072
00073     size_t d_item_size;
00074     std::string d_dump_filename;
00075     bool d_dump;
00076 };
00077
00078 /** \} */
00079 /** \} */
00080 #endif

```

## 11.359 ad936x\_iio\_source.h File Reference

A direct IIO custom front-end gnss-sdr signal gnuradio block for the AD936x AD front-end family with special FPGA custom functionalities.

```

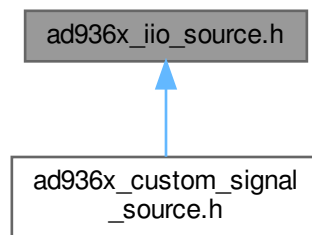
#include "ad936x_iio_custom.h"
#include "concurrent_queue.h"
#include "gnss_block_interface.h"
#include "ppstcprx.h"
#include <gnuradio/block.h>
#include <pmt/pmt.h>
#include <cstddef>
#include <cstdint>
#include <fstream>
#include <iostream>
#include <memory>
#include <string>
#include <thread>
#include <vector>

```

Include dependency graph for ad936x\_iio\_source.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [ad936x\\_iio\\_source](#)

*This class implements conversion between Labsat 2, 3 and 3 Wideband formats to gr\_complex.*

## Typedefs

- using [ad936x\\_iio\\_source\\_sptr](#) = gnss\_shared\_ptr<[ad936x\\_iio\\_source](#)>

## Functions

- `ad936x_iio_source_sptr ad936x_iio_make_source_sptr (const std::string &pluto_uri_, const std::string &board_type_, int64_t bandwidth_, int64_t sample_rate_, int64_t freq_, const std::string &rf_port_select_, const std::string &rf_filter, const std::string &gain_mode_rx0_, const std::string &gain_mode_rx1_, double rf_gain_rx0_, double rf_gain_rx1_, bool enable_ch0, bool enable_ch1, int64_t freq_2ch, bool ppsmode_, bool customsamplesize_, const std::string &fe_ip_, int fe_ctlport_, int ssize_, int bshift_, bool spattern_, double lo_attenuation_db_, bool high_side_lo_, int tx_lo_channel_)`

## 11.359.1 Detailed Description

A direct IIO custom front-end gnss-sdr signal gnuradio block for the AD936x AD front-end family with special FPGA custom functionalities.

### Author

Javier Arribas, jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [ad936x\\_iio\\_source.h](#).

---

## 11.360 ad936x\_iio\_source.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file ad936x_iio_source.h
00003  * \brief A direct IIO custom front-end gnss-sdr signal gnuradio block for the
00004  * AD936x AD front-end family with special FPGA custom functionalities.
00005  * \author Javier Arribas, jarribas(at)cttc.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *

```



```

00012  * Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018
00019 #ifndef GNSS_SDR_AD936X_IIO_SOURCE_H
00020 #define GNSS_SDR_AD936X_IIO_SOURCE_H
00021
00022 #include "ad936x_iio_custom.h"
00023 #include "concurrent_queue.h"
00024 #include "gnss_block_interface.h"
00025 #include "ppstcprx.h"
00026 #include <gnuradio/block.h>
00027 #include <pmt/pmt.h>
00028 #include <cstdlib>
00029 #include <stdint>
00030 #include <fstream>
00031 #include <iostream>
00032 #include <memory>
00033 #include <string>
00034 #include <thread>
00035 #include <vector>
00036
00037 /** \addtogroup Signal_Source
00038  * \{ */
00039 /** \addtogroup Signal_Source_gnuradio_blocks
00040  * \{ */
00041
00042
00043 class ad936x_iio_source;
00044
00045 using ad936x_iio_source_sptr = gnss_shared_ptr<ad936x_iio_source>;
00046
00047 ad936x_iio_source_sptr ad936x_iio_make_source_sptr(
00048     const std::string &pluto_uri_,
00049     const std::string &board_type_,
00050     int64_t bandwidth_,
00051     int64_t sample_rate_,
00052     int64_t freq_,
00053     const std::string &rfr_port_select_,
00054     const std::string &rfr_filter_,
00055     const std::string &gain_mode_rx0_,
00056     const std::string &gain_mode_rx1_,
00057     double rfr_gain_rx0_,
00058     double rfr_gain_rx1_,
00059     bool enable_ch0,
00060     bool enable_ch1,
00061     int64_t freq_2ch,
00062     bool ppsmode_,
00063     bool customsamplesize_,
00064     const std::string &fe_ip_,
00065     int fe_ctlport_,
00066     int ssize_,
00067     int bshift_,
00068     bool spattern_,
00069     double lo_attenuation_db_,
00070     bool high_side_lo_,
00071     int tx_lo_channel_);
00072
00073 /*!
00074  * \brief This class implements conversion between Labsat 2, 3 and 3 Wideband
00075  * formats to gr_complex
00076  */
00077 class ad936x_iio_source : public gr::block
00078 {
00079 public:
00080     ~ad936x_iio_source();
00081
00082     //! start the sample transmission
00083     bool start();
00084     //! stop the sample transmission
00085     bool stop();
00086
00087     int general_work(int noutput_items,
00088                     gr_vector_int &ninput_items,
00089                     gr_vector_const_void_star &input_items,
00090                     gr_vector_void_star &output_items);
00091
00092 private:
00093     friend ad936x_iio_source_sptr ad936x_iio_make_source_sptr(
00094         const std::string &pluto_uri_,
00095         const std::string &board_type_,
00096         int64_t bandwidth_,
00097         int64_t sample_rate_,
00098         int64_t freq_,

```

```

00099         const std::string &rf_port_select_,
00100         const std::string &rf_filter,
00101         const std::string &gain_mode_rx0_,
00102         const std::string &gain_mode_rx1_,
00103         double rf_gain_rx0_,
00104         double rf_gain_rx1_,
00105         bool enable_ch0,
00106         bool enable_ch1,
00107         int64_t freq_2ch,
00108         bool ppsmode_,
00109         bool customsamplesize_,
00110         const std::string &fe_ip_,
00111         int fe_ctlport_,
00112         int ssize_,
00113         int bshift_,
00114         bool spattern_,
00115         double lo_attenuation_db_,
00116         bool high_side_lo_,
00117         int tx_lo_channel_);
00118
00119     ad936x_iio_source(
00120         const std::string &pluto_uri_,
00121         const std::string &board_type_,
00122         int64_t bandwidth_,
00123         int64_t sample_rate_,
00124         int64_t freq_,
00125         const std::string &rf_port_select_,
00126         const std::string &rf_filter,
00127         const std::string &gain_mode_rx0_,
00128         const std::string &gain_mode_rx1_,
00129         double rf_gain_rx0_,
00130         double rf_gain_rx1_,
00131         bool enable_ch0,
00132         bool enable_ch1,
00133         int64_t freq_2ch,
00134         bool ppsmode_,
00135         bool customsamplesize_,
00136         const std::string &fe_ip_,
00137         int fe_ctlport_,
00138         int ssize_,
00139         int bshift_,
00140         bool spattern_,
00141         double lo_attenuation_db_,
00142         bool high_side_lo_,
00143         int tx_lo_channel_);
00144
00145     void ad9361_channel_demux_to_buffer(ad936x_iio_samples *samples_in, int nchannels,
gr_vector_void_star &output_items);
00146     void ad9361_channel_demux_and_record(ad936x_iio_samples *samples_in, int nchannels,
std::vector<std::fstream> *files_out);
00147
00148     std::thread pps_rx_thread;
00149     std::unique_ptr<ad936x_iio_custom> ad936x_custom;
00150     std::shared_ptr<pps_tcp_rx> pps_rx;
00151     std::shared_ptr<Concurrent_Queue<PpsSamplestamp>> ppsqueue;
00152
00153     std::vector<std::fstream> samplesfile;
00154 };
00155
00156
00157 /** \} */
00158 /** \} */
00159 #endif // GNSS_SDR_AD936X_IIO_SOURCE_H

```

## 11.361 fifo\_reader.h File Reference

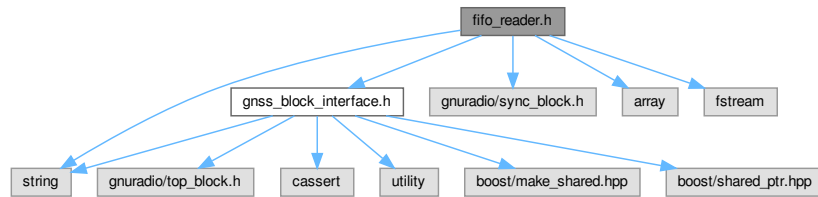
Header file to retrieve samples from an existing Unix FIFO.

```

#include "gnss_block_interface.h"
#include <gnuradio/sync_block.h>
#include <array>
#include <fstream>
#include <string>

```

Include dependency graph for fifo\_reader.h:



## Classes

- class [FifoReader](#)

### 11.361.1 Detailed Description

Header file to retrieve samples from an existing Unix FIFO.

#### Author

Malte Lenhart, 2021. [malte.lenhart\(at\)mailbox.org](mailto:malte.lenhart(at)mailbox.org)

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [fifo\\_reader.h](#).

## 11.362 fifo\_reader.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file fifo_reader.h
00003  *
00004  * \brief Header file to retrieve samples from an existing Unix FIFO
00005  * \author Malte Lenhart, 2021. malte.lenhart(at)mailbox.org
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_FIFO_READER_H_
00019 #define GNSS_SDR_FIFO_READER_H_
00020
00021 #include "gnss_block_interface.h"
00022 #include <gnuradio/sync_block.h>
00023 #include <array>
00024 #include <fstream> // std::ifstream
00025 #include <string>
00026
00027 /** \addtogroup Signal_Source
00028  * \{ */
00029 /** \addtogroup Signal_Source_gnuradio_blocks
00030  * \{ */
00031 class FifoReader : virtual public gr::sync_block
00032 {
00033 public:
00034     /** \brief static function to create a class instance
00035     using sptr = gnss_shared_ptr<FifoReader>;
00036     static sptr make(const std::string &file_name, const std::string &sample_type);
00037
00038     ~FifoReader() = default;
00039
00040     /** initialize istream resource for FIFO

```

```

00041     bool start();
00042
00043     // gnu radio work cycle function
00044     int work(int noutput_items,
00045             gr_vector_const_void_star &input_items,
00046             gr_vector_void_star &output_items);
00047
00048 private:
00049     /// \brief Constructor
00050     /// private constructor called by function make
00051     /// (gr handles this with public and private header pair)
00052     FifoReader(const std::string &file_name, const std::string &sample_type);
00053
00054     size_t read_gr_complex(int noutput_items, gr_vector_void_star &output_items);
00055
00056     /// function to read data out of FIFO which is stored as interleaved I/Q stream.
00057     /// template argument determines sample_type
00058     // Note: template definition necessary in header file
00059     // See also:
https://stackoverflow.com/questions/495021/why-can-templates-only-be-implemented-in-the-header-file
00060     template <typename Type>
00061     size_t read_interleaved(int noutput_items, gr_vector_void_star &output_items)
00062     {
00063         size_t items_retrieved = 0;
00064         for (int n = 0; n < noutput_items; n++)
00065         {
00066             // TODO: try if performance increases if we copy larger chunks to vector.
00067             // how to read from stream: https://en.cppreference.com/w/cpp/io/basic\_ifstream
00068             std::array<char, 2 * sizeof(Type)> buffer;
00069             fifo_.read(reinterpret_cast<char *>(buffer.data()), buffer.size());
00070             if (fifo_.good())
00071             {
00072                 auto real = reinterpret_cast<Type const *>(&buffer[0]);
00073                 auto imag = reinterpret_cast<Type const *>(&buffer[sizeof(Type)]);
00074                 static_cast<gr_complex *>(output_items[0])[n] = gr_complex(*real, *imag);
00075                 items_retrieved++;
00076             }
00077             else if (fifo_.eof())
00078             {
00079                 fifo_.clear();
00080                 break;
00081             }
00082             else
00083             {
00084                 fifo_error_output();
00085                 break;
00086             }
00087         }
00088         return items_retrieved;
00089     }
00090
00091     /// this function moves logging output from this header into the source file
00092     /// thereby eliminating the need to include glog/logging.h in this header
00093     void fifo_error_output() const;
00094
00095     const std::string file_name_;
00096     const std::string sample_type_;
00097     std::ifstream fifo_;
00098 };
00099
00100 /** \} */
00101 /** \} */
00102 #endif /* GNSS_SDR_FIFO_READER_H_ */

```

## 11.363 gr\_complex\_ip\_packet\_source.h File Reference

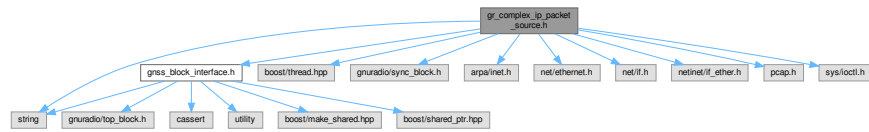
Receives ip frames containing samples in UDP frame encapsulation using a high performance packet capture library (libpcap)

```

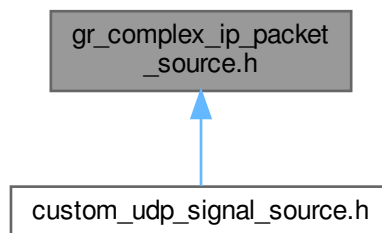
#include "gnss_block_interface.h"
#include <boost/thread.hpp>
#include <gnuradio/sync_block.h>
#include <arpa/inet.h>
#include <net/ethernet.h>
#include <net/if.h>
#include <netinet/if_ether.h>
#include <pcap.h>
#include <string>
#include <sys/ioctl.h>

```

Include dependency graph for gr\_complex\_ip\_packet\_source.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Gr\\_Complex\\_Ip\\_Packet\\_Source](#)

### 11.363.1 Detailed Description

Receives ip frames containing samples in UDP frame encapsulation using a high performance packet capture library (libpcap)

#### Author

Javier Arribas jarribas (at) cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gr\\_complex\\_ip\\_packet\\_source.h](#).

---

## 11.364 gr\_complex\_ip\_packet\_source.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file gr_complex_ip_packet_source.h
00003  *
00004  * \brief Receives ip frames containing samples in UDP frame encapsulation
00005  * using a high performance packet capture library (libpcap)
00006  * \author Javier Arribas jarribas (at) cttc.es
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
  
```

```

00018
00019
00020 #ifndef GNSS_SDR_GR_COMPLEX_IP_PACKET_SOURCE_H
00021 #define GNSS_SDR_GR_COMPLEX_IP_PACKET_SOURCE_H
00022
00023 #include "gnss_block_interface.h"
00024 #include <boost/thread.hpp>
00025 #include <gnuradio/sync_block.h>
00026 #include <arpa/inet.h>
00027 #include <net/ethernet.h>
00028 #include <net/if.h>
00029 #include <netinet/if_ether.h>
00030 #include <pcap.h>
00031 #include <string>
00032 #include <sys/ioctl.h>
00033
00034 /** \addtogroup Signal_Source
00035  * \{ */
00036 /** \addtogroup Signal_Source_gnuradio_blocks signal_source_gr_blocks
00037  * GNU Radio blocks for signal sources.
00038  * \{ */
00039
00040
00041 class Gr_Complex_Ip_Packet_Source : virtual public gr::sync_block
00042 {
00043 public:
00044     using sptr = gnss_shared_ptr<Gr_Complex_Ip_Packet_Source>;
00045     static sptr make(std::string src_device,
00046                     const std::string &origin_address,
00047                     int udp_port,
00048                     int udp_packet_size,
00049                     int n_baseband_channels,
00050                     const std::string &wire_sample_type,
00051                     size_t item_size,
00052                     bool IQ_swap_);
00053     Gr_Complex_Ip_Packet_Source(std::string src_device,
00054                                const std::string &origin_address,
00055                                int udp_port,
00056                                int udp_packet_size,
00057                                int n_baseband_channels,
00058                                const std::string &wire_sample_type,
00059                                size_t item_size,
00060                                bool IQ_swap_);
00061     ~Gr_Complex_Ip_Packet_Source();
00062
00063     // Called by gnuradio to enable drivers, etc for i/o devices.
00064     bool start();
00065
00066     // Called by gnuradio to disable drivers, etc for i/o devices.
00067     bool stop();
00068
00069     // Where all the action really happens
00070     int work(int noutput_items,
00071             gr_vector_const_void_star &input_items,
00072             gr_vector_void_star &output_items);
00073
00074 private:
00075     void demux_samples(const gr_vector_void_star &output_items, int num_samples_readed);
00076     void my_pcap_loop_thread(pcap_t *pcap_handle);
00077     void pcap_callback(u_char *args, const struct pcap_pkthdr *pkthdr, const u_char *packet);
00078     static void static_pcap_callback(u_char *args, const struct pcap_pkthdr *pkthdr, const u_char
00079 *packet);
00080     /**
00081      * Opens the ethernet device using libpcap raw capture mode
00082      * If any of these fail, the function returns the error and exits.
00083      */
00084     bool open();
00085
00086     boost::thread *d_pcap_thread;
00087     // boost::mutex d_mutex;
00088     struct sockaddr_in si_me{};
00089     std::string d_src_device;
00090     std::string d_origin_address;
00091     pcap_t *d_descr; // ethernet pcap device descriptor
00092     char *d_fifo_buff;
00093     int d_fifo_read_ptr;
00094     int d_fifo_write_ptr;
00095     int d_fifo_items;
00096     int d_sock_raw;
00097     int d_udp_port;
00098     int d_n_baseband_channels;
00099     int d_wire_sample_type;
00100     float d_bytes_per_sample;
00101     bool d_IQ_swap;
00102 };
00103

```

```

00104 /** \} */
00105 /** \} */
00106 #endif // GNSS_SDR_GR_COMPLEX_IP_PACKET_SOURCE_H

```

## 11.365 ion\_gsms.h File Reference

GNU Radio block that reads a Block from a file following ION's GNSS-SDR metadata standard.

```

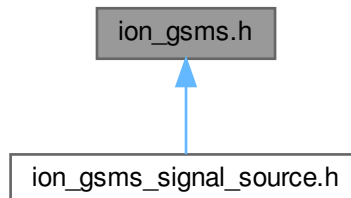
#include "gnss_block_interface.h"
#include "gnss_sdr_filesystem.h"
#include "ion_gsms_chunk_data.h"
#include <gnuradio/block.h>
#include <gnuradio/sync_block.h>
#include <cstdio>
#include <fstream>
#include <memory>
#include <string>
#include <vector>

```

Include dependency graph for ion\_gsms.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [IONGSMSFileSource](#)

### 11.365.1 Detailed Description

GNU Radio block that reads a Block from a file following ION's GNSS-SDR metadata standard.

#### Author

Víctor Castillo Agüero, 2024. [victorcastilloagüero\(at\)gmail.com](mailto:victorcastilloagüero(at)gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [ion\\_gsms.h](#).

---

## 11.366 ion\_gsms.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file ion_gsms.h
00003  * \brief GNU Radio block that reads a Block from a file following ION's GNSS-SDR metadata standard
00004  * \author Victor Castillo Agüero, 2024. victorcastilloaguero(at)gmail.com
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_ION_GSMS_H
00018 #define GNSS_SDR_ION_GSMS_H
00019
00020 #include "gnss_block_interface.h"
00021 #include "gnss_sdr_filesystem.h"
00022 #include "ion_gsms_chunk_data.h"
00023 #include <gnuradio/block.h>
00024 #include <gnuradio/sync_block.h>
00025 #include <cstdint>
00026 #include <fstream>
00027 #include <memory>
00028 #include <string>
00029 #include <vector>
00030
00031 /** \addtogroup Signal_Source
00032  * \{ */
00033 /** \addtogroup Signal_Source_gnuradio_blocks
00034  * \{ */
00035
00036 class IONGSMSFileSource : public gr::sync_block
00037 {
00038 public:
00039     using sptr = gnss_shared_ptr<IONGSMSFileSource>;
00040
00041     IONGSMSFileSource(
00042         const fs::path& metadata_filepath,
00043         const GnssMetadata::File& file,
00044         const GnssMetadata::Block& block,
00045         const std::vector<std::string>& stream_ids);
00046
00047     int work(
00048         int noutput_items,
00049         gr_vector_const_void_star& input_items,
00050         gr_vector_void_star& output_items) override;
00051
00052     std::size_t output_stream_count() const;
00053     std::size_t output_stream_item_size(std::size_t stream_index) const;
00054     std::size_t output_stream_total_sample_count(std::size_t stream_index) const;
00055
00056 private:
00057     static gr::io_signature::sptr make_output_signature(const GnssMetadata::Block& block, const
std::vector<std::string>& stream_ids);
00058
00059     std::ifstream file_stream_;
00060     std::vector<char> io_buffer_;
00061     std::size_t io_buffer_offset_;
00062     std::vector<int> items_produced_;
00063     std::size_t output_stream_count_;
00064     std::vector<std::size_t> output_stream_item_sizes_;
00065     std::vector<std::size_t> output_stream_item_rates_;
00066     std::vector<std::size_t> output_stream_total_sample_counts_;
00067     std::size_t maximum_item_rate_;
00068     std::vector<std::shared_ptr<IONGSMSChunkData>> chunk_data_;
00069     std::size_t chunk_cycle_length_;
00070 };
00071
00072 /** \} */
00073 /** \} */
00074 #endif // GNSS_SDR_ION_GSMS_H

```

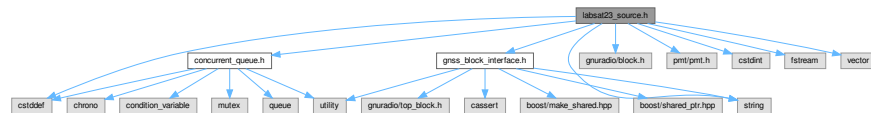
## 11.367 labsat23\_source.h File Reference

Unpacks capture files in the LabSat 2 (ls2), LabSat 3 (ls3), or LabSat 3 Wideband (LS3W) formats.



```
#include "concurrent_queue.h"
#include "gnss_block_interface.h"
#include <gnuradio/block.h>
#include <pmt/pmt.h>
#include <cstdint>
#include <stdint>
#include <fstream>
#include <string>
#include <vector>
```

Include dependency graph for labsat23\_source.h:



## Classes

- class [labsat23\\_source](#)

*This class implements conversion between Labsat 2, 3 and 3 Wideband formats to gr\_complex.*

## Typedefs

- using [labsat23\\_source\\_sptr](#) = `gnss_shared_ptr<labsat23\_source>`

## Functions

- `labsat23_source_sptr labsat23\_make\_source\_sptr (const char *signal_file_basename, const std::vector<int> &channel_selector, Concurrent\_Queue<pmt::pmt_t> *queue, bool digital_io_enabled)`

### 11.367.1 Detailed Description

Unpacks capture files in the LabSat 2 (ls2), LabSat 3 (ls3), or LabSat 3 Wideband (LS3W) formats.

#### Author

Javier Arribas jarribas (at) cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [labsat23\\_source.h](#).

---

## 11.368 labsat23\_source.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file labsat23_source.h
00003  *
00004  * \brief Unpacks capture files in the LabSat 2 (ls2), LabSat 3 (ls3), or LabSat
00005  * 3 Wideband (LS3W) formats.
00006  * \author Javier Arribas jarribas (at) cttc.es
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
```

```

00019 #ifndef GNSS_SDR_LABSAT23_SOURCE_H
00020 #define GNSS_SDR_LABSAT23_SOURCE_H
00021
00022 #include "concurrent_queue.h"
00023 #include "gnss_block_interface.h"
00024 #include <gnuradio/block.h>
00025 #include <pmt/pmt.h>
00026 #include <cstdint>
00027 #include <fstream>
00028 #include <string>
00029 #include <vector>
00030
00031 /** \addtogroup Signal_Source
00032 * \{ */
00033 /** \addtogroup Signal_Source_gnuradio_blocks
00034 * \{ */
00035
00036
00037
00038 class labsat23_source;
00039
00040 using labsat23_source_sptr = gnss_shared_ptr<labsat23_source>;
00041
00042 labsat23_source_sptr labsat23_make_source_sptr(
00043     const char *signal_file_basename,
00044     const std::vector<int> &channel_selector,
00045     Concurrent_Queue<pmt::pmt_t> *queue,
00046     bool digital_io_enabled);
00047
00048 /*!
00049 * \brief This class implements conversion between Labsat 2, 3 and 3 Wideband
00050 * formats to gr_complex
00051 */
00052 class labsat23_source : public gr::block
00053 {
00054 public:
00055     ~labsat23_source();
00056
00057     int general_work(int noutput_items,
00058         gr_vector_int &ninput_items,
00059         gr_vector_const_void_star &input_items,
00060         gr_vector_void_star &output_items);
00061
00062 private:
00063     friend labsat23_source_sptr labsat23_make_source_sptr(
00064         const char *signal_file_basename,
00065         const std::vector<int> &channel_selector,
00066         Concurrent_Queue<pmt::pmt_t> *queue,
00067         bool digital_io_enabled);
00068
00069     labsat23_source(const char *signal_file_basename,
00070         const std::vector<int> &channel_selector,
00071         Concurrent_Queue<pmt::pmt_t> *queue,
00072         bool digital_io_enabled);
00073
00074     std::string generate_filename();
00075
00076     int parse_header();
00077     int getBit(uint8_t byte, int position);
00078     int read_ls3w_ini(const std::string &filename);
00079     int number_of_samples_per_ls3w_register() const;
00080
00081     void decode_samples_one_channel(int16_t input_short, gr_complex *out, int type);
00082     void decode_ls3w_register(uint64_t input, std::vector<gr_complex *> &out, std::size_t
00083         output_pointer) const;
00084
00085     std::ifstream binary_input_file;
00086     std::string d_signal_file_basename;
00087     Concurrent_Queue<pmt::pmt_t> *d_queue;
00088     std::vector<int> d_channel_selector_config;
00089     int d_current_file_number;
00090     uint8_t d_labsat_version;
00091     uint8_t d_channel_selector;
00092     uint8_t d_ref_clock;
00093     uint8_t d_bits_per_sample;
00094     bool d_header_parsed;
00095
00096     // Data members for Labsat 3 Wideband
00097     std::string d_ls3w_OSC;
00098     std::vector<int> d_ls3w_selected_channel_offset;
00099     int64_t d_ls3w_SMP{};
00100     int32_t d_ls3w_QUA{};
00101     int32_t d_ls3w_CHN{};
00102     int32_t d_ls3w_SFT{};
00103     int32_t d_ls3w_CFA{};
00104     int32_t d_ls3w_CFB{};
00105     int32_t d_ls3w_CFC{};

```

```

00105     int32_t d_ls3w_BWA{};
00106     int32_t d_ls3w_BWB{};
00107     int32_t d_ls3w_BWC{};
00108     int d_ls3w_spare_bits{};
00109     int d_ls3w_samples_per_register{};
00110     bool d_is_ls3w = false;
00111     bool d_ls3w_digital_io_enabled = false;
00112 };
00113
00114
00115 /** \} */
00116 /** \} */
00117 #endif // GNSS_SDR_LABSAT23_SOURCE_H

```

## 11.369 rtl\_tcp\_signal\_source\_c.h File Reference

Interface of an rtl\_tcp signal source reader.

```

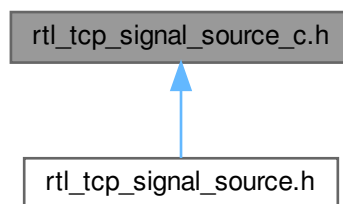
#include "gnss_block_interface.h"
#include "rtl_tcp_dongle_info.h"
#include <boost/array.hpp>
#include <boost/asio.hpp>
#include <boost/circular_buffer.hpp>
#include <boost/thread/condition.hpp>
#include <boost/thread/mutex.hpp>
#include <gnuradio/sync_block.h>
#include <cstdint>
#include <string>
#include <vector>

```

Include dependency graph for rtl\_tcp\_signal\_source\_c.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [rtl\\_tcp\\_signal\\_source\\_c](#)

*This class reads interleaved I/Q samples from an rtl\_tcp server and outputs complex types.*

### Typedefs

- using [rtl\\_tcp\\_signal\\_source\\_c\\_sptr](#) = [gnss\\_shared\\_ptr](#)<[rtl\\_tcp\\_signal\\_source\\_c](#)>
- using [b\\_io\\_context](#) = [boost::asio::io\\_service](#)

## Functions

- `rtl_tcp_signal_source_c_sptr rtl_tcp_make_signal_source_c` (const std::string &address, int16\_t port, bool flip\_iq=false)

### 11.369.1 Detailed Description

Interface of an `rtl_tcp` signal source reader.

#### Author

Anthony Arnold, 2015. [anthony.arnold\(at\)uqconnect.edu.au](mailto:anthony.arnold(at)uqconnect.edu.au)

The implementation of this block is a combination of various helpful sources. The data format and command structure is taken from the original Osmocom `rtl_tcp_source_f` (<https://git.osmocom.org/gr-osmosdr>). The asynchronous reading code comes from the examples provides by Boost.Asio and the bounded buffer producer-consumer solution is taken from the Boost.CircularBuffer examples (<https://www.boost.org/>). GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [rtl\\_tcp\\_signal\\_source\\_c.h](#).

## 11.370 rtl\_tcp\_signal\_source\_c.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file rtl_tcp_signal_source_c.h
00003  * \brief Interface of an rtl_tcp signal source reader.
00004  * \author Anthony Arnold, 2015. anthony.arnold\(at\)uqconnect.edu.au
00005  *
00006  * The implementation of this block is a combination of various helpful
00007  * sources. The data format and command structure is taken from the
00008  * original Osmocom rtl_tcp_source_f (https://git.osmocom.org/gr-osmosdr).
00009  * The asynchronous reading code comes from the examples provides
00010  * by Boost.Asio and the bounded buffer producer-consumer solution is
00011  * taken from the Boost.CircularBuffer examples (https://www.boost.org/).
00012  *
00013  * -----
00014  *
00015  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00016  * This file is part of GNSS-SDR.
00017  *
00018  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00019  * SPDX-License-Identifier: GPL-3.0-or-later
00020  *
00021  * -----
00022  */
00023
00024 #ifndef GNSS_SDR_RTL_TCP_SIGNAL_SOURCE_C_H
00025 #define GNSS_SDR_RTL_TCP_SIGNAL_SOURCE_C_H
00026
00027 #include "gnss_block_interface.h"
00028 #include "rtl_tcp_dongle_info.h"
00029 #include <boost/array.hpp>
00030 #include <boost/asio.hpp>
00031 #include <boost/circular_buffer.hpp>
00032 #include <boost/thread/condition.hpp>
00033 #include <boost/thread/mutex.hpp>
00034 #include <gnuradio/sync_block.h>
00035 #include <cstdint>
00036 #include <string>
00037 #include <vector>
00038
00039
00040 /** \addtogroup Signal_Source
00041  * \{ */
00042 /** \addtogroup Signal_Source_gnuradio_blocks
00043  * \{ */
00044
00045
00046 class rtl_tcp_signal_source_c;
00047
00048 using rtl_tcp_signal_source_c_sptr = gnss_shared_ptr<rtl_tcp_signal_source_c>;
00049
00050 #if USE_BOOST_ASIO_IO_CONTEXT
00051 using b_io_context = boost::asio::io_context;
00052 #else
00053 using b_io_context = boost::asio::io_service;
```

```

00054 #endif
00055
00056 rtl_tcp_signal_source_c_sptr
00057 rtl_tcp_make_signal_source_c(const std::string &address,
00058     int16_t port,
00059     bool flip_iq = false);
00060
00061 /*!
00062  * \brief This class reads interleaved I/Q samples
00063  * from an rtl_tcp server and outputs complex types.
00064  */
00065 class rtl_tcp_signal_source_c : public gr::sync_block
00066 {
00067 public:
00068     ~rtl_tcp_signal_source_c();
00069
00070     int work(int noutput_items,
00071         gr_vector_const_void_star &input_items,
00072         gr_vector_void_star &output_items);
00073
00074     void set_frequency(int frequency);
00075     void set_sample_rate(int sample_rate);
00076     void set_agc_mode(bool agc);
00077     void set_gain(int gain);
00078     void set_if_gain(int gain);
00079
00080 private:
00081     friend rtl_tcp_signal_source_c_sptr
00082     rtl_tcp_make_signal_source_c(const std::string &address,
00083         int16_t port,
00084         bool flip_iq);
00085
00086     rtl_tcp_signal_source_c(const std::string &address,
00087         int16_t port,
00088         bool flip_iq);
00089
00090     // async read callback
00091     void handle_read(const boost::system::error_code &ec,
00092         size_t bytes_transferred);
00093
00094     inline bool not_full() const
00095     {
00096         return unread_ < buffer_.capacity();
00097     }
00098
00099     inline bool not_empty() const
00100     {
00101         return unread_ > 0 || io_context_.stopped();
00102     }
00103
00104     boost::circular_buffer_space_optimized<float> buffer_;
00105     // producer-consumer helpers
00106     boost::mutex mutex_;
00107     boost::condition not_full_;
00108     boost::condition not_empty_;
00109
00110     // lookup for scaling data
00111     boost::array<float, 0xff> lookup_{};
00112
00113     // IO members
00114     b_io_context io_context_;
00115     boost::asio::ip::tcp::socket socket_;
00116     std::vector<unsigned char> data_;
00117
00118     Rtl_Tcp_Dongle_Info info_;
00119     size_t unread_;
00120     bool flip_iq_;
00121 };
00122
00123
00124 /** \} */
00125 /** \} */
00126 #endif // GNSS_SDR_RTL_TCP_SIGNAL_SOURCE_C_H

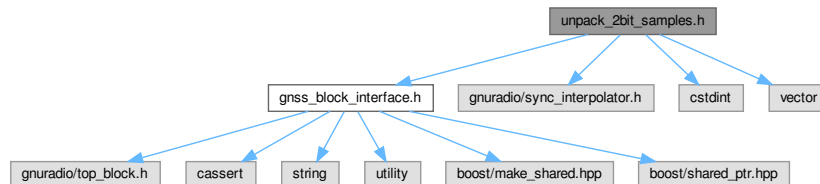
```

## 11.371 unpack\_2bit\_samples.h File Reference

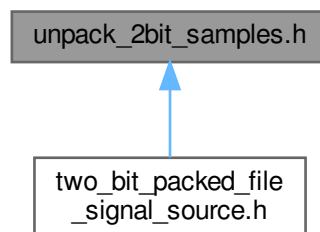
Unpacks 2 bit samples samples may be packed in any of the following ways: 1) Into bytes [ item == byte ] 1a) Big endian ordering within the byte 1b) Little endian ordering within the byte 2) Into shorts [ item == short ] 2a) Big endian ordering of bytes, big endian within the byte 2b) Big endian ordering of bytes, little endian within the byte 2c) Little endian ordering of bytes, big endian within the byte 2d) Little endian ordering of bytes, little endian within the byte.

```
#include "gnss_block_interface.h"
#include <gnuradio/sync_interpolator.h>
#include <stdint>
#include <vector>
```

Include dependency graph for unpack\_2bit\_samples.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [unpack\\_2bit\\_samples](#)

*This class takes 2 bit samples that have been packed into bytes or shorts as input and generates a byte for each sample. It generates eight times as much data as is input (every two bits become 16 bits)*

## Typedefs

- using [unpack\\_2bit\\_samples\\_sptr](#) = `gnss_shared_ptr<unpack\_2bit\_samples>`

## Functions

- `unpack_2bit_samples_sptr make_unpack_2bit_samples (bool big_endian_bytes, size_t item_size, bool big_endian_items, bool reverse_interleaving=false)`

### 11.371.1 Detailed Description

Unpacks 2 bit samples samples may be packed in any of the following ways: 1) Into bytes [ item == byte ] 1a) Big endian ordering within the byte 1b) Little endian ordering within the byte 2) Into shorts [ item == short ] 2a) Big endian ordering of bytes, big endian within the byte 2b) Big endian ordering of bytes, little endian within the byte 2c) Little endian ordering of bytes, big endian within the byte 2d) Little endian ordering of bytes, little endian within the byte.

Within a byte the two possibilities look like this: 7 6 5 4 3 2 1 0 : Bit number  $x_{n,1}$   $x_{n,0}$   $x_{n+1,1}$   $x_{n+1,0}$   $x_{n+2,1}$   $x_{n+2,0}$   $x_{n+3,1}$   $x_{n+3,0}$  : Little endian  $x_{n+3,1}$   $x_{n+3,0}$   $x_{n+2,1}$   $x_{n+2,0}$   $x_{n+1,0}$   $x_{n+1,0}$   $x_{n,1}$   $x_{n,0}$  : Big Endian

For a short (uint16\_t) the bytes are either transmitted as follows:

1 0 : Byte number Byte\_n Byte\_n+1 : Little endian Byte\_n+1 Byte\_n : Bit endian

The two bit values are assumed to have the following mapping:

$x_1$   $x_0$  Value 0 0 +1 0 1 +3 1 0 -3 1 1 -1

Letting  $x$  denote the two's complement interpretation of  $x_1$   $x_0$ , then:

Value =  $2*x + 1$

We want to output the data in the order:

Value\_0, Value\_1, Value\_2, ..., Value\_n, Value\_n+1, Value\_n+2, ...

#### Author

Cillian O'Driscoll cillian.odriscoll (at) gmail . com

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [unpack\\_2bit\\_samples.h](#).

## 11.372 unpack\_2bit\_samples.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file unpack_2bit_samples.h
00003  *
00004  * \brief Unpacks 2 bit samples
00005  * samples may be packed in any of the following ways:
00006  * 1) Into bytes [ item == byte ]
00007  *     1a) Big endian ordering within the byte
00008  *     1b) Little endian ordering within the byte
00009  * 2) Into shorts [ item == short ]
00010  *     2a) Big endian ordering of bytes, big endian within the byte
00011  *     2b) Big endian ordering of bytes, little endian within the byte
00012  *     2c) Little endian ordering of bytes, big endian within the byte
00013  *     2d) Little endian ordering of bytes, little endian within the byte
00014  *
00015  * Within a byte the two possibilities look like this:
00016  *       7       6       5       4       3       2       1       0       : Bit number
00017  *    $x_{n,1}$    $x_{n,0}$    $x_{n+1,1}$   $x_{n+1,0}$   $x_{n+2,1}$   $x_{n+2,0}$   $x_{n+3,1}$   $x_{n+3,0}$  : Little endian
00018  *    $x_{n+3,1}$   $x_{n+3,0}$   $x_{n+2,1}$   $x_{n+2,0}$   $x_{n+1,0}$   $x_{n+1,0}$   $x_{n,1}$    $x_{n,0}$  : Big Endian
00019  *
00020  * For a short (uint16_t) the bytes are either transmitted as follows:
00021  *
00022  *       1       0       : Byte number
00023  *   Byte_n  Byte_n+1  : Little endian
00024  *   Byte_n+1 Byte_n    : Bit endian
00025  *
00026  * The two bit values are assumed to have the following mapping:
00027  *
00028  *    $x_1$    $x_0$     Value
00029  *       0       0      +1
00030  *       0       1      +3
00031  *       1       0      -3
00032  *       1       1      -1
00033  *
00034  * Letting  $x$  denote the two's complement interpretation of  $x_1$   $x_0$ , then:
00035  *
00036  *   Value =  $2*x + 1$ 
00037  *
00038  * We want to output the data in the order:
00039  *
00040  *   Value_0, Value_1, Value_2, ..., Value_n, Value_n+1, Value_n+2, ...
00041  *
00042  * \author Cillian O'Driscoll cillian.odriscoll (at) gmail . com
00043  *
00044  * -----
00045  *
00046  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00047  * This file is part of GNSS-SDR.
00048  *
00049  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00050  * SPDX-License-Identifier: GPL-3.0-or-later
00051  *
00052  * -----
00053  */
00054
```

```

00055 #ifndef GNSS_SDR_UNPACK_2BIT_SAMPLES_H
00056 #define GNSS_SDR_UNPACK_2BIT_SAMPLES_H
00057
00058 #include "gnss_block_interface.h"
00059 #include <gnuradio/sync_interpolator.h>
00060 #include <cstdint>
00061 #include <vector>
00062
00063 /** \addtogroup Signal_Source
00064  * \{ */
00065 /** \addtogroup Signal_Source_gnuradio_blocks
00066  * \{ */
00067
00068
00069 class unpack_2bit_samples;
00070
00071 using unpack_2bit_samples_sptr = gnss_shared_ptr<unpack_2bit_samples>;
00072
00073 unpack_2bit_samples_sptr make_unpack_2bit_samples(
00074     bool big_endian_bytes,
00075     size_t item_size,
00076     bool big_endian_items,
00077     bool reverse_interleaving = false);
00078
00079 /*!
00080  * \brief This class takes 2 bit samples that have been packed into bytes or
00081  * shorts as input and generates a byte for each sample. It generates eight
00082  * times as much data as is input (every two bits become 16 bits)
00083  */
00084 class unpack_2bit_samples : public gr::sync_interpolator
00085 {
00086 public:
00087     ~unpack_2bit_samples() = default;
00088
00089     unpack_2bit_samples(bool big_endian_bytes,
00090         size_t item_size,
00091         bool big_endian_items,
00092         bool reverse_interleaving);
00093
00094     int work(int noutput_items,
00095         gr_vector_const_void_star &input_items,
00096         gr_vector_void_star &output_items);
00097
00098 private:
00099     friend unpack_2bit_samples_sptr make_unpack_2bit_samples_sptr(
00100         bool big_endian_bytes,
00101         size_t item_size,
00102         bool big_endian_items,
00103         bool reverse_interleaving);
00104
00105     std::vector<int8_t> work_buffer_;
00106     size_t item_size_;
00107     bool big_endian_bytes_;
00108     bool big_endian_items_;
00109     bool swap_endian_items_;
00110     bool swap_endian_bytes_;
00111     bool reverse_interleaving_;
00112 };
00113
00114
00115 /** \} */
00116 /** \} */
00117 #endif // GNSS_SDR_UNPACK_2BIT_SAMPLES_H

```

## 11.373 unpack\_byte\_2bit\_cpx\_samples.h File Reference

Unpacks byte samples to 2 bits complex samples. Packing Order Most Significant Nibble - Sample n Least Significant Nibble - Sample n+1 Packing order in Nibble Q1 Q0 I1 I0.

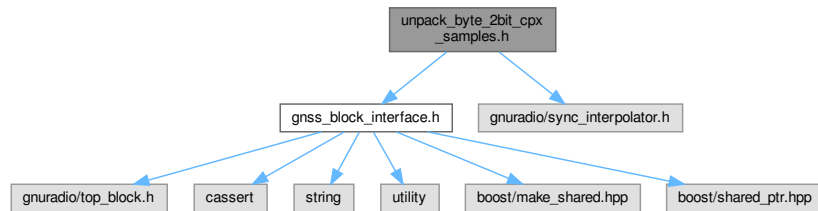
```

#include "gnss_block_interface.h"
#include <gnuradio/sync_interpolator.h>

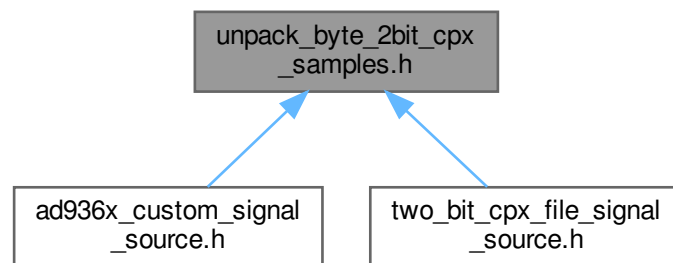
```



Include dependency graph for unpack\_byte\_2bit\_cpx\_samples.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [unpack\\_byte\\_2bit\\_cpx\\_samples](#)

*This class implements conversion between byte packet samples to 2bit\_cpx samples 1 byte = 2 x complex 2bit I, + 2bit Q samples.*

## Typedefs

- using [unpack\\_byte\\_2bit\\_cpx\\_samples\\_sptr](#) = `gnss_shared_ptr<unpack\_byte\_2bit\_cpx\_samples>`

## Functions

- `unpack_byte_2bit_cpx_samples_sptr` **make\_unpack\_byte\_2bit\_cpx\_samples** ()

### 11.373.1 Detailed Description

Unpacks byte samples to 2 bits complex samples. Packing Order Most Significant Nibble - Sample n Least Significant Nibble - Sample n+1 Packing order in Nibble Q1 Q0 I1 I0.

#### Author

Javier Arribas jarribas (at) cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [unpack\\_byte\\_2bit\\_cpx\\_samples.h](#).

---

## 11.374 unpack\_byte\_2bit\_cpx\_samples.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file unpack_byte_2bit_cpx_samples.h
00003  *
00004  * \brief Unpacks byte samples to 2 bits complex samples.
00005  *      Packing Order
00006  *      Most Significant Nibble - Sample n
00007  *      Least Significant Nibble - Sample n+1
00008  *      Packing order in Nibble Q1 Q0 I1 I0
00009  * \author Javier Arribas jarribas (at) cttc.es
00010  *
00011  * -----
00012  *
00013  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00014  * This file is part of GNSS-SDR.
00015  *
00016  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00017  * SPDX-License-Identifier: GPL-3.0-or-later
00018  *
00019  * -----
00020  */
00021
00022 #ifndef GNSS_SDR_UNPACK_BYTE_2BIT_CPX_SAMPLES_H
00023 #define GNSS_SDR_UNPACK_BYTE_2BIT_CPX_SAMPLES_H
00024
00025 #include "gnss_block_interface.h"
00026 #include <gnuradio/sync_interpolator.h>
00027
00028 /** \addtogroup Signal_Source
00029  * \{ */
00030 /** \addtogroup Signal_Source_gnuradio_blocks
00031  * \{ */
00032
00033
00034 class unpack_byte_2bit_cpx_samples;
00035
00036 using unpack_byte_2bit_cpx_samples_sptr = gnss_shared_ptr<unpack_byte_2bit_cpx_samples>;
00037
00038 unpack_byte_2bit_cpx_samples_sptr make_unpack_byte_2bit_cpx_samples();
00039
00040 /*!
00041  * \brief This class implements conversion between byte packet samples to 2bit_cpx samples
00042  *      1 byte = 2 x complex 2bit I, + 2bit Q samples
00043  */
00044 class unpack_byte_2bit_cpx_samples : public gr::sync_interpolator
00045 {
00046 public:
00047     unpack_byte_2bit_cpx_samples();
00048     ~unpack_byte_2bit_cpx_samples() = default;
00049     int work(int noutput_items,
00050             gr_vector_const_void_star &input_items,
00051             gr_vector_void_star &output_items);
00052 private:
00053     friend unpack_byte_2bit_cpx_samples_sptr make_unpack_byte_2bit_cpx_samples_sptr();
00054 };
00055
00056
00057
00058 /** \} */
00059 /** \} */
00060 #endif // GNSS_SDR_UNPACK_BYTE_2BIT_CPX_SAMPLES_H

```

## 11.375 unpack\_byte\_2bit\_samples.h File Reference

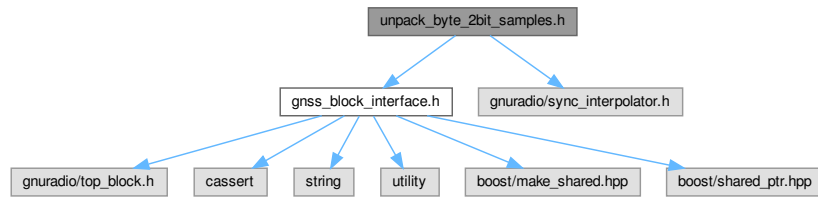
Unpacks byte samples to NSR 2 bits samples.

```

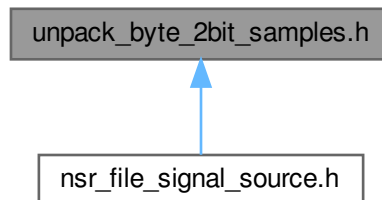
#include "gnss_block_interface.h"
#include <gnuradio/sync_interpolator.h>

```

Include dependency graph for unpack\_byte\_2bit\_samples.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [unpack\\_byte\\_2bit\\_samples](#)

*This class implements conversion between byte packet samples to 2bit samples 1 byte = 4 2bit samples.*

## Typedefs

- using [unpack\\_byte\\_2bit\\_samples\\_sptr](#) = [gnss\\_shared\\_ptr](#)<[unpack\\_byte\\_2bit\\_samples](#)>

## Functions

- [unpack\\_byte\\_2bit\\_samples\\_sptr](#) **make\_unpack\_byte\_2bit\_samples** ()

### 11.375.1 Detailed Description

Unpacks byte samples to NSR 2 bits samples.

#### Author

Javier Arribas jarribas (at) cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [unpack\\_byte\\_2bit\\_samples.h](#).

---

## 11.376 unpack\_byte\_2bit\_samples.h

[Go to the documentation of this file.](#)

00001 /\*!

```

00002  * \file unpack_byte_2bit_samples.h
00003  *
00004  * \brief Unpacks byte samples to NSR 2 bits samples
00005  * \author Javier Arribas jarribas (at) cttc.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_UNPACK_BYTE_2BIT_SAMPLES_H
00019 #define GNSS_SDR_UNPACK_BYTE_2BIT_SAMPLES_H
00020
00021 #include "gnss_block_interface.h"
00022 #include <gnuradio/sync_interpolator.h>
00023
00024
00025 /** \addtogroup Signal_Source
00026  * \{ */
00027 /** \addtogroup Signal_Source_gnuradio_blocks
00028  * \{ */
00029
00030 class unpack_byte_2bit_samples;
00031
00032 using unpack_byte_2bit_samples_sptr = gnss_shared_ptr<unpack_byte_2bit_samples>;
00033
00034 unpack_byte_2bit_samples_sptr make_unpack_byte_2bit_samples();
00035
00036 /*!
00037  * \brief This class implements conversion between byte packet samples to 2bit samples
00038  * 1 byte = 4 2bit samples
00039  */
00040 class unpack_byte_2bit_samples : public gr::sync_interpolator
00041 {
00042 public:
00043     unpack_byte_2bit_samples();
00044     ~unpack_byte_2bit_samples() = default;
00045     int work(int noutput_items,
00046             gr_vector_const_void_star &input_items,
00047             gr_vector_void_star &output_items);
00048 private:
00049     friend unpack_byte_2bit_samples_sptr make_unpack_byte_2bit_samples_sptr();
00050 };
00051
00052
00053
00054 /** \} */
00055 /** \} */
00056 #endif // GNSS_SDR_UNPACK_BYTE_2BIT_SAMPLES_H

```

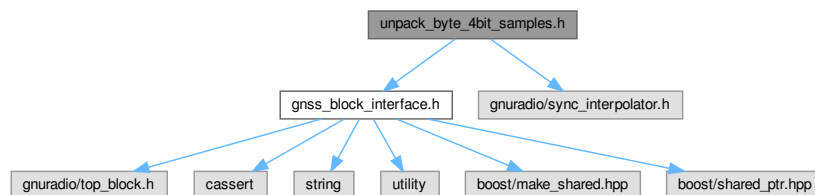
## 11.377 unpack\_byte\_4bit\_samples.h File Reference

Unpacks byte samples to 4 bits samples. Packing Order Packing order in Nibble I0 I1 I2 I3 I0 I1 I2 I3.

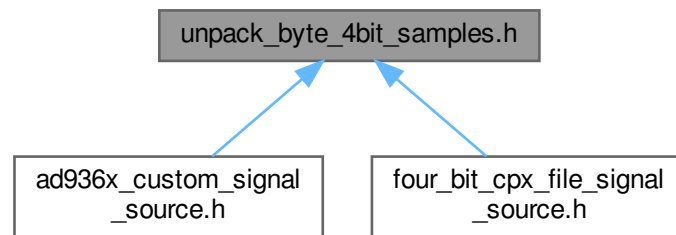
```
#include "gnss_block_interface.h"
```

```
#include <gnuradio/sync_interpolator.h>
```

Include dependency graph for unpack\_byte\_4bit\_samples.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [unpack\\_byte\\_4bit\\_samples](#)

*This class implements conversion between byte packet samples to 4bit\_cpx samples 1 byte = 1 x complex 4bit I, + 4bit Q samples.*

## Typedefs

- using [unpack\\_byte\\_4bit\\_samples\\_sptr](#) = gnss\_shared\_ptr<[unpack\\_byte\\_4bit\\_samples](#)>

## Functions

- [unpack\\_byte\\_4bit\\_samples\\_sptr](#) **make\_unpack\_byte\_4bit\_samples** ()

### 11.377.1 Detailed Description

Unpacks byte samples to 4 bits samples. Packing Order Packing order in Nibble I0 I1 I2 I3 I0 I1 I2 I3.

#### Author

Javier Arribas jarribas (at) cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [unpack\\_byte\\_4bit\\_samples.h](#).

---

## 11.378 unpack\_byte\_4bit\_samples.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file unpack_byte_4bit_samples.h
00003  *
00004  * \brief Unpacks byte samples to 4 bits samples.
00005  *      Packing Order
00006  *      Packing order in Nibble I0 I1 I2 I3 I0 I1 I2 I3
00007  * \author Javier Arribas jarribas (at) cttc.es
00008  *
00009  * -----
00010  *
00011  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00012  * This file is part of GNSS-SDR.
00013  *
00014  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00015  * SPDX-License-Identifier: GPL-3.0-or-later
00016  *
00017  * -----
00018  */
00019  
```

```

00020 #ifndef GNSS_SDR_UNPACK_BYTE_4BIT_SAMPLES_H
00021 #define GNSS_SDR_UNPACK_BYTE_4BIT_SAMPLES_H
00022
00023 #include "gnss_block_interface.h"
00024 #include <gnuradio/sync_interpolator.h>
00025
00026 /** \addtogroup Signal_Source
00027 * \{ */
00028 /** \addtogroup Signal_Source_gnuradio_blocks
00029 * \{ */
00030
00031
00032 class unpack_byte_4bit_samples;
00033
00034 using unpack_byte_4bit_samples_sptr = gnss_shared_ptr<unpack_byte_4bit_samples>;
00035
00036 unpack_byte_4bit_samples_sptr make_unpack_byte_4bit_samples();
00037
00038 /*!
00039 * \brief This class implements conversion between byte packet samples to 4bit_cpx samples
00040 * 1 byte = 1 x complex 4bit I, + 4bit Q samples
00041 */
00042 class unpack_byte_4bit_samples : public gr::sync_interpolator
00043 {
00044 public:
00045     unpack_byte_4bit_samples();
00046     ~unpack_byte_4bit_samples() = default;
00047     int work(int noutput_items,
00048             gr_vector_const_void_star &input_items,
00049             gr_vector_void_star &output_items);
00050 private:
00051     friend unpack_byte_4bit_samples_sptr make_unpack_byte_4bit_samples_sptr();
00052 };
00053
00054
00055
00056 /** \} */
00057 /** \} */
00058 #endif // GNSS_SDR_UNPACK_BYTE_4BIT_SAMPLES_H

```

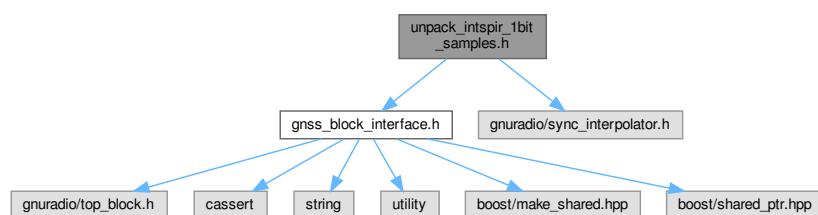
## 11.379 unpack\_intspir\_1bit\_samples.h File Reference

Unpacks SPIR int samples to NSR 1 bit samples.

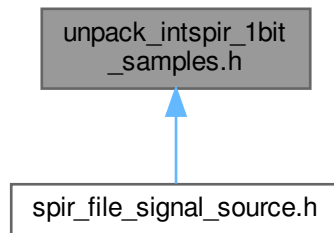
```
#include "gnss_block_interface.h"
```

```
#include <gnuradio/sync_interpolator.h>
```

Include dependency graph for unpack\_intspir\_1bit\_samples.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [unpack\\_intspir\\_1bit\\_samples](#)

*This class implements conversion between byte packet samples to 2bit samples 1 byte = 4 2bit samples.*

## Typedefs

- using [unpack\\_intspir\\_1bit\\_samples\\_sptr](#) = gnss\_shared\_ptr<[unpack\\_intspir\\_1bit\\_samples](#)>

## Functions

- [unpack\\_intspir\\_1bit\\_samples\\_sptr](#) **make\_unpack\_intspir\_1bit\_samples** ()

## 11.379.1 Detailed Description

Unpacks SPIR int samples to NSR 1 bit samples.

### Author

Fran Fabra fabra (at) ice.csic.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is not part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [unpack\\_intspir\\_1bit\\_samples.h](#).

---

## 11.380 unpack\_intspir\_1bit\_samples.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file unpack_intspir_1bit_samples.h
00003  *
00004  * \brief Unpacks SPIR int samples to NSR 1 bit samples
00005  * \author Fran Fabra fabra (at) ice.csic.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is not part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_UNPACK_INTSPIR_1BIT_SAMPLES_H
00019 #define GNSS_SDR_UNPACK_INTSPIR_1BIT_SAMPLES_H
00020

```

```

00021 #include "gnss_block_interface.h"
00022 #include <gnuradio/sync_interpolator.h>
00023
00024
00025 /** \addtogroup Signal_Source
00026 * \{ */
00027 /** \addtogroup Signal_Source_gnuradio_blocks
00028 * \{ */
00029
00030
00031 class unpack_intspir_lbit_samples;
00032
00033 using unpack_intspir_lbit_samples_sptr = gnss_shared_ptr<unpack_intspir_lbit_samples>;
00034
00035 unpack_intspir_lbit_samples_sptr make_unpack_intspir_lbit_samples();
00036
00037 /*!
00038 * \brief This class implements conversion between byte packet samples to 2bit samples
00039 * 1 byte = 4 2bit samples
00040 */
00041 class unpack_intspir_lbit_samples : public gr::sync_interpolator
00042 {
00043 public:
00044     unpack_intspir_lbit_samples();
00045     ~unpack_intspir_lbit_samples() = default;
00046     int work(int noutput_items,
00047             gr_vector_const_void_star &input_items,
00048             gr_vector_void_star &output_items);
00049 private:
00050     friend unpack_intspir_lbit_samples_sptr make_unpack_intspir_lbit_samples_sptr();
00051 };
00052
00053
00054
00055 /** \} */
00056 /** \} */
00057 #endif // GNSS_SDR_UNPACK_INTSPIR_1BIT_SAMPLES_H

```

## 11.381 unpack\_short\_byte\_samples.h

```

00001 /*!
00002 * \file unpack_short_byte_samples.cc
00003 *
00004 * \brief Unpacks shorts samples to byte samples (1 short = 2 byte samples).
00005 *      Packing Order
00006 *      Packing order in Nibble IO I1
00007 * \author Javier Arribas jarribas (at) cttc.es
00008 * -----
00009 *
00010 * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011 * This file is part of GNSS-SDR.
00012 *
00013 * Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors)
00014 * SPDX-License-Identifier: GPL-3.0-or-later
00015 *
00016 * -----
00017 */
00018
00019 #ifndef GNSS_SDR_UNPACK_SHORT_BYTE_SAMPLES_H
00020 #define GNSS_SDR_UNPACK_SHORT_BYTE_SAMPLES_H
00021
00022 #include "gnss_block_interface.h"
00023 #include <gnuradio/sync_interpolator.h>
00024
00025 /** \addtogroup Signal_Source
00026 * \{ */
00027 /** \addtogroup Signal_Source_gnuradio_blocks
00028 * \{ */
00029
00030
00031 class unpack_short_byte_samples;
00032
00033 using unpack_short_byte_samples_sptr = gnss_shared_ptr<unpack_short_byte_samples>;
00034
00035 unpack_short_byte_samples_sptr make_unpack_short_byte_samples();
00036
00037 /*!
00038 * \brief This class implements conversion between short packet samples to byte samples
00039 * 1 short = 2 byte samples
00040 */
00041 class unpack_short_byte_samples : public gr::sync_interpolator
00042 {
00043 public:
00044     unpack_short_byte_samples();

```



```

00045     void forecast(int noutput_items, gr_vector_int &ninput_items_required);
00046     ~unpack_short_byte_samples() = default;
00047     int work(int noutput_items,
00048             gr_vector_const_void_star &input_items,
00049             gr_vector_void_star &output_items);
00050
00051 private:
00052     friend unpack_short_byte_samples_sptr make_unpack_short_byte_samples_sptr();
00053 };
00054
00055
00056 /** \} */
00057 /** \} */
00058 #endif // GNSS_SDR_UNPACK_SHORT_BYTE_SAMPLES_H

```

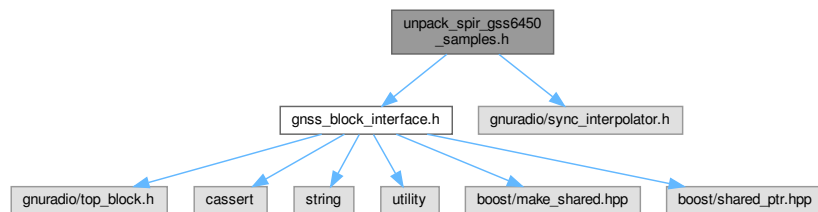
## 11.382 unpack\_spir\_gss6450\_samples.h File Reference

Unpacks SPIR int samples.

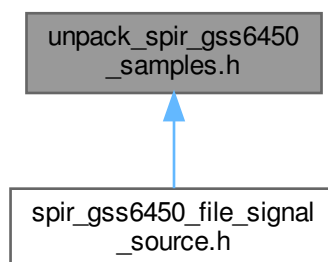
```
#include "gnss_block_interface.h"
```

```
#include <gnuradio/sync_interpolator.h>
```

Include dependency graph for unpack\_spir\_gss6450\_samples.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [unpack\\_spir\\_gss6450\\_samples](#)

### Typedefs

- using [unpack\\_spir\\_gss6450\\_samples\\_sptr](#) = `gnss_shared_ptr<unpack_spir_gss6450_samples>`

## Functions

- `unpack_spir_gss6450_samples_sptr make_unpack_spir_gss6450_samples (int adc_nbit_)`

### 11.382.1 Detailed Description

Unpacks SPIR int samples.

#### Author

Antonio Ramos, antonio.ramos(at)cttc.es

Javier Arribas jarribas (at) cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is not part of GNSS-SDR. Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later Definition in file [unpack\\_spir\\_gss6450\\_samples.h](#).

---

## 11.383 unpack\_spir\_gss6450\_samples.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file unpack_spir_gss6450_samples.h
00003  * \brief Unpacks SPIR int samples
00004  * \author Antonio Ramos, antonio.ramos(at)cttc.es
00005  * \author Javier Arribas jarribas (at) cttc.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is not part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_UNPACK_SPIR_GSS6450_SAMPLES_H
00019 #define GNSS_SDR_UNPACK_SPIR_GSS6450_SAMPLES_H
00020
00021 #include "gnss_block_interface.h"
00022 #include <gnuradio/sync_interpolator.h>
00023
00024 /** \addtogroup Signal_Source
00025  * \{ */
00026 /** \addtogroup Signal_Source_gnuradio_blocks
00027  * \{ */
00028
00029
00030 class unpack_spir_gss6450_samples;
00031
00032 using unpack_spir_gss6450_samples_sptr = gnss_shared_ptr<unpack_spir_gss6450_samples>;
00033
00034 unpack_spir_gss6450_samples_sptr make_unpack_spir_gss6450_samples(int adc_nbit_);
00035
00036
00037 class unpack_spir_gss6450_samples : public gr::sync_interpolator
00038 {
00039 public:
00040     explicit unpack_spir_gss6450_samples(int adc_nbit);
00041     ~unpack_spir_gss6450_samples() = default;
00042     void decode_4bits_word(uint32_t input_uint32, gr_complex *out, int adc_bits_);
00043     int work(int noutput_items,
00044             gr_vector_const_void_star &input_items, gr_vector_void_star &output_items);
00045
00046 private:
00047     friend unpack_spir_gss6450_samples_sptr make_unpack_spir_gss6450_samples_sptr(int adc_nbit);
00048     int adc_bits;
00049     int samples_per_int;
00050 };
00051
00052
00053 /** \} */
00054 /** \} */
00055 #endif // GNSS_SDR_UNPACK_SPIR_GSS6450_SAMPLES_H

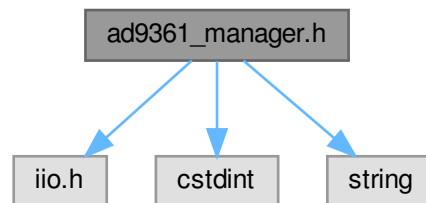
```

## 11.384 ad9361\_manager.h File Reference

An Analog Devices AD9361 front-end configuration library wrapper for configure some functions via iiod link.

```
#include <iio.h>
#include <cstdint>
#include <string>
```

Include dependency graph for ad9361\_manager.h:



### Classes

- struct [stream\\_cfg](#)

### Macros

- #define [FIR\\_BUF\\_SIZE](#) 8192

### Enumerations

- enum **iodev** { **RX** , **TX** }

### Functions

- void **errchk** (int v, const char \*what)
- void **wr\_ch\_lll** (struct iio\_channel \*chn, const char \*what, int64\_t val)
- void **wr\_ch\_str** (struct iio\_channel \*chn, const char \*what, const char \*str)
- struct iio\_device \* **get\_ad9361\_phy** (struct iio\_context \*ctx)
- bool **get\_ad9361\_stream\_dev** (struct iio\_context \*ctx, enum iodev d, struct iio\_device \*\*dev)
- bool **get\_ad9361\_stream\_ch** (struct iio\_context \*ctx, enum iodev d, struct iio\_device \*dev, int chid, struct iio\_channel \*\*chn)
- bool **get\_phy\_chan** (struct iio\_context \*ctx, enum iodev d, int chid, struct iio\_channel \*\*chn)
- bool **get\_lo\_chan** (struct iio\_context \*ctx, enum iodev d, struct iio\_channel \*\*chn)
- bool **cfg\_ad9361\_streaming\_ch** (struct iio\_context \*ctx, struct [stream\\_cfg](#) \*cfg, enum iodev type, int chid)
- bool **config\_ad9361\_rx\_local** (uint64\_t bandwidth\_, uint64\_t sample\_rate\_, uint64\_t freq0\_, uint64\_t freq1\_↵\_, const std::string &rf\_port\_select\_, bool rx1\_enable\_, bool rx2\_enable\_, const std::string &gain\_mode\_↵\_rx1\_, const std::string &gain\_mode\_rx2\_, double rf\_gain\_rx1\_, double rf\_gain\_rx2\_, bool quadrature\_, bool rfdc\_, bool bbdc\_, std::string filter\_source\_, std::string filter\_filename\_, float Fpass\_, float Fstop\_)
- bool **config\_ad9361\_rx\_remote** (const std::string &remote\_host, uint64\_t bandwidth\_, uint64\_t sample\_↵\_rate\_, uint64\_t freq\_, const std::string &rf\_port\_select\_, bool rx1\_enable\_, bool rx2\_enable\_, const std\_↵\_::string &gain\_mode\_rx1\_, const std::string &gain\_mode\_rx2\_, double rf\_gain\_rx1\_, double rf\_gain\_rx2\_, bool quadrature\_, bool rfdc\_, bool bbdc\_, std::string filter\_source\_, std::string filter\_filename\_, float Fpass\_, float Fstop\_)
- bool **config\_ad9361\_lo\_local** (uint64\_t bandwidth\_, uint64\_t sample\_rate\_, uint64\_t freq\_rf\_tx\_hz\_, double tx\_attenuation\_db\_, int64\_t freq\_dds\_tx\_hz\_, double scale\_dds\_dbfs\_, double phase\_dds\_deg\_)

- bool **config\_ad9361\_lo\_remote** (const std::string &remote\_host, uint64\_t bandwidth\_, uint64\_t sample\_rate\_, uint64\_t freq\_rf\_tx\_hz\_, double tx\_attenuation\_db\_, int64\_t freq\_dds\_tx\_hz\_, double scale\_dds\_dbfs\_, double phase\_dds\_deg\_)
- bool **ad9361\_disable\_lo\_remote** (const std::string &remote\_host)
- bool **ad9361\_disable\_lo\_local** ()
- bool **load\_fir\_filter** (std::string &filter, struct iio\_device \*phy)
- bool **disable\_ad9361\_rx\_local** ()
- bool **disable\_ad9361\_rx\_remote** (const std::string &remote\_host)

### 11.384.1 Detailed Description

An Analog Devices AD9361 front-end configuration library wrapper for configure some functions via iiod link.

Author

Javier Arribas, jarribas(at)cttc.es

This file contains information taken from librtlsdr: <https://git.osmocom.org/rtl-sdr>  
 GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [ad9361\\_manager.h](#).

## 11.385 ad9361\_manager.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file ad9361_manager.h
00003  * \brief An Analog Devices AD9361 front-end configuration library wrapper for configure some
00004  * \author Javier Arribas, jarribas(at)cttc.es
00005  *
00006  * This file contains information taken from librtlsdr:
00007  * https://git.osmocom.org/rtl-sdr
00008  *
00009  * -----
00010  *
00011  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00012  * This file is part of GNSS-SDR.
00013  *
00014  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00015  * SPDX-License-Identifier: GPL-3.0-or-later
00016  *
00017  * -----
00018  */
00019
00020 #ifndef GNSS_SDR_AD9361_MANAGER_H
00021 #define GNSS_SDR_AD9361_MANAGER_H
00022
00023 #include <iio.h>
00024 #include <cstdint>
00025 #include <string>
00026
00027 /** \addtogroup Signal_Source
00028  * \{ */
00029 /** \addtogroup Signal_Source_libs signal_source_libs
00030  * Library with utilities for signal sources.
00031  * \{ */
00032
00033 #define FIR_BUF_SIZE 8192
00034
00035 static const std::string RX_DEV_A = "ad9361-phy"; // one or two AD9361s are present
00036 static const std::string RX_DEV_B = "ad9361-phy-B"; // one or two AD9361s are present
00037 static const std::string RX_STREAM_DEV = "cf-ad9361-lpc"; // one AD9361 is present
00038 static const std::string RX_STREAM_DEV_A = "cf-ad9361-A"; // two AD9361s are present
00039 static const std::string RX_STREAM_DEV_B = "cf-ad9361-B"; // two AD9361s are present
00040 static const std::string TX_STREAM_DEV = "cf-ad9361-dds-core-lpc";
00041
00042 /* RX is input, TX is output */
00043 enum iodev
00044 {
00045     RX,
00046     TX
00047 };
00048
00049 /* common RX and TX streaming params */
00050 struct stream_cfg
```

```

00051 {
00052     int64_t bw_hz;           // Analog bandwidth in Hz
00053     int64_t fs_hz;          // Baseband sample rate in Hz
00054     int64_t lo_hz;          // Local oscillator frequency in Hz
00055     const char *rfport;     // Port name
00056 };
00057
00058 /* check return value of attr_write function */
00059 void errchk(int v, const char *what);
00060
00061 /* write attribute: int64_t int */
00062 void wr_ch_lli(struct iio_channel *chn, const char *what, int64_t val);
00063
00064 /* write attribute: string */
00065 void wr_ch_str(struct iio_channel *chn, const char *what, const char *str);
00066
00067 /* returns ad9361 phy device */
00068 struct iio_device *get_ad9361_phy(struct iio_context *ctx);
00069
00070 /* finds AD9361 streaming IIO devices */
00071 bool get_ad9361_stream_dev(struct iio_context *ctx, enum iodev d, struct iio_device **dev);
00072
00073 /* finds AD9361 streaming IIO channels */
00074 bool get_ad9361_stream_ch(struct iio_context *ctx, enum iodev d, struct iio_device *dev, int chid,
    struct iio_channel **chn);
00075
00076 /* finds AD9361 phy IIO configuration channel with id chid */
00077 bool get_phy_chan(struct iio_context *ctx, enum iodev d, int chid, struct iio_channel **chn);
00078
00079 /* finds AD9361 local oscillator IIO configuration channels */
00080 bool get_lo_chan(struct iio_context *ctx, enum iodev d, struct iio_channel **chn);
00081
00082 /* applies streaming configuration through IIO */
00083 bool cfg_ad9361_streaming_ch(struct iio_context *ctx, struct stream_cfg *cfg, enum iodev type, int
    chid);
00084
00085 bool config_ad9361_rx_local(uint64_t bandwidth_,
00086     uint64_t sample_rate_,
00087     uint64_t freq0_,
00088     uint64_t freq1_,
00089     const std::string &rf_port_select_,
00090     bool rx1_enable_,
00091     bool rx2_enable_,
00092     const std::string &gain_mode_rx1_,
00093     const std::string &gain_mode_rx2_,
00094     double rf_gain_rx1_,
00095     double rf_gain_rx2_,
00096     bool quadrature_,
00097     bool rf_dc_,
00098     bool bbdc_,
00099     std::string filter_source_,
00100     std::string filter_filename_,
00101     float Fpass_,
00102     float Fstop_);
00103
00104 bool config_ad9361_rx_remote(const std::string &remote_host,
00105     uint64_t bandwidth_,
00106     uint64_t sample_rate_,
00107     uint64_t freq_,
00108     const std::string &rf_port_select_,
00109     bool rx1_enable_,
00110     bool rx2_enable_,
00111     const std::string &gain_mode_rx1_,
00112     const std::string &gain_mode_rx2_,
00113     double rf_gain_rx1_,
00114     double rf_gain_rx2_,
00115     bool quadrature_,
00116     bool rf_dc_,
00117     bool bbdc_,
00118     std::string filter_source_,
00119     std::string filter_filename_,
00120     float Fpass_,
00121     float Fstop_);
00122
00123 bool config_ad9361_lo_local(uint64_t bandwidth_,
00124     uint64_t sample_rate_,
00125     uint64_t freq_rf_tx_hz_,
00126     double tx_attenuation_db_,
00127     int64_t freq_dds_tx_hz_,
00128     double scale_dds_dbfs_,
00129     double phase_dds_deg_);
00130
00131 bool config_ad9361_lo_remote(const std::string &remote_host,
00132     uint64_t bandwidth_,
00133     uint64_t sample_rate_,
00134     uint64_t freq_rf_tx_hz_,
00135     double tx_attenuation_db_,

```

```

00136     int64_t freq_dds_tx_hz_,
00137     double scale_dds_dbfs_,
00138     double phase_dds_deg_);
00139
00140 bool ad9361_disable_lo_remote(const std::string &remote_host);
00141
00142 bool ad9361_disable_lo_local();
00143
00144 bool load_fir_filter(std::string &filter, struct iio_device *phy);
00145
00146 bool disable_ad9361_rx_local();
00147
00148 bool disable_ad9361_rx_remote(const std::string &remote_host);
00149
00150
00151 /** \} */
00152 /** \} */
00153 #endif // GNSS_SDR_AD9361_MANAGER_H

```

## 11.386 ad936x\_iio\_custom.h File Reference

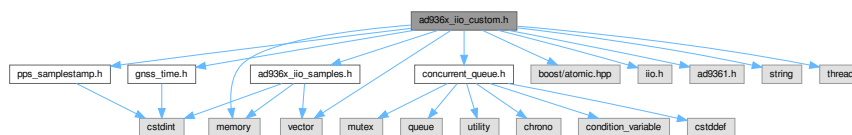
A direct IIO custom front-end driver for the AD936x AD front-end family with special FPGA custom functionalities.

```

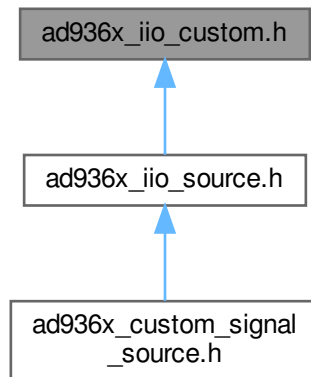
#include "ad936x_iio_samples.h"
#include "concurrent_queue.h"
#include "gnss_time.h"
#include "pps_samplestamp.h"
#include <boost/atomic.hpp>
#include <iio.h>
#include <ad9361.h>
#include <memory>
#include <string>
#include <thread>
#include <vector>

```

Include dependency graph for ad936x\_iio\_custom.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [ad936x\\_iio\\_custom](#)

### 11.386.1 Detailed Description

A direct IIO custom front-end driver for the AD936x AD front-end family with special FPGA custom functionalities.

#### 11.386.1.1 autotoc\_md383

##### Author

Javier Arribas, jarribas(at)cttc.es

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [ad936x\\_iio\\_custom.h](#).

## 11.387 ad936x\_iio\_custom.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file ad936x_iio_custom.h
00003  * \brief A direct IIO custom front-end driver for the AD936x AD front-end
00004  * family with special FPGA custom functionalities.
00005  * \author Javier Arribas, jarribas(at)cttc.es
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_AD936X_IIO_CUSTOM_H
00019 #define GNSS_SDR_AD936X_IIO_CUSTOM_H
00020
00021 #include "ad936x_iio_samples.h"
00022 #include "concurrent_queue.h"
00023 #include "gnss_time.h"

```

```

00024 #include "pps_samplestamp.h"
00025 #include <boost/atomic.hpp>
00026 #include <iio.h>
00027 #include <ad9361.h> // multichip sync and high level functions
00028 #include <memory>
00029 #include <string>
00030 #include <thread>
00031 #include <vector>
00032
00033 /** \addtogroup Signal_Source
00034 * \{ */
00035 /** \addtogroup Signal_Source_libs
00036 * \{ */
00037
00038
00039 class ad936x_iio_custom
00040 {
00041 public:
00042     ad936x_iio_custom(int debug_level_, int log_level_);
00043     virtual ~ad936x_iio_custom();
00044     bool initialize_device(const std::string &pluto_device_uri, const std::string &board_type);
00045
00046     bool init_config_ad9361_rx(long long bandwidth_,
00047                               long long sample_rate_,
00048                               long long freq_,
00049                               const std::string &rf_port_select_,
00050                               const std::string &rf_filter_,
00051                               const std::string &gain_mode_rx0_,
00052                               const std::string &gain_mode_rx1_,
00053                               double rf_gain_rx0_,
00054                               double rf_gain_rx1_,
00055                               bool enable_ch0,
00056                               bool enable_ch1,
00057                               long long freq_2ch,
00058                               double lo_attenuation_db_,
00059                               bool high_side_lo_,
00060                               int tx_lo_channel_);
00061
00062     bool calibrate(int ch, double bw_hz);
00063
00064     double get_rx_gain(int ch_num);
00065     bool setRXGain(int ch_num, const std::string &gain_mode, double gain_db);
00066
00067     bool set_antenna_port(int ch, int antenna_idx);
00068     double get_frequency(int ch);
00069     bool set_frequency(int ch, double freq_hz);
00070
00071     bool start_sample_rx(bool ppsmode);
00072     void stop_record();
00073
00074     void set_gnss_time_queue(std::shared_ptr<Concurrent_Queue<GnssTime>> queue);
00075     void set_pps_samplestamp_queue(std::shared_ptr<Concurrent_Queue<PpsSamplestamp>> queue);
00076
00077     bool get_rx_frequency(long long &freq_hz);
00078     bool set_rx_frequency(long long freq_hz);
00079     bool read_die_temp(double &temp_c);
00080
00081     void pop_sample_buffer(std::shared_ptr<ad936x_iio_samples> &current_buffer);
00082
00083     void push_sample_buffer(std::shared_ptr<ad936x_iio_samples> &current_buffer);
00084     int n_channels;
00085
00086 private:
00087     std::shared_ptr<Concurrent_Queue<GnssTime>> GnssTime_queue;
00088     std::shared_ptr<Concurrent_Queue<PpsSamplestamp>> Pps_queue;
00089     bool check_device();
00090     bool get_iio_param(iio_device *dev, const std::string &param, std::string &value);
00091     void configure_params(struct iio_device *phy,
00092                          const std::vector<std::string> &params);
00093     void set_params_rx(struct iio_device *phy_device,
00094                       unsigned long long frequency,
00095                       unsigned long samplerate, unsigned long bandwidth,
00096                       bool quadrature, bool rfdc, bool bbdc,
00097                       const std::string &gain1, double gain1_value,
00098                       const std::string &gain2, double gain2_value,
00099                       const std::string &port_select);
00100
00101     bool config_ad9361_dds(uint64_t freq_rf_tx_hz_,
00102                           double tx_attenuation_db_,
00103                           int64_t freq_dds_tx_hz_,
00104                           double scale_dds_,
00105                           double phase_dds_deg_,
00106                           int channel);
00107
00108     void get_PPS_timestamp();
00109     void capture(const std::vector<std::string> &channels);
00110

```



```

00111     bool select_rf_filter(const std::string &rf_filter);
00112
00113     void monitor_thread_fn();
00114
00115     void PlutoTxEnable(bool txon);
00116     void setPlutoGpo(int p);
00117
00118     // Device structure
00119     struct iio_context *ctx;
00120     struct iio_device *phy;
00121     struct iio_device *stream_dev;
00122     struct iio_device *dds_dev;
00123
00124     std::mutex mtx;
00125     std::condition_variable cv;
00126
00127     boost::atomic<bool> receive_samples;
00128
00129     boost::atomic<bool> fpga_overflow;
00130     // using queues of smart pointers to preallocated buffers
00131     Concurrent_Queue<std::shared_ptr<ad936x_iio_samples>> free_buffers;
00132     Concurrent_Queue<std::shared_ptr<ad936x_iio_samples>> used_buffers;
00133
00134     std::thread capture_samples_thread;
00135     std::thread overflow_monitor_thread;
00136     std::thread capture_time_thread;
00137
00138     // stream
00139     uint64_t sample_rate_sps;
00140     int debug_level;
00141     int log_level;
00142     bool PPS_mode;
00143 };
00144
00145 /** \} */
00146 /** \} */
00147 #endif // GNSS_SDR_AD936X_IIO_CUSTOM_H

```

## 11.388 ad936x\_iio\_samples.h File Reference

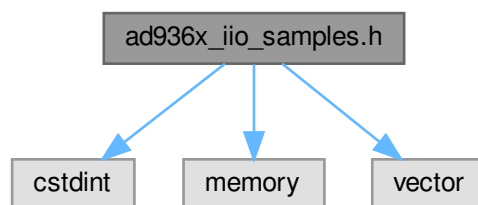
A class that holds a custom sample buffer for Analog Devices AD936x family front-ends.

```
#include <stdint>
```

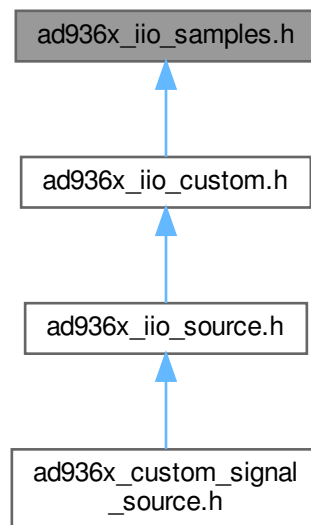
```
#include <memory>
```

```
#include <vector>
```

Include dependency graph for ad936x\_iio\_samples.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [ad936x\\_iio\\_samples](#)

## Macros

- `#define IIO_DEFAULTAD936XAPIFIFOSIZE_SAMPLES 32768 * 4`
- `#define IIO_INPUTRAMFIFOSIZE 256`

### 11.388.1 Detailed Description

A class that holds a custom sample buffer for Analog Devices AD936x family front-ends.

#### Author

Javier Arribas, jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [ad936x\\_iio\\_samples.h](#).

---

### 11.388.2 Macro Definition Documentation

#### 11.388.2.1 IIO\_DEFAULTAD936XAPIFIFOSIZE\_SAMPLES

```
#define IIO_DEFAULTAD936XAPIFIFOSIZE_SAMPLES 32768 * 4
```

Definition at line 21 of file [ad936x\\_iio\\_samples.h](#).

#### 11.388.2.2 IIO\_INPUTRAMFIFOSIZE

```
#define IIO_INPUTRAMFIFOSIZE 256
```

Definition at line 22 of file [ad936x\\_iio\\_samples.h](#).

## 11.389 ad936x\_iio\_samples.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file ad936x_iio_samples.h
00003  * \brief A class that holds a custom sample buffer for Analog Devices AD936x family front-ends.
00004  * \author Javier Arribas, jarribas(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_AD936X_IIO_SAMPLES_H
00019 #define GNSS_SDR_AD936X_IIO_SAMPLES_H
00020
00021 #define IIO_DEFAULTAD936XAPIFIFO_SIZE_SAMPLES 32768 * 4
00022 #define IIO_INPUTRAMFIFO_SIZE 256
00023
00024 #include <stdint>
00025 #include <memory>
00026 #include <vector>
00027
00028 /** \addtogroup Signal_Source
00029 * \{ */
00030 /** \addtogroup Signal_Source_libs
00031 * \{ */
00032
00033 class ad936x_iio_samples
00034 {
00035 public:
00036     ad936x_iio_samples() = default;
00037     uint32_t n_bytes{0};
00038     uint32_t n_interleaved_iq_samples{0};
00039     uint16_t n_channels{0};
00040     uint16_t step_bytes{0};
00041     char buffer[IIO_DEFAULTAD936XAPIFIFO_SIZE_SAMPLES * 4 * 4]; // max 16 bits samples per buffer (4
        channels, 2-bytes per I + 2-bytes per Q)
00042 };
00043
00044 /** \} */
00045 /** \} */
00046 #endif

```

## 11.390 fpga\_buffer\_monitor.h File Reference

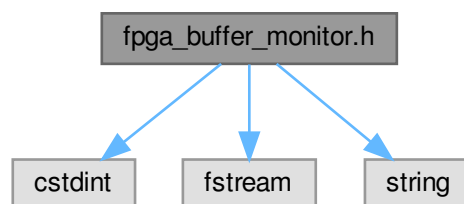
Check receiver buffer overflow and monitor the status of the receiver buffers.

```

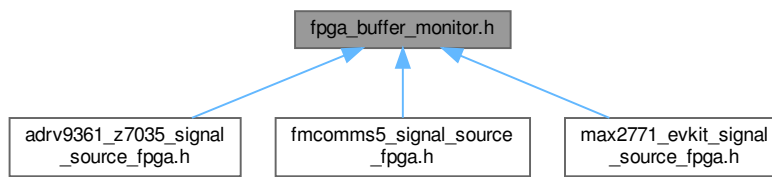
#include <stdint>
#include <fstream>
#include <string>

```

Include dependency graph for fpga\_buffer\_monitor.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Fpga\\_buffer\\_monitor](#)

*Class that checks the receiver buffer overflow flags and monitors the status of the receiver buffers.*

### 11.390.1 Detailed Description

Check receiver buffer overflow and monitor the status of the receiver buffers.

#### Authors

- Marc Majoral, 2021. mmajoral(at)cttc.es

Class that checks the receiver buffer overflow flags and monitors the status of the receiver buffers.

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [fpga\\_buffer\\_monitor.h](#).

### 11.391 fpga\_buffer\_monitor.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file fpga_buffer_monitor.h
00003  * \brief Check receiver buffer overflow and monitor the status of the receiver
00004  * buffers.
00005  * \authors
00006  * <ul>
00007  *   <li> Marc Majoral, 2021. mmajoral(at)cttc.es
00008  * </ul>
00009  *
00010  * Class that checks the receiver buffer overflow flags and monitors the status
00011  * of the receiver buffers.
00012  *
00013  *
00014  * -----
00015  *
00016  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00017  * This file is part of GNSS-SDR.
00018  *
00019  * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)
00020  * SPDX-License-Identifier: GPL-3.0-or-later
00021  *
00022  * -----
00023  */
00024
00025 #ifndef GNSS_SDR_FPGA_BUFFER_MONITOR_H
00026 #define GNSS_SDR_FPGA_BUFFER_MONITOR_H
00027
00028 #include <stdint> // for int32_t
00029 #include <fstream> // for std::ofstream
00030 #include <string> // for std::string
00031
00032 /** \addtogroup Signal_Source
00033  * \{ */
00034 /** \addtogroup Signal_Source_libs
00035  * \{ */
00036

```

```

00037
00038 /*!
00039  * \brief Class that checks the receiver buffer overflow flags and monitors the
00040  * status of the receiver buffers.
00041  */
00042 class Fpga_buffer_monitor
00043 {
00044 public:
00045 /*!
00046  * \brief Constructor
00047  */
00048 explicit Fpga_buffer_monitor(uint32_t num_freq_bands,
00049     bool dump,
00050     std::string dump_filename);
00051 // explicit Fpga_buffer_monitor(const std::string& device_name,
00052 //     uint32_t num_freq_bands,
00053 //     bool dump,
00054 //     std::string dump_filename);
00055
00056 /*!
00057  * \brief Destructor
00058  */
00059 ~Fpga_buffer_monitor();
00060
00061 /*!
00062  * \brief This function checks buffer overflow and monitors the FPGA buffer status
00063  */
00064 void check_buffer_overflow_and_monitor_buffer_status();
00065
00066 private:
00067     const std::string BUFFER_MONITOR_DEVICE_NAME = std::string("buffer_monitor"); // buffer monitor
00068     device name
00069     static const size_t FPGA_PAGE_SIZE = 0x1000;
00070     static const uint32_t test_register_writeval = 0x55AA;
00071     static const uint32_t num_samples_per_buffer_element = 2;
00072     // write addresses
00073     static const uint32_t reset_overflow_flags_and_max_buff_size_reg_addr = 0;
00074     // read-write addresses
00075     static const uint32_t test_reg_addr = 7;
00076     // read addresses
00077     static const uint32_t current_buff_occ_freq_band_0_reg_addr = 0;
00078     static const uint32_t current_buff_occ_freq_band_1_reg_addr = 1;
00079     static const uint32_t max_buff_occ_freq_band_0_reg_addr = 2;
00080     static const uint32_t max_buff_occ_freq_band_1_reg_addr = 3;
00081     static const uint32_t overflow_flags_reg_addr = 4;
00082     // FPGA-related constants
00083     static const uint32_t overflow_freq_band_0_bit_pos = 1;
00084     static const uint32_t overflow_freq_band_1_bit_pos = 2;
00085
00086     int32_t buffer_monitor_test_register();
00087     void close_device();
00088
00089     std::string d_dump_filename;
00090     std::ofstream d_dump_file;
00091
00092     volatile unsigned* d_map_base; // driver memory map corresponding to the FPGA buffer monitor
00093     int d_device_descriptor; // driver descriptor corresponding to the FPGA buffer monitor
00094
00095     uint32_t d_num_freq_bands;
00096
00097     uint32_t d_max_buff_occ_freq_band_0;
00098     uint32_t d_max_buff_occ_freq_band_1;
00099
00100     bool d_dump;
00101 };
00102
00103 /** \} */
00104 /** \} */
00105 #endif // GNSS_SDR_FPGA_BUFFER_MONITOR_H

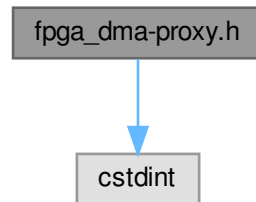
```

## 11.392 fpga\_dma-proxy.h File Reference

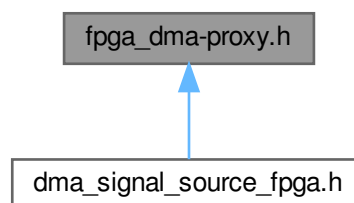
FPGA DMA control. This code is based in the Xilinx DMA proxy test application: <https://github.com/Xilinx-Wiki-Projects/software-prototypes/tree/master/linux-user-space-dma/Software>.

```
#include <stdint>
```

Include dependency graph for fpga\_dma-proxy.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Fpga\\_DMA](#)

*Class that controls the switch DMA in the FPGA.*

## Enumerations

- enum **proxy\_status** { **PROXY\_NO\_ERROR** = 0 , **PROXY\_BUSY** = 1 , **PROXY\_TIMEOUT** = 2 , **PROXY\_↔  
ERROR** = 3 }

## Variables

- int8\_t [buffer](#) [DMA\_MAX\_BUFFER\_SIZE]
- enum proxy\_status **status**
- unsigned int [length](#)

### 11.392.1 Detailed Description

FPGA DMA control. This code is based in the Xilinx DMA proxy test application: [https://github.com/↔  
Xilinx-Wiki-Projects/software-prototypes/tree/master/linux-user-space-dma/↔  
Software](https://github.com/Xilinx-Wiki-Projects/software-prototypes/tree/master/linux-user-space-dma/Software).

**Author**

Marc Majoral, mmajoral(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [fpga\\_dma-proxy.h](#).

---

**11.392.2 Enumeration Type Documentation****11.392.2.1 proxy\_status**

enum proxy\_status

Definition at line 1 of file [fpga\\_dma-proxy.h](#).

**11.392.3 Variable Documentation****11.392.3.1 buffer**

int8\_t buffer[DMA\_MAX\_BUFFER\_SIZE]

Definition at line 0 of file [fpga\\_dma-proxy.h](#).

**11.392.3.2 length**

unsigned int length

Definition at line 8 of file [fpga\\_dma-proxy.h](#).

**11.393 fpga\_dma-proxy.h**

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file fpga_dma-proxy.h
00003  * \brief FPGA DMA control. This code is based in the Xilinx DMA proxy test application:
00004  *
00005  * \author Marc Majoral, mmajoral(at)cttc.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_FPGA_DMA_PROXY_H
00019 #define GNSS_SDR_FPGA_DMA_PROXY_H
00020
00021 #include <stdint> // for std::int8_t
00022
00023 /*!
00024  * \brief Class that controls the switch DMA in the FPGA
00025  */
00026 class Fpga_DMA
00027 {
00028 public:
00029     /*!
00030      * \brief Default constructor.
00031      */
00032     Fpga_DMA() = default;
00033
00034     /*!
00035      * \brief Default destructor.
00036      */
00037     ~Fpga_DMA() = default;
00038
00039     /*!
00040      * \brief Open the DMA device driver.
00041      */
00042     int DMA_open(void);
00043
00044     /*!

```

```

00045     * \brief Obtain DMA buffer address.
00046     */
00047     int8_t *get_buffer_address(void); // NOLINT(readability-make-member-function-const)
00048
00049     /*!
00050     * \brief Transfer DMA data
00051     */
00052     int DMA_write(int nbytes) const;
00053
00054     /*!
00055     * \brief Close the DMA device driver
00056     */
00057     int DMA_close(void) const;
00058
00059 private:
00060     static const uint32_t DMA_MAX_BUFFER_SIZE = (128 * 1024); /* must match driver exactly */
00061     static const uint32_t TX_BUFFER_COUNT = 1;
00062
00063     // channel buffer structure
00064     struct channel_buffer
00065     {
00066         int8_t buffer[DMA_MAX_BUFFER_SIZE];
00067         enum proxy_status
00068         {
00069             PROXY_NO_ERROR = 0,
00070             PROXY_BUSY = 1,
00071             PROXY_TIMEOUT = 2,
00072             PROXY_ERROR = 3
00073         } status;
00074         unsigned int length;
00075     } __attribute__((aligned(1024))); /* 64 byte alignment required for DMA, but 1024 handy for
00076     viewing memory */
00077
00078     // internal DMA channel data structure
00079     struct channel
00080     {
00081         struct channel_buffer *buf_ptr;
00082         int fd;
00083     };
00084     channel tx_channel;
00085 };
00086 #endif // GNSS_SDR_FPGA_DMA_PROXY_H

```

## 11.394 fpga\_dynamic\_bit\_selection.h File Reference

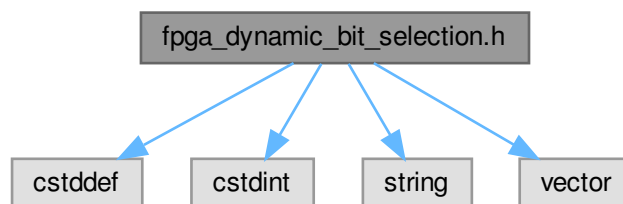
Dynamic bit selection in the received signal.

```

#include <cstdint>
#include <string>
#include <vector>

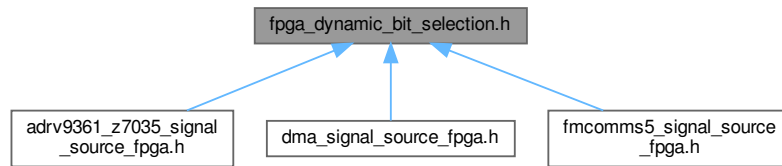
```

Include dependency graph for fpga\_dynamic\_bit\_selection.h:





This graph shows which files directly or indirectly include this file:



## Classes

- class [Fpga\\_dynamic\\_bit\\_selection](#)

*Class that controls the switch in the FPGA, which connects the FPGA acquisition and multicorrelator modules to either the DMA or the Analog Front-End.*

### 11.394.1 Detailed Description

Dynamic bit selection in the received signal.

#### Authors

- Marc Majoral, 2020. mmajoral(at)cttc.es

Class that controls the Dynamic Bit Selection in the FPGA.

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [fpga\\_dynamic\\_bit\\_selection.h](#).

## 11.395 fpga\_dynamic\_bit\_selection.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file fpga_dynamic_bit_selection.h
00003  * \brief Dynamic bit selection in the received signal.
00004  * \authors <ul>
00005  *      <li> Marc Majoral, 2020. mmajoral(at)cttc.es
00006  *      </li>
00007  *
00008  * Class that controls the Dynamic Bit Selection in the FPGA.
00009  *
00010  *
00011  * -----
00012  *
00013  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00014  * This file is part of GNSS-SDR.
00015  *
00016  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00017  * SPDX-License-Identifier: GPL-3.0-or-later
00018  *
00019  * -----
00020  */
00021
00022 #ifndef GNSS_SDR_FPGA_DYNAMIC_BIT_SELECTION_H
00023 #define GNSS_SDR_FPGA_DYNAMIC_BIT_SELECTION_H
00024
00025 #include <stddef>
00026 #include <stdint>
00027 #include <string>
00028 #include <vector>
00029
00030 /** \addtogroup Signal_Source
00031  * \{ */
00032 /** \addtogroup Signal_Source_libs
00033  * \{ */
00034
00035
  
```

```

00036  /*!
00037  * \brief Class that controls the switch in the FPGA, which connects the FPGA acquisition and
00038  *        multicorrelator modules to
00039  *        either the DMA or the Analog Front-End.
00040  */
00041  class Fpga_dynamic_bit_selection
00042  {
00043  public:
00044      /*!
00045       * \brief Constructor
00046       */
00047      explicit Fpga_dynamic_bit_selection(bool enable_rx1_band, bool enable_rx2_band);
00048      /*!
00049       * \brief Destructor
00050       */
00051      ~Fpga_dynamic_bit_selection();
00052      /*!
00053       * \brief This function configures the switch in the FPGA
00054       */
00055      void bit_selection(void);
00056 private:
00057     const std::string dyn_bit_sel_device_name = std::string("dynamic_bits_selector"); // Switch
00058     dynamic_bit_selector device_name
00059     static const size_t FPGA_PAGE_SIZE = 0x1000;
00060     static const uint32_t Num_bits_ADC = 12; // Number of bits in
00061     the ADC
00062     static const uint32_t Num_bits_FPGA = 4; // Number of bits
00063     after the bit selection
00064     static const uint32_t shift_out_bits_default = Num_bits_ADC - Num_bits_FPGA; // take the most
00065     significant bits by default
00066     static const uint32_t shift_out_bits_min = 0; // minimum possible
00067     value for the bit selection
00068     static const uint32_t shift_out_bit_max = Num_bits_ADC - Num_bits_FPGA; // maximum possible
00069     value for the bit selection
00070     // received signal power thresholds for the bit selection
00071     // the received signal power is estimated as the averaged squared absolute value of the received
00072     signal samples
00073     static const uint32_t Power_Threshold_High = 9000;
00074     static const uint32_t Power_Threshold_Low = 3000;
00075     void open_device(volatile unsigned *d_map_base, int &d_dev_descr, int freq_band);
00076     void bit_selection_per_rf_band(volatile unsigned *d_map_base, uint32_t shift_out_bits);
00077     void close_device(volatile unsigned *d_map_base, int &d_dev_descr);
00078     volatile unsigned *d_map_base_freq_band_1;
00079     volatile unsigned *d_map_base_freq_band_2;
00080     int d_dev_descr_freq_band_1;
00081     int d_dev_descr_freq_band_2;
00082     uint32_t d_shift_out_bits_freq_band_1;
00083     uint32_t d_shift_out_bits_freq_band_2;
00084     bool d_enable_rx1_band;
00085     bool d_enable_rx2_band;
00086 };
00087 /** \} */
00088 /** \} */
00089 #endif // GNSS_SDR_FPGA_DYNAMIC_BIT_SELECTION_H

```

## 11.396 fpga\_spidev.h File Reference

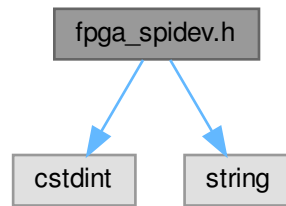
FPGA SPI control.

```

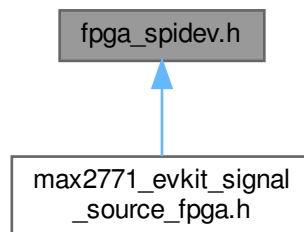
#include <stdint>
#include <string>

```

Include dependency graph for fpga\_spidev.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Fpga\\_spidev](#)

### 11.396.1 Detailed Description

FPGA SPI control.

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [fpga\\_spidev.h](#).

## 11.397 fpga\_spidev.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002   * \file fpga_spidev.h
00003   * \brief FPGA SPI control.
00004   *
00005   * -----
00006   *
00007   * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00008   * This file is part of GNSS-SDR.
00009   *
00010   * Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors)
00011   * SPDX-License-Identifier: GPL-3.0-or-later
00012   *
00013   * -----
00014   */
  
```

```

00015
00016 #ifndef GNSS_SDR_FPGA_SPIDEV_H
00017 #define GNSS_SDR_FPGA_SPIDEV_H
00018
00019 #include <stdint>
00020 #include <string>
00021
00022 class Fpga_spidev
00023 {
00024 public:
00025     /*!
00026      * \brief Default constructor.
00027      */
00028     Fpga_spidev() = default;
00029
00030     /*!
00031      * \brief Default destructor.
00032      */
00033     ~Fpga_spidev() = default;
00034
00035     /*!
00036      * \brief write a register through the SPI.
00037      */
00038     int write_reg32(char addr, uint32_t data);
00039
00040     /*!
00041      * \brief read a register through the SPI.
00042      */
00043     int read_reg32(uint8_t addr, uint32_t* copy_to);
00044     /*!
00045      * \brief Open the SPI device driver.
00046      */
00047     int SPI_open(void);
00048
00049     /*!
00050      * \brief Close the SPI device driver
00051      */
00052     int SPI_close(void) const;
00053 private:
00054     static const uint32_t SPI_SPEED = 250000;
00055     const std::string SPI_DEVICE_NAME = std::string("/dev/spidev2.0"); // Switch UIO device name
00056     int d_fd;
00057 };
00058
00059 #endif // GNSS_SDR_FPGA_SPIDEV_H

```

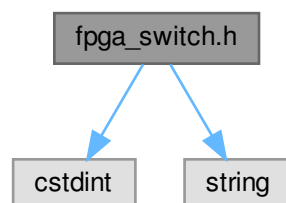
## 11.398 fpga\_switch.h File Reference

Switch that connects the HW accelerator queues to the analog front end or the DMA.

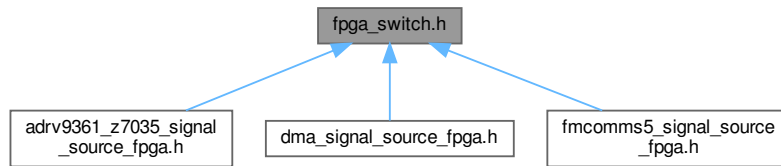
```
#include <stdint>
```

```
#include <string>
```

Include dependency graph for fpga\_switch.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Fpga\\_Switch](#)

*Class that controls the switch in the FPGA, which connects the FPGA acquisition and multicorrelator modules to either the DMA or the Analog Front-End.*

### 11.398.1 Detailed Description

Switch that connects the HW accelerator queues to the analog front end or the DMA.

#### Authors

- Marc Majoral, 2019. mmajoral(at)cttc.cat
- Javier Arribas, 2016. jarribas(at)cttc.es

Class that controls a switch in the FPGA

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [fpga\\_switch.h](#).

## 11.399 fpga\_switch.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file fpga_switch.h
00003  * \brief Switch that connects the HW accelerator queues to the analog front end or the DMA.
00004  * \authors <ul>
00005  *     <li> Marc Majoral, 2019. mmajoral(at)cttc.cat
00006  *     <li> Javier Arribas, 2016. jarribas(at)cttc.es
00007  * </ul>
00008  *
00009  * Class that controls a switch in the FPGA
00010  *
00011  *
00012  * -----
00013  *
00014  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00015  * This file is part of GNSS-SDR.
00016  *
00017  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00018  * SPDX-License-Identifier: GPL-3.0-or-later
00019  *
00020  * -----
00021  */
00022
00023 #ifndef GNSS_SDR_FPGA_SWITCH_H
00024 #define GNSS_SDR_FPGA_SWITCH_H
00025
00026 #include <stdint>
00027 #include <string>
00028
00029 /** \addtogroup Signal_Source
00030  * \{ */
00031 /** \addtogroup Signal_Source_libs
00032  * \{ */
00033

```

```

00034
00035 /*!
00036 * \brief Class that controls the switch in the FPGA, which connects the FPGA acquisition and
00037 multicorrelator modules to
00038 * either the DMA or the Analog Front-End.
00039 */
00039 class Fpga_Switch
00040 {
00041 public:
00042 /*!
00043 * \brief Constructor
00044 */
00045 Fpga_Switch(void);
00046 /*!
00047 * \brief Destructor
00048 */
00049 ~Fpga_Switch();
00050
00051 /*!
00052 * \brief This function configures the switch in the FPGA
00053 */
00054 void set_switch_position(int32_t switch_position);
00055
00056 private:
00057 const std::string SWITCH_DEVICE_NAME = std::string("AXIS_Switch_v1_0_0"); // Switch UIO device
00058 name
00058 static const size_t FPGA_PAGE_SIZE = 0x1000;
00059 static const uint32_t TEST_REGISTER_TRACK_WRITEVAL = 0x55AA;
00060 static const uint32_t MAX_LENGTH_DEVICEIO_NAME = 50;
00061
00062 // private functions
00063 unsigned fpga_switch_test_register(unsigned writeval);
00064 void close_device(void);
00065
00066 volatile unsigned* d_map_base; // driver memory map
00067 int d_device_descriptor; // driver descriptor
00068 };
00069
00070
00071 ** \} */
00072 ** \} */
00073 #endif // GNSS_SDR_FPGA_SWITCH_H

```

## 11.400 gnss\_sdr\_timestamp.h File Reference

GNURadio block that adds to sample stream timestamp metadata information stored on a separated file.

```

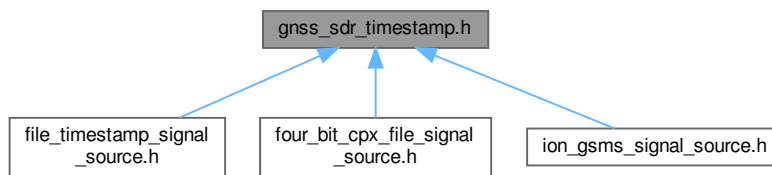
#include "gnss_block_interface.h"
#include "gnss_time.h"
#include <gnuradio/sync_block.h>
#include <gnuradio/types.h>
#include <pmt/pmt.h>
#include <cstdint>
#include <fstream>
#include <string>

```

Include dependency graph for gnss\_sdr\_timestamp.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Gnss\\_Sdr\\_Timestamp](#)

## Functions

- `gnss_shared_ptr< Gnss\_Sdr\_Timestamp > gnss_sdr_make_Timestamp` (`size_t sizeof_stream_item`, `std::string timestamp_file`, `double clock_offset_ms`, `int items_to_samples`)

## 11.400.1 Detailed Description

GNURadio block that adds to sample stream timestamp metadata information stored on a separated file.

### Author

Javier Arribas, 2021. [jarribas\(at\)cttc.es](mailto:jarribas(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gnss\\_sdr\\_timestamp.h](#).

---

## 11.401 gnss\_sdr\_timestamp.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file gnss_sdr_timestamp.h
00003  * \brief GNURadio block that adds to sample stream timestamp metadata information stored on a
00004  *        separated file
00005  * \author Javier Arribas, 2021. jarribas\(at\)cttc.es
00006  *
00007  * -----
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_GNSS_SDR_TIMESTAMP_H
00019 #define GNSS_SDR_GNSS_SDR_TIMESTAMP_H
00020
00021 #include "gnss_block_interface.h"
00022 #include "gnss_time.h"
00023 #include <gnuradio/sync_block.h> // for sync_block
00024 #include <gnuradio/types.h> // for gr_vector_const_void_star
00025 #include <pmt/pmt.h>
00026 #include <cstdint> // for size_t
00027 #include <fstream>
00028 #include <string>
00029
00030
00031 /** \addtogroup Signal_Source

```

```

00032  * \{ */
00033  /** \addtogroup Signal_Source_libs
00034  * \{ */
00035
00036
00037  class Gnss_Sdr_Timestamp;
00038
00039  gnss_shared_ptr<Gnss_Sdr_Timestamp> gnss_sdr_make_Timestamp(
00040      size_t sizeof_stream_item,
00041      std::string timestamp_file,
00042      double clock_offset_ms,
00043      int items_to_samples);
00044
00045
00046  class Gnss_Sdr_Timestamp : public gr::sync_block
00047  {
00048  public:
00049      int work(int noutput_items,
00050              gr_vector_const_void_star& input_items,
00051              gr_vector_void_star& output_items);
00052      bool start();
00053
00054  private:
00055      friend gnss_shared_ptr<Gnss_Sdr_Timestamp> gnss_sdr_make_Timestamp(
00056          size_t sizeof_stream_item,
00057          std::string timestamp_file,
00058          double clock_offset_ms,
00059          int items_to_samples);
00060
00061      Gnss_Sdr_Timestamp(size_t sizeof_stream_item,
00062                          std::string timestamp_file,
00063                          double clock_offset_ms,
00064                          int items_to_samples);
00065
00066      int64_t uint64diff(uint64_t first, uint64_t second);
00067      bool read_next_timetag();
00068      std::string d_timefile;
00069      std::fstream d_timefilestream;
00070      GnssTime next_timetag{};
00071      double d_clock_offset_ms;
00072      double d_fraction_ms_offset;
00073      double d_integer_ms_offset;
00074      int d_items_to_samples;
00075      uint64_t d_next_timetag_samplecount;
00076      bool d_get_next_timetag;
00077  };
00078
00079
00080  /** \} */
00081  /** \} */
00082  #endif // GNSS_SDR_GNSS_SDR_TIMESTAMP_H

```

## 11.402 gnss\_sdr\_valve.h File Reference

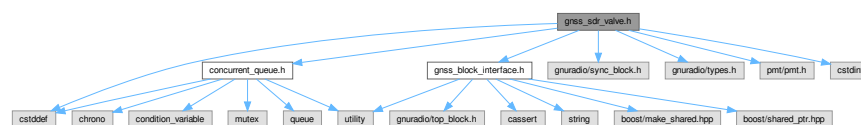
Interface of a GNU Radio block that sends a STOP message to the control queue right after a specific number of samples have passed through it.

```

#include "concurrent_queue.h"
#include "gnss_block_interface.h"
#include <gnuradio/sync_block.h>
#include <gnuradio/types.h>
#include <pmt/pmt.h>
#include <cstdint>
#include <cstdint>

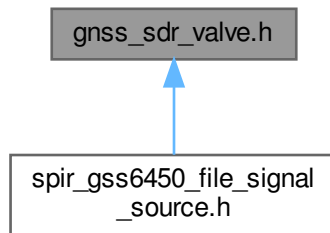
```

Include dependency graph for gnss\_sdr\_valve.h:





This graph shows which files directly or indirectly include this file:



## Classes

- class [Gnss\\_Sdr\\_Valve](#)

*Implementation of a GNU Radio block that sends a STOP message to the control queue right after a specific number of samples have passed through it.*

## Functions

- gnss\_shared\_ptr< [Gnss\\_Sdr\\_Valve](#) > **gnss\_sdr\_make\_valve** (size\_t sizeof\_stream\_item, uint64\_t nitems, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue)
- gnss\_shared\_ptr< [Gnss\\_Sdr\\_Valve](#) > **gnss\_sdr\_make\_valve** (size\_t sizeof\_stream\_item, uint64\_t nitems, [Concurrent\\_Queue](#)< pmt::pmt\_t > \*queue, bool stop\_flowgraph)

### 11.402.1 Detailed Description

Interface of a GNU Radio block that sends a STOP message to the control queue right after a specific number of samples have passed through it.

#### Author

Javier Arribas, 2018. jarribas(at)cttc.es

Carlos Aviles, 2010. carlos.avilesr(at)googlemail.com

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gnss\\_sdr\\_valve.h](#).

---

## 11.403 gnss\_sdr\_valve.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file gnss_sdr_valve.h
00003  * \brief Interface of a GNU Radio block that sends a STOP message to the
00004  * control queue right after a specific number of samples have passed through it.
00005  * \author Javier Arribas, 2018. jarribas(at)cttc.es
00006  * \author Carlos Aviles, 2010. carlos.avilesr(at)googlemail.com
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
  
```

```

00017  */
00018
00019
00020 #ifndef GNSS_SDR_GNSS_SDR_VALVE_H
00021 #define GNSS_SDR_GNSS_SDR_VALVE_H
00022
00023 #include "concurrent_queue.h"
00024 #include "gnss_block_interface.h"
00025 #include <gnuradio/sync_block.h> // for sync_block
00026 #include <gnuradio/types.h> // for gr_vector_const_void_star
00027 #include <pmt/pmt.h>
00028 #include <cstdint> // for size_t
00029 #include <cstdint>
00030
00031 /** \addtogroup Signal_Source
00032  * \{ */
00033 /** \addtogroup Signal_Source_libs
00034  * \{ */
00035
00036
00037 class Gnss_Sdr_Valve;
00038
00039 gnss_shared_ptr<Gnss_Sdr_Valve> gnss_sdr_make_valve(
00040     size_t sizeof_stream_item,
00041     uint64_t nitems,
00042     Concurrent_Queue<pmt::pmt_t>* queue);
00043
00044 gnss_shared_ptr<Gnss_Sdr_Valve> gnss_sdr_make_valve(
00045     size_t sizeof_stream_item,
00046     uint64_t nitems,
00047     Concurrent_Queue<pmt::pmt_t>* queue,
00048     bool stop_flowgraph);
00049
00050 /*!
00051  * \brief Implementation of a GNU Radio block that sends a STOP message to the
00052  * control queue right after a specific number of samples have passed through it.
00053  */
00054 class Gnss_Sdr_Valve : public gr::sync_block
00055 {
00056 public:
00057     void open_valve();
00058
00059     int work(int noutput_items,
00060             gr_vector_const_void_star& input_items,
00061             gr_vector_void_star& output_items);
00062
00063 private:
00064     friend gnss_shared_ptr<Gnss_Sdr_Valve> gnss_sdr_make_valve(
00065         size_t sizeof_stream_item,
00066         uint64_t nitems,
00067         Concurrent_Queue<pmt::pmt_t>* queue);
00068
00069     friend gnss_shared_ptr<Gnss_Sdr_Valve> gnss_sdr_make_valve(
00070         size_t sizeof_stream_item,
00071         uint64_t nitems,
00072         Concurrent_Queue<pmt::pmt_t>* queue,
00073         bool stop_flowgraph);
00074
00075     Gnss_Sdr_Valve(size_t sizeof_stream_item,
00076                   uint64_t nitems,
00077                   Concurrent_Queue<pmt::pmt_t>* queue, bool stop_flowgraph);
00078
00079     uint64_t d_nitems;
00080     uint64_t d_ncopied_items;
00081     Concurrent_Queue<pmt::pmt_t>* d_queue;
00082     bool d_stop_flowgraph;
00083     bool d_open_valve;
00084 };
00085
00086
00087 /** \} */
00088 /** \} */
00089 #endif // GNSS_SDR_GNSS_SDR_VALVE_H

```

## 11.404 ion\_gsms\_chunk\_data.h File Reference

Holds logic for reading and decoding samples from a chunk.

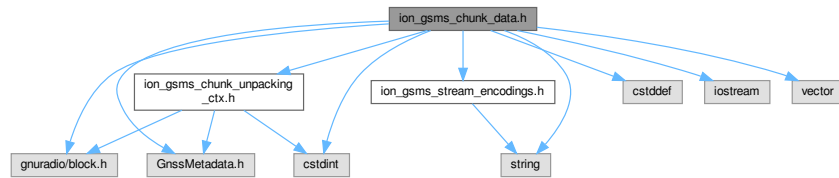
```

#include "ion_gsms_chunk_unpacking_ctx.h"
#include "ion_gsms_stream_encodings.h"
#include <gnuradio/block.h>
#include <GnssMetadata.h>
#include <cstdint>

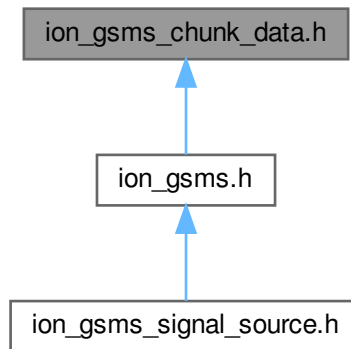
```

```
#include <cstdint>
#include <iostream>
#include <string>
#include <vector>
```

Include dependency graph for ion\_gsms\_chunk\_data.h:



This graph shows which files directly or indirectly include this file:



## Classes

- struct [Allocator](#)
- struct [Deleter](#)
- class [IONGSMSChunkData](#)

## Functions

- `std::size_t bits\_to\_item\_size(std::size_t bit_count)`
- `template<typename Callback>  
void with\_word\_type(uint8_t word_size, Callback callback)`

### 11.404.1 Detailed Description

Holds logic for reading and decoding samples from a chunk.

#### Author

Víctor Castillo Agüero, 2024. [victorcastilloaguero\(at\)gmail.com](mailto:victorcastilloaguero(at)gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [ion\\_gsms\\_chunk\\_data.h](#).

---

## 11.404.2 Function Documentation

### 11.404.2.1 bits\_to\_item\_size()

```
std::size_t bits_to_item_size (
    std::size_t bit_count) [inline]
```

Definition at line 31 of file [ion\\_gsms\\_chunk\\_data.h](#).

### 11.404.2.2 with\_word\_type()

```
template<typename Callback>
void with_word_type (
    uint8_t word_size,
    Callback callback)
```

Definition at line 90 of file [ion\\_gsms\\_chunk\\_data.h](#).

## 11.405 ion\_gsms\_chunk\_data.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file ion_gsms_chunk_data.h
00003  * \brief Holds logic for reading and decoding samples from a chunk
00004  * \author Victor Castillo Agüero, 2024. victorcastilloaguero(at)gmail.com
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_ION_GSMS_CHUNK_DATA_H
00018 #define GNSS_SDR_ION_GSMS_CHUNK_DATA_H
00019
00020 #include "ion_gsms_chunk_unpacking_ctx.h"
00021 #include "ion_gsms_stream_encodings.h"
00022 #include <gnuradio/block.h>
00023 #include <GnssMetadata.h>
00024 #include <cstdint>
00025 #include <cstdint>
00026 #include <iostream>
00027 #include <string>
00028 #include <vector>
00029
00030
00031 inline std::size_t bits_to_item_size(std::size_t bit_count)
00032 {
00033     if (bit_count <= 8)
00034     {
00035         return 1;
00036     }
00037     if (bit_count <= 16)
00038     {
00039         return 2;
00040     }
00041     if (bit_count <= 32)
00042     {
00043         return 4;
00044     }
00045     if (bit_count <= 64)
00046     {
00047         return 8;
00048     }
00049
00050     // You are asking too much of this humble processor
00051     std::cerr << "Item size too large (" << std::to_string(bit_count) << "), returning nonsense.\n";
00052     return 1;
00053 }
00054
00055
00056 // Define a functor that has a templated operator()
00057 struct Allocator
00058 {
00059     size_t countwords_;
```

```

00060     void*& buffer_; // Using void* to hold any type of pointer
00061
00062     Allocator(size_t countwords, void*& buffer)
00063         : countwords_(countwords), buffer_(buffer) {}
00064
00065     template <typename WordType>
00066     void operator() () const
00067     {
00068         buffer_ = new WordType[countwords_];
00069     }
00070 };
00071
00072
00073 // Define a functor to delete the allocated memory
00074 struct Deleter
00075 {
00076     void* buffer_;
00077
00078     explicit Deleter(void* buffer)
00079         : buffer_(buffer) {}
00080
00081     template <typename WordType>
00082     void operator() () const
00083     {
00084         delete[] static_cast<WordType*>(buffer_);
00085     }
00086 };
00087
00088
00089 template <typename Callback>
00090 void with_word_type(uint8_t word_size, Callback callback)
00091 {
00092     switch (word_size)
00093     {
00094     case 1:
00095         callback.template operator()<int8_t>();
00096         break;
00097     case 2:
00098         callback.template operator()<int16_t>();
00099         break;
00100     case 4:
00101         callback.template operator()<int32_t>();
00102         break;
00103     case 8:
00104         callback.template operator()<int64_t>();
00105         break;
00106     default:
00107         std::cerr << "Unknown word size (" << std::to_string(word_size) << ")", returning
nonsense.\n";
00108         break;
00109     }
00110 }
00111
00112 class IONGSMSChunkData
00113 {
00114 public:
00115     IONGSMSChunkData(const GnssMetadata::Chunk& chunk, const std::vector<std::string>& stream_ids,
std::size_t output_stream_offset);
00116
00117     ~IONGSMSChunkData();
00118
00119     IONGSMSChunkData(const IONGSMSChunkData& rhl) = delete;
00120     IONGSMSChunkData& operator=(const IONGSMSChunkData& rhl) = delete;
00121
00122     IONGSMSChunkData(IONGSMSChunkData&& rhl) = delete;
00123     IONGSMSChunkData& operator=(IONGSMSChunkData&& rhl) = delete;
00124
00125     std::size_t read_from_buffer(uint8_t* buffer, std::size_t offset);
00126
00127     void write_to_output(gr_vector_void_star& outputs, std::vector<int>& output_items);
00128
00129     std::size_t output_stream_count() const;
00130     std::size_t output_stream_item_size(std::size_t stream_index) const;
00131     std::size_t output_stream_item_rate(std::size_t stream_index) const;
00132 private:
00133     template <typename WT>
00134     void unpack_words(gr_vector_void_star& outputs, std::vector<int>& output_items);
00135
00136     template <typename WT>
00137     std::size_t write_stream_samples(
IONGSMSChunkUnpackingCtx<WT>& ctx,
const GnssMetadata::Lump& lump,
const GnssMetadata::IonStream& stream,
GnssMetadata::StreamEncoding stream_encoding,
void** out);
00143
00144

```

```

00145     template <typename WT, typename OT>
00146     void write_n_samples(
00147         IONGSMSChunkUnpackingCtx<WT>& ctx,
00148         GnssMetadata::Lump::LumpShift lump_shift,
00149         uint8_t sample_bitsize,
00150         std::size_t sample_count,
00151         GnssMetadata::StreamEncoding stream_encoding,
00152         OT** out);
00153
00154     template <typename Sample>
00155     static void decode_sample(uint8_t sample_bitsize, Sample* sample, GnssMetadata::StreamEncoding
encoding);
00156
00157     const GnssMetadata::Chunk& chunk_;
00158     uint8_t sizeword_;
00159     uint8_t countwords_;
00160     uint8_t padding_bitsize_;
00161     std::size_t output_stream_count_;
00162     std::vector<std::size_t> output_stream_item_size_;
00163     std::vector<std::size_t> output_stream_item_rate_;
00164
00165     struct stream_metadata_t
00166     {
00167         const GnssMetadata::Lump& lump_;
00168         const GnssMetadata::IonStream& stream_;
00169         GnssMetadata::StreamEncoding stream_encoding_;
00170         int output_index = -1;
00171
00172         stream_metadata_t(
00173             const GnssMetadata::Lump& lump_,
00174             const GnssMetadata::IonStream& stream_,
00175             GnssMetadata::StreamEncoding stream_encoding_,
00176             int output_index_ = -1) : lump(lump_),
stream(stream_),
stream_encoding(stream_encoding_),
output_index(output_index_)
00177         {
00178         }
00179     };
00180
00181     std::vector<stream_metadata_t> streams_;
00182
00183     void* buffer_;
00184 };
00185
00186 #endif // GNSS_SDR_ION_GSMS_CHUNK_DATA_H

```

## 11.406 ion\_gsms\_chunk\_unpacking\_ctx.h File Reference

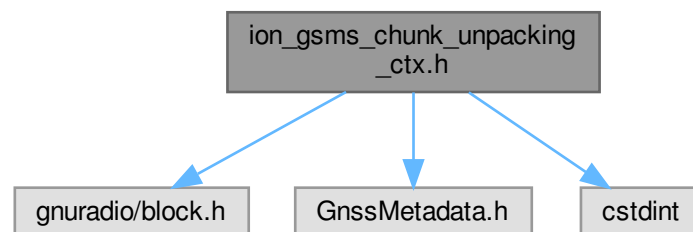
Holds state and provides utilities for unpacking samples from a chunk.

```
#include <gnuradio/block.h>
```

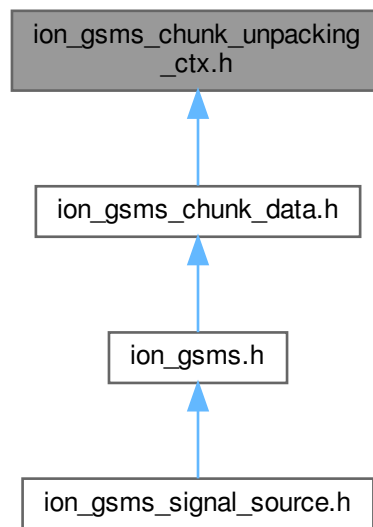
```
#include <GnssMetadata.h>
```

```
#include <cstdint>
```

Include dependency graph for ion\_gsms\_chunk\_unpacking\_ctx.h:



This graph shows which files directly or indirectly include this file:



## Classes

- struct [IONGSMSChunkUnpackingCtx](#)< WT >

### 11.406.1 Detailed Description

Holds state and provides utilities for unpacking samples from a chunk.

#### Author

Víctor Castillo Agüero, 2024. [victorcastilloaguero\(at\)gmail.com](mailto:victorcastilloaguero(at)gmail.com)

This is a template class, and thus, its member functions must be defined in the header file.

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [ion\\_gsms\\_chunk\\_unpacking\\_ctx.h](#).

## 11.407 ion\_gsms\_chunk\_unpacking\_ctx.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file ion_gsms_chunk_unpacking_ctx.h
00003  * \brief Holds state and provides utilities for unpacking samples from a chunk
00004  * \author Víctor Castillo Agüero, 2024. victorcastilloaguero(at)gmail.com
00005  *
00006  * This is a template class, and thus, its member functions must be defined in the header file.
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
  
```

```

00019 #ifndef GNSS_SDR_ION_GSMS_CHUNK_UNPACKING_CTX_H
00020 #define GNSS_SDR_ION_GSMS_CHUNK_UNPACKING_CTX_H
00021
00022 #include <gnuradio/block.h>
00023 #include <GnssMetadata.h>
00024 #include <stdint>
00025
00026 /** \addtogroup Signal_Source
00027 * \{ */
00028 /** \addtogroup Signal_Source_libs
00029 * \{ */
00030
00031 template <typename WT>
00032 struct IONGSMSChunkUnpackingCtx
00033 {
00034     static constexpr uint8_t word_bitsize_ = sizeof(WT) * 8;
00035
00036     const GnssMetadata::Chunk::WordShift word_shift_direction_;
00037     WT* iterator_ = nullptr; // Not owned by this class, MUST NOT destroy
00038     WT current_word_{};
00039     uint8_t bitshift_ = 0;
00040
00041     IONGSMSChunkUnpackingCtx(
00042         const GnssMetadata::Chunk::WordShift word_shift,
00043         WT* data_buffer,
00044         uint8_t data_buffer_word_count) : word_shift_direction_(word_shift)
00045     {
00046         if (word_shift_direction_ == GnssMetadata::Chunk::Left)
00047         {
00048             iterator_ = data_buffer;
00049         }
00050         else if (word_shift_direction_ == GnssMetadata::Chunk::Right)
00051         {
00052             iterator_ = &data_buffer[data_buffer_word_count];
00053         }
00054         if (iterator_)
00055         {
00056             advance_word(); // Initializes current_word_
00057         }
00058     }
00059
00060     void advance_word()
00061     {
00062         WT word = *iterator_;
00063         if (word_shift_direction_ == GnssMetadata::Chunk::Left)
00064         {
00065             ++iterator_;
00066         }
00067         else if (word_shift_direction_ == GnssMetadata::Chunk::Right)
00068         {
00069             --iterator_;
00070         }
00071
00072         current_word_ = word;
00073     }
00074
00075     void shift_current_word(uint8_t n)
00076     {
00077         if ((n % word_bitsize_) == 0)
00078         {
00079             for (uint8_t i = 0; i < (n / word_bitsize_); ++i)
00080             {
00081                 advance_word();
00082             }
00083             return;
00084         }
00085
00086         if (word_shift_direction_ == GnssMetadata::Chunk::Left)
00087         {
00088             current_word_ <= n;
00089         }
00090         else if (word_shift_direction_ == GnssMetadata::Chunk::Right)
00091         {
00092             current_word_ >= n;
00093         }
00094
00095         bitshift_ += n;
00096         if (bitshift_ >= word_bitsize_)
00097         {
00098             advance_word();
00099             bitshift_ -= word_bitsize_;
00100         }
00101     }
00102
00103     void shift_padding(uint8_t n_bits)
00104     {
00105         if (n_bits == 0)

```



```

00106         {
00107             return;
00108         }
00109
00110         if ((n_bits + (bitshift_ % word_bitsize_)) >= word_bitsize_)
00111         {
00112             const uint8_t bits_shifted = word_bitsize_ - (bitshift_ % word_bitsize_);
00113
00114             shift_current_word(bits_shifted);
00115             shift_padding(n_bits - bits_shifted);
00116         }
00117         else
00118         {
00119             shift_current_word(n_bits);
00120         }
00121     }
00122
00123     template <typename OT>
00124     void shift_sample(uint8_t sample_bitsize, OT* output, uint8_t output_bit_offset = 0)
00125     {
00126         if (sample_bitsize % word_bitsize_ == 0)
00127         {
00128             const uint8_t words_per_sample = sample_bitsize / word_bitsize_;
00129             for (uint8_t i = 0; i < words_per_sample; ++i)
00130             {
00131                 if (word_shift_direction_ == GnssMetadata::Chunk::Left)
00132                 {
00133                     *output |= (current_word_ << ((words_per_sample - 1 - i) *
word_bitsize_));
00134                 }
00135                 else if (word_shift_direction_ == GnssMetadata::Chunk::Right)
00136                 {
00137                     *output |= (current_word_ << (i * word_bitsize_));
00138                     // TODO - reverse bit order of sample? maybe?
00139                 }
00140                 advance_word();
00141             }
00142         }
00143         else if ((sample_bitsize + (bitshift_ % word_bitsize_)) > word_bitsize_)
00144         {
00145             const uint8_t bits_shifted = word_bitsize_ - (bitshift_ % word_bitsize_);
00146
00147             if (word_shift_direction_ == GnssMetadata::Chunk::Left)
00148             {
00149                 WT mask = ~((1 << (word_bitsize_ - bits_shifted)) - 1);
00150                 *output |= ((current_word_ & mask) >> output_bit_offset);
00151             }
00152             else if (word_shift_direction_ == GnssMetadata::Chunk::Right)
00153             {
00154                 WT mask = ((1 << (bits_shifted)) - 1);
00155                 *output |= (current_word_ & mask) << output_bit_offset;
00156                 // TODO - reverse bit order of sample? maybe?
00157             }
00158
00159             shift_current_word(bits_shifted);
00160             shift_sample(sample_bitsize - bits_shifted, output, bits_shifted);
00161         }
00162         else
00163         {
00164             if (word_shift_direction_ == GnssMetadata::Chunk::Left)
00165             {
00166                 WT mask = ~((1 << (word_bitsize_ - sample_bitsize)) - 1);
00167                 OT sample = (current_word_ & mask) >> (word_bitsize_ - sample_bitsize);
00168                 *output |= (sample) >> output_bit_offset;
00169             }
00170             else if (word_shift_direction_ == GnssMetadata::Chunk::Right)
00171             {
00172                 WT mask = ((1 << (sample_bitsize)) - 1);
00173                 *output |= (current_word_ & mask) << output_bit_offset;
00174                 // TODO - reverse bit order of sample? maybe?
00175             }
00176
00177             shift_current_word(sample_bitsize);
00178         }
00179     }
00180 };
00181
00182 /** \} */
00183 /** \} */
00184 #endif // GNSS_SDR_ION_GSMS_CHUNK_UNPACKING_CTX_H

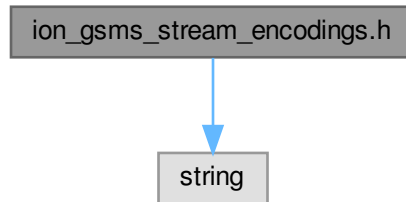
```

## 11.408 ion\_gsms\_stream\_encodings.h File Reference

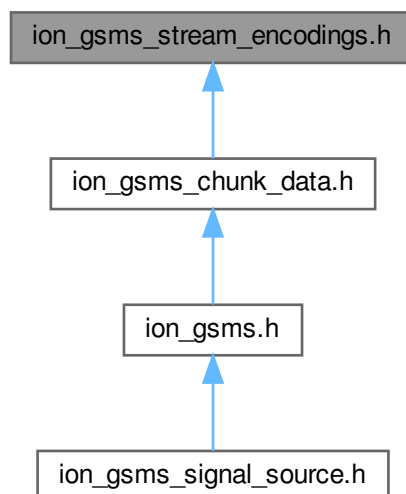
Implements look up tables for all encodings in the standard.

```
#include <string>
```

Include dependency graph for `ion_gsms_stream_encodings.h`:



This graph shows which files directly or indirectly include this file:



### Typedefs

- using [GnssMetadata::StreamEncoding](#) = unsigned char

### Functions

- StreamEncoding [GnssMetadata::encoding\\_from\\_string](#) (const std::string &str)

### Variables

- constexpr unsigned char [GnssMetadata::StreamEncodings::SIGN](#) = 0
- constexpr unsigned char [GnssMetadata::StreamEncodings::OB](#) = 1
- constexpr unsigned char [GnssMetadata::StreamEncodings::SM](#) = 2
- constexpr unsigned char [GnssMetadata::StreamEncodings::MS](#) = 3
- constexpr unsigned char [GnssMetadata::StreamEncodings::TC](#) = 4

- constexpr unsigned char [GnssMetadata::StreamEncodings::OG](#) = 5
- constexpr unsigned char [GnssMetadata::StreamEncodings::OBA](#) = 6
- constexpr unsigned char [GnssMetadata::StreamEncodings::SMA](#) = 7
- constexpr unsigned char [GnssMetadata::StreamEncodings::MSA](#) = 8
- constexpr unsigned char [GnssMetadata::StreamEncodings::TCA](#) = 9
- constexpr unsigned char [GnssMetadata::StreamEncodings::OGA](#) = 10
- constexpr unsigned char [GnssMetadata::StreamEncodings::FP](#) = 11
- template<typename T>  
T [GnssMetadata::two\\_bit\\_look\\_up](#) [11][4]
- template<typename T>  
T [GnssMetadata::three\\_bit\\_look\\_up](#) [11][8]
- template<typename T>  
T [GnssMetadata::four\\_bit\\_look\\_up](#) [11][16]
- template<typename T>  
T [GnssMetadata::five\\_bit\\_look\\_up](#) [11][32]

### 11.408.1 Detailed Description

Implements look up tables for all encodings in the standard.

#### Author

Víctor Castillo Agüero, 2024. victorcastilloaguero(at)gmail.com

These tables are taken from the standard's official document.

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [ion\\_gsms\\_stream\\_encodings.h](#).

### 11.408.2 Typedef Documentation

#### 11.408.2.1 StreamEncoding

using [GnssMetadata::StreamEncoding](#) = unsigned char

Definition at line 32 of file [ion\\_gsms\\_stream\\_encodings.h](#).

### 11.408.3 Function Documentation

#### 11.408.3.1 encoding\_from\_string()

[StreamEncoding](#) [GnssMetadata::encoding\\_from\\_string](#) (  
const std::string & *str*) [inline]

Definition at line 52 of file [ion\\_gsms\\_stream\\_encodings.h](#).

### 11.408.4 Variable Documentation

#### 11.408.4.1 five\_bit\_look\_up

template<typename T>

T [GnssMetadata::five\\_bit\\_look\\_up](#)[11][32] [inline]

##### Initial value:

```
{
    {},
    {-16, -15, -14, -13, -12, -11, -10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
     10, 11, 12, 13, 14, 15},
    {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 0, -1, -2, -3, -4, -5, -6, -7, -8, -9, -10, -11,
     -12, -13, -14, -15},
    {0, 0, 1, -1, 0, 0, 1, -1, 0, 0, 1, -1, 0, 0, 1, -1, 0, 0, 1, -1, 0, 0, 1, -1, 0, 0, 1, -1, 0, 0, 1,
     -1},
    {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, -16, -15, -14, -13, -12, -11, -10, -9, -8, -7,
     -6, -5, -4, -3, -2, -1},
    {-16, -15, -13, -14, -9, -10, -12, -11, -1, -2, -4, -3, -8, -7, -5, -6, 15, 14, 12, 13, 8, 9, 11, 10, 0,
     1, 3, 2, 7, 6, 4, 5},
}
```

```

{-31, -29, -27, -25, -23, -21, -19, -17, -15, -13, -11, -9, -7, -5, -3, -1, 1, 3, 5, 7, 9, 11, 13, 15,
 17, 19, 21, 23, 25, 27, 29, 31},
{1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, -1, -3, -5, -7, -9, -11, -13, -15, -17, -19,
-21, -23, -25, -27, -29, -31},
{1, -1, 3, -3, 5, -5, 7, -7, 9, -9, 11, -11, 13, -13, 15, -15, 17, -17, 19, -19, 21, -21, 23, -23, 25,
-25, 27, -27, 29, -29, 31, -31},
{1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, -31, -29, -27, -25, -23, -21, -19, -17, -15,
-13, -11, -9, -7, -5, -3, -1},
{-31, -29, -25, -27, -17, -19, -23, -21, -1, -3, -7, -5, -15, -13, -9, -11, 31, 29, 25, 27, 17, 19, 23,
 21, 1, 3, 7, 5, 15, 13, 9, 11},
}

```

Definition at line 151 of file [ion\\_gsms\\_stream\\_encodings.h](#).

#### 11.408.4.2 four\_bit\_look\_up

```

template<typename T>
T GnssMetadata::four_bit_look_up[11][16] [inline]

```

**Initial value:**

```

{
    {},
    {-8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7},
    {0, 1, 2, 3, 4, 5, 6, 7, 0, -1, -2, -3, -4, -5, -6, -7},
    {0, 0, 1, -1, 0, 0, 1, -1, 0, 0, 1, -1, 0, 0, 1, -1},
    {0, 1, 2, 3, 4, 5, 6, 7, -8, -7, -6, -5, -4, -3, -2, -1},
    {-8, -7, -5, -6, -1, -2, -4, -3, 7, 6, 4, 5, 0, 1, 3, 2},
    {-15, -13, -11, -9, -7, -5, -3, -1, 1, 3, 5, 7, 9, 11, 13, 15},
    {1, 3, 5, 7, 9, 11, 13, 15, -1, -3, -5, -7, -9, -11, -13, -15},
    {1, -1, 3, -3, 5, -5, 7, -7, 9, -9, 11, -11, 13, -13, 15, -15},
    {1, 3, 5, 7, 9, 11, 13, 15, -15, -13, -11, -9, -7, -5, -3, -1},
    {-15, -13, -9, -11, -1, -3, -7, -5, 15, 13, 9, 11, 1, 3, 7, 5},
}

```

Definition at line 136 of file [ion\\_gsms\\_stream\\_encodings.h](#).

#### 11.408.4.3 FP

```

unsigned char GnssMetadata::StreamEncodings::FP = 11 [constexpr]

```

Definition at line 48 of file [ion\\_gsms\\_stream\\_encodings.h](#).

#### 11.408.4.4 MS

```

unsigned char GnssMetadata::StreamEncodings::MS = 3 [constexpr]

```

Definition at line 40 of file [ion\\_gsms\\_stream\\_encodings.h](#).

#### 11.408.4.5 MSA

```

unsigned char GnssMetadata::StreamEncodings::MSA = 8 [constexpr]

```

Definition at line 45 of file [ion\\_gsms\\_stream\\_encodings.h](#).

#### 11.408.4.6 OB

```

unsigned char GnssMetadata::StreamEncodings::OB = 1 [constexpr]

```

Definition at line 38 of file [ion\\_gsms\\_stream\\_encodings.h](#).

#### 11.408.4.7 OBA

```

unsigned char GnssMetadata::StreamEncodings::OBA = 6 [constexpr]

```

Definition at line 43 of file [ion\\_gsms\\_stream\\_encodings.h](#).

#### 11.408.4.8 OG

```

unsigned char GnssMetadata::StreamEncodings::OG = 5 [constexpr]

```

Definition at line 42 of file [ion\\_gsms\\_stream\\_encodings.h](#).

#### 11.408.4.9 OGA

```

unsigned char GnssMetadata::StreamEncodings::OGA = 10 [constexpr]

```

Definition at line 47 of file [ion\\_gsms\\_stream\\_encodings.h](#).

**11.408.4.10 SIGN**

unsigned char GnssMetadata::StreamEncodings::SIGN = 0 [constexpr]  
 Definition at line 37 of file [ion\\_gsms\\_stream\\_encodings.h](#).

**11.408.4.11 SM**

unsigned char GnssMetadata::StreamEncodings::SM = 2 [constexpr]  
 Definition at line 39 of file [ion\\_gsms\\_stream\\_encodings.h](#).

**11.408.4.12 SMA**

unsigned char GnssMetadata::StreamEncodings::SMA = 7 [constexpr]  
 Definition at line 44 of file [ion\\_gsms\\_stream\\_encodings.h](#).

**11.408.4.13 TC**

unsigned char GnssMetadata::StreamEncodings::TC = 4 [constexpr]  
 Definition at line 41 of file [ion\\_gsms\\_stream\\_encodings.h](#).

**11.408.4.14 TCA**

unsigned char GnssMetadata::StreamEncodings::TCA = 9 [constexpr]  
 Definition at line 46 of file [ion\\_gsms\\_stream\\_encodings.h](#).

**11.408.4.15 three\_bit\_look\_up**

template<typename T>  
 T GnssMetadata::three\_bit\_look\_up[11][8] [inline]

**Initial value:**

```
{
    {},
    {-4, -3, -2, -1, 0, 1, 2, 3},
    {0, 1, 2, 3, 0, -1, -2, -3},
    {0, 0, 1, -1, 0, 0, 1, -1},
    {0, 1, 2, 3, -4, -3, -2, -1},
    {-4, -3, -1, -2, 3, 2, 0, 1},
    {-7, -5, -3, -1, 1, 3, 5, 7},
    {1, 3, 5, 7, -1, -3, -5, -7},
    {1, -1, 3, -3, 5, -5, 7, -7},
    {1, 3, 5, 7, -7, -5, -3, -1},
    {-7, -5, -1, -3, 7, 5, 1, 3},
}
```

Definition at line 121 of file [ion\\_gsms\\_stream\\_encodings.h](#).

**11.408.4.16 two\_bit\_look\_up**

template<typename T>  
 T GnssMetadata::two\_bit\_look\_up[11][4] [inline]

**Initial value:**

```
{
    {},
    {-2, -1, 0, 1},
    {0, 1, 0, -1},
    {0, 0, 1, -1},
    {0, 1, -2, -1},
    {-2, -1, 1, 0},
    {-3, -1, 1, 3},
    {1, 3, -1, -3},
    {1, -1, 3, -3},
    {1, 3, -3, -1},
    {-3, -1, 3, 1},
}
```

Definition at line 106 of file [ion\\_gsms\\_stream\\_encodings.h](#).

## 11.409 ion\_gsms\_stream\_encodings.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file ion_gsms_stream_encodings.h
00003  * \brief Implements look up tables for all encodings in the standard
00004  * \author Victor Castillo Agüero, 2024. victorcastilloaguero(at)gmail.com
00005  *
00006  * These tables are taken from the standard's official document.
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019 #ifndef GNSS_SDR_ION_GSMS_STREAM_ENCODINGS_H
00020 #define GNSS_SDR_ION_GSMS_STREAM_ENCODINGS_H
00021
00022 #include <string>
00023
00024 /** \addtogroup Signal_Source
00025  * \{ */
00026 /** \addtogroup Signal_Source_libs
00027  * \{ */
00028
00029 namespace GnssMetadata
00030 {
00031
00032 using StreamEncoding = unsigned char;
00033
00034 namespace StreamEncodings
00035 {
00036
00037 constexpr unsigned char SIGN = 0;
00038 constexpr unsigned char OB = 1;
00039 constexpr unsigned char SM = 2;
00040 constexpr unsigned char MS = 3;
00041 constexpr unsigned char TC = 4;
00042 constexpr unsigned char OG = 5;
00043 constexpr unsigned char OBA = 6;
00044 constexpr unsigned char SMA = 7;
00045 constexpr unsigned char MSA = 8;
00046 constexpr unsigned char TCA = 9;
00047 constexpr unsigned char OGA = 10;
00048 constexpr unsigned char FP = 11;
00049
00050 } // namespace StreamEncodings
00051
00052 inline StreamEncoding encoding_from_string(const std::string& str)
00053 {
00054     if (str == "SIGN")
00055     {
00056         return StreamEncodings::SIGN;
00057     }
00058     if (str == "OB")
00059     {
00060         return StreamEncodings::OB;
00061     }
00062     if (str == "SM")
00063     {
00064         return StreamEncodings::SM;
00065     }
00066     if (str == "MS")
00067     {
00068         return StreamEncodings::MS;
00069     }
00070     if (str == "TC")
00071     {
00072         return StreamEncodings::TC;
00073     }
00074     if (str == "OG")
00075     {
00076         return StreamEncodings::OG;
00077     }
00078     if (str == "OBA")
00079     {
00080         return StreamEncodings::OBA;
00081     }
00082     if (str == "SMA")
00083     {

```

```

00084         return StreamEncodings::SMA;
00085     }
00086     if (str == "MSA")
00087     {
00088         return StreamEncodings::MSA;
00089     }
00090     if (str == "TCA")
00091     {
00092         return StreamEncodings::TCA;
00093     }
00094     if (str == "OGA")
00095     {
00096         return StreamEncodings::OGA;
00097     }
00098     if (str == "FP")
00099     {
00100         return StreamEncodings::FP;
00101     }
00102     return 0;
00103 }
00104
00105 template <typename T>
00106 inline T two_bit_look_up[11][4]{
00107     {}, // [0]
00108     {-2, -1, 0, 1}, // [1 /*OB*/]
00109     {0, 1, 0, -1}, // [2 /*SM*/]
00110     {0, 0, 1, -1}, // [3 /*MS*/]
00111     {0, 1, -2, -1}, // [4 /*TC*/]
00112     {-2, -1, 1, 0}, // [5 /*OG*/]
00113     {-3, -1, 1, 3}, // [6 /*OBA*/]
00114     {1, 3, -1, -3}, // [7 /*SMA*/]
00115     {1, -1, 3, -3}, // [8 /*MSA*/]
00116     {1, 3, -3, -1}, // [9 /*TCA*/]
00117     {-3, -1, 3, 1}, // [10 /*OGA*/]
00118 };
00119
00120 template <typename T>
00121 inline T three_bit_look_up[11][8]{
00122     {}, // [0]
00123     {-4, -3, -2, -1, 0, 1, 2, 3}, // [1 /*OB*/]
00124     {0, 1, 2, 3, 0, -1, -2, -3}, // [2 /*SM*/]
00125     {0, 0, 1, -1, 0, 0, 1, -1}, // [3 /*MS*/]
00126     {0, 1, 2, 3, -4, -3, -2, -1}, // [4 /*TC*/]
00127     {-4, -3, -1, -2, 3, 2, 0, 1}, // [5 /*OG*/]
00128     {-7, -5, -3, -1, 1, 3, 5, 7}, // [6 /*OBA*/]
00129     {1, 3, 5, 7, -1, -3, -5, -7}, // [7 /*SMA*/]
00130     {1, -1, 3, -3, 5, -5, 7, -7}, // [8 /*MSA*/]
00131     {1, 3, 5, 7, -7, -5, -3, -1}, // [9 /*TCA*/]
00132     {-7, -5, -1, -3, 7, 5, 1, 3}, // [10 /*OGA*/]
00133 };
00134
00135 template <typename T>
00136 inline T four_bit_look_up[11][16]{
00137     {}, // [0]
00138     {-8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7}, // [1 /*OB*/]
00139     {0, 1, 2, 3, 4, 5, 6, 7, 0, -1, -2, -3, -4, -5, -6, -7}, // [2 /*SM*/]
00140     {0, 0, 1, -1, 0, 0, 1, -1, 0, 0, 1, -1, 0, 0, 1, -1}, // [3 /*MS*/]
00141     {0, 1, 2, 3, 4, 5, 6, 7, -8, -7, -6, -5, -4, -3, -2, -1}, // [4 /*TC*/]
00142     {-8, -7, -5, -6, -1, -2, -4, -3, 7, 6, 4, 5, 0, 1, 3, 2}, // [5 /*OG*/]
00143     {-15, -13, -11, -9, -7, -5, -3, -1, 1, 3, 5, 7, 9, 11, 13, 15}, // [6 /*OBA*/]
00144     {1, 3, 5, 7, 9, 11, 13, 15, -1, -3, -5, -7, -9, -11, -13, -15}, // [7 /*SMA*/]
00145     {1, -1, 3, -3, 5, -5, 7, -7, 9, -9, 11, -11, 13, -13, 15, -15}, // [8 /*MSA*/]
00146     {1, 3, 5, 7, 9, 11, 13, 15, -15, -13, -11, -9, -7, -5, -3, -1}, // [9 /*TCA*/]
00147     {-15, -13, -9, -11, -1, -3, -7, -5, 15, 13, 9, 11, 1, 3, 7, 5}, // [10 /*OGA*/]
00148 };
00149
00150 template <typename T>
00151 inline T five_bit_look_up[11][32]{
00152     {}, // [0]
00153     {-16, -15, -14, -13, -12, -11, -10, -9, -8, -7, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15}, // [1 /*OB*/]
00154     {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 0, -1, -2, -3, -4, -5, -6, -7, -8, -9, -10, -11, -12, -13, -14, -15}, // [2 /*SM*/]
00155     {0, 0, 1, -1, 0, 0, 1, -1, 0, 0, 1, -1, 0, 0, 1, -1, 0, 0, 1, -1, 0, 0, 1, -1, 0, 0, 1, -1, 0, 0, 1, -1}, // [3 /*MS*/]
00156     {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, -16, -15, -14, -13, -12, -11, -10, -9, -8, -7, -6, -5, -4, -3, -2, -1}, // [4 /*TC*/]
00157     {-16, -15, -13, -14, -9, -10, -12, -11, -1, -2, -4, -3, -8, -7, -5, -6, 15, 14, 12, 13, 8, 9, 11, 10, 0, 1, 3, 2, 7, 6, 4, 5}, // [5 /*OG*/]
00158     {-31, -29, -27, -25, -23, -21, -19, -17, -15, -13, -11, -9, -7, -5, -3, -1, 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31}, // [6 /*OBA*/]
00159     {1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, -1, -3, -5, -7, -9, -11, -13, -15, -17, -19, -21, -23, -25, -27, -29, -31}, // [7 /*SMA*/]
00160     {1, -1, 3, -3, 5, -5, 7, -7, 9, -9, 11, -11, 13, -13, 15, -15, 17, -17, 19, -19, 21, -21, 23, -23, 25, -25, 27, -27, 29, -29, 31, -31}, // [8 /*MSA*/]
00161     {1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, -31, -29, -27, -25, -23, -21, -19,

```

```

-17, -15, -13, -11, -9, -7, -5, -3, -1}, // [9 /*TCA*/]
00162 { -31, -29, -25, -27, -17, -19, -23, -21, -1, -3, -7, -5, -15, -13, -9, -11, 31, 29, 25, 27, 17,
19, 23, 21, 1, 3, 7, 5, 15, 13, 9, 11}, // [10 /*OGA*/]
00163 };
00164
00165 } // namespace GnssMetadata
00166
00167
00168 /** \} */
00169 /** \} */
00170 #endif // GNSS_SDR_ION_GSMS_STREAM_ENCODINGS_H

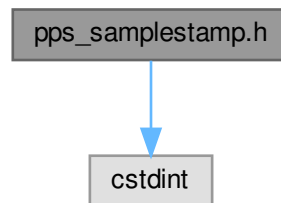
```

## 11.410 pps\_samplestamp.h File Reference

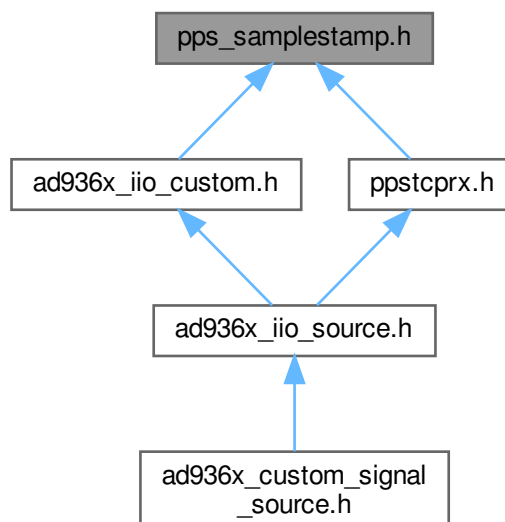
A simple container for the sample count associated to PPS rising edge.

```
#include <cstdint>
```

Include dependency graph for pps\_samplestamp.h:



This graph shows which files directly or indirectly include this file:





**Classes**

- class [PpsSamplestamp](#)

**11.410.1 Detailed Description**

A simple container for the sample count associated to PPS rising edge.

**11.410.1.1 autotoc\_md407****Author**

Javier Arribas, jarribas(at)cttc.es

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2023 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [pps\\_samplestamp.h](#).

**11.411 pps\_samplestamp.h**

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file pps_samplestamp.h
00003  * \brief A simple container for the sample count associated to PPS rising edge
00004  * \author Javier Arribas, jarribas(at)cttc.es
00005  * -----
00006  *
00007  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00008  * This file is part of GNSS-SDR.
00009  *
00010  * Copyright (C) 2010-2023 (see AUTHORS file for a list of contributors)
00011  * SPDX-License-Identifier: GPL-3.0-or-later
00012  * -----
00013  */
00014 */
00015
00016 #ifndef IIOPPS_PPS_SAMPLESTAMP_H
00017 #define IIOPPS_PPS_SAMPLESTAMP_H
00018
00019 #include <stdint>
00020
00021 class PpsSamplestamp
00022 {
00023 public:
00024     uint64_t samplestamp = 0; // PPS rising edge samples counter from the beginning of rx stream
00025     // operation. Notice that it is reset to zero if sample buffer overflow is detected on the FPGA side
00026     uint32_t overflow_reg = 0; // >0 indicates overflow situation in the FPGA RX buffer
00027 };
00028 #endif

```

**11.412 ppstcprx.h File Reference**

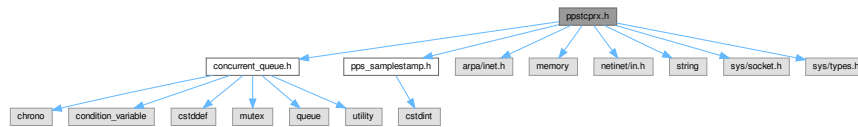
TCP client class for front-end PPS samplestamp information reception.

```

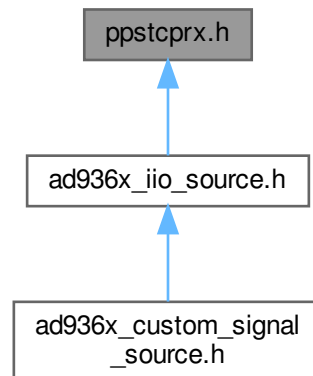
#include "concurrent_queue.h"
#include "pps_samplestamp.h"
#include <arpa/inet.h>
#include <memory>
#include <netinet/in.h>
#include <string>
#include <sys/socket.h>
#include <sys/types.h>

```

Include dependency graph for ppstcprx.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [pps\\_tcp\\_rx](#)

## 11.412.1 Detailed Description

TCP client class for front-end PPS samplestamp information reception.

### 11.412.1.1 autotoc\_md409

Author

Javier Arribas, jarribas(at)cttc.es

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2023 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [ppstcprx.h](#).

## 11.413 ppstcprx.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file ppstcprx.h
00003  * \brief TCP client class for front-end PPS samplestamp information reception
00004  * \author Javier Arribas, jarribas(at)cttc.es
00005  * -----
00006  *
00007  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00008  * This file is part of GNSS-SDR.

```

```

00009  *
00010  * Copyright (C) 2010-2023 (see AUTHORS file for a list of contributors)
00011  * SPDX-License-Identifier: GPL-3.0-or-later
00012  *
00013  * -----
00014  */
00015
00016 #ifndef SRC_LIBS_PPSTCPRX_H_
00017 #define SRC_LIBS_PPSTCPRX_H_
00018 #include "concurrent_queue.h"
00019 #include "pps_samplestamp.h"
00020 #include <arpa/inet.h>
00021 #include <memory>
00022 #include <netinet/in.h>
00023 #include <string>
00024 #include <sys/socket.h>
00025 #include <sys/types.h>
00026
00027 class pps_tcp_rx
00028 {
00029 private:
00030     std::shared_ptr<Concurrent_Queue<PpsSamplestamp>> Pps_queue;
00031     int clientSd{-1};
00032
00033 public:
00034     volatile bool is_connected{false};
00035     pps_tcp_rx() = default;
00036     virtual ~pps_tcp_rx() = default;
00037
00038     void receive_pps(const std::string& ip_address, int port);
00039     bool send_cmd(std::string cmd) const;
00040
00041     void set_pps_samplestamp_queue(std::shared_ptr<Concurrent_Queue<PpsSamplestamp>> queue);
00042 };
00043
00044 #endif /* SRC_LIBS_PPSTCPRX_H_ */

```

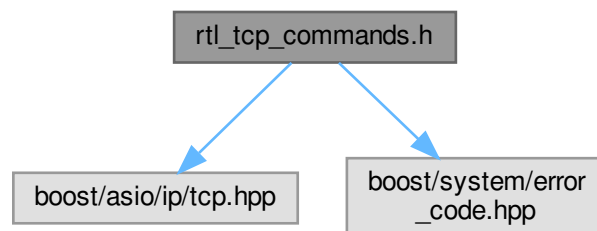
## 11.414 rtl\_tcp\_commands.h File Reference

Defines structures and constants for communicating with rtl\_tcp.

```
#include <boost/asio/ip/tcp.hpp>
```

```
#include <boost/system/error_code.hpp>
```

Include dependency graph for rtl\_tcp\_commands.h:



### Enumerations

- enum [RTL\\_TCP\\_COMMAND](#) {  
**RTL\_TCP\_SET\_FREQUENCY** = 1 , **RTL\_TCP\_SET\_SAMPLE\_RATE** = 2 , **RTL\_TCP\_SET\_GAIN\_MODE**  
= 3 , **RTL\_TCP\_SET\_GAIN** = 4 ,  
**RTL\_TCP\_SET\_IF\_GAIN** = 6 , **RTL\_TCP\_SET\_AGC\_MODE** = 8 }

*Command IDs for configuration rtl\_tcp.*

## Functions

- boost::system::error\_code [rtl\\_tcp\\_command](#) (RTL\_TCP\_COMMAND id, unsigned param, boost::asio::ip::tcp::socket &socket)

*Send a command to rtl\_tcp over the given socket.*

### 11.414.1 Detailed Description

Defines structures and constants for communicating with rtl\_tcp.

#### Author

Anthony Arnold, 2015. [anthony.arnold\(at\)uqconnect.edu.au](mailto:anthony.arnold(at)uqconnect.edu.au)

This file contains information taken from librtlsdr:

#### 11.414.1.1 <https://git.osmocom.org/rtl-sdr>

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [rtl\\_tcp\\_commands.h](#).

## 11.415 rtl\_tcp\_commands.h

[Go to the documentation of this file.](#)

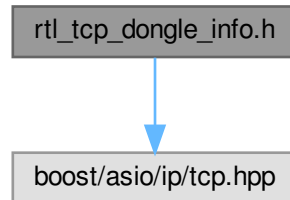
```
00001 /*!
00002  * \file rtl_tcp_commands.h
00003  * \brief Defines structures and constants for communicating with rtl_tcp
00004  * \author Anthony Arnold, 2015. anthony.arnold\(at\)uqconnect.edu.au
00005  *
00006  * This file contains information taken from librtlsdr:
00007  * https://git.osmocom.org/rtl-sdr
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019 #ifndef GNSS_SDR_RTL_TCP_COMMANDS_H
00020 #define GNSS_SDR_RTL_TCP_COMMANDS_H
00021
00022 #include <boost/asio/ip/tcp.hpp> // for tcp, tcp::socket
00023 #include <boost/system/error_code.hpp> // for error_code
00024
00025 /** \addtogroup Signal_Source
00026  * \{ */
00027 /** \addtogroup Signal_Source_libs
00028  * \{ */
00029
00030
00031 /// Command IDs for configuration rtl_tcp
00032 enum RTL_TCP_COMMAND
00033 {
00034     RTL_TCP_SET_FREQUENCY = 1,
00035     RTL_TCP_SET_SAMPLE_RATE = 2,
00036     RTL_TCP_SET_GAIN_MODE = 3,
00037     RTL_TCP_SET_GAIN = 4,
00038     RTL_TCP_SET_IF_GAIN = 6,
00039     RTL_TCP_SET_AGC_MODE = 8
00040 };
00041
00042
00043 /*!
00044  * \brief Send a command to rtl_tcp over the given socket.
00045  */
00046 boost::system::error_code rtl_tcp_command(RTL_TCP_COMMAND id, unsigned param,
00047     boost::asio::ip::tcp::socket &socket);
00048
00049
00050 /** \} */
00051 /** \} */
00052 #endif // GNSS_SDR_RTL_TCP_COMMANDS_H
```

## 11.416 rtl\_tcp\_dongle\_info.h File Reference

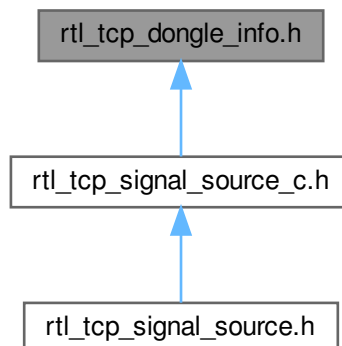
Interface for a structure sent by rtl\_tcp defining the hardware.

```
#include <boost/asio/ip/tcp.hpp>
```

Include dependency graph for rtl\_tcp\_dongle\_info.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [Rtl\\_Tcp\\_Dongle\\_Info](#)

*This class represents the dongle information which is sent by rtl\_tcp.*

### 11.416.1 Detailed Description

Interface for a structure sent by rtl\_tcp defining the hardware.

#### Author

Anthony Arnold, 2015. [anthony.arnold@uqconnect.edu.au](mailto:anthony.arnold@uqconnect.edu.au)

This file contains information taken from librtlsdr:

#### 11.416.1.1 <https://git.osmocom.org/rtl-sdr>

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [rtl\\_tcp\\_dongle\\_info.h](#).

## 11.417 rtl\_tcp\_dongle\_info.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file rtl_tcp_dongle_info.h
00003  * \brief Interface for a structure sent by rtl_tcp defining the hardware.
00004  * \author Anthony Arnold, 2015. anthony.arnold(at)uqconnect.edu.au
00005  *
00006  * This file contains information taken from librtlsdr:
00007  * https://git.osmocom.org/rtl-sdr
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019 #ifndef GNSS_SDR_RTL_TCP_DONGLE_INFO_H
00020 #define GNSS_SDR_RTL_TCP_DONGLE_INFO_H
00021
00022 #include <boost/asio/ip/tcp.hpp>
00023
00024
00025 /** \addtogroup Signal_Source
00026 * \{ */
00027 /** \addtogroup Signal_Source_libs
00028 * \{ */
00029
00030
00031 /*!
00032 * \brief This class represents the dongle information
00033 * which is sent by rtl_tcp.
00034 */
00035 class Rtl_Tcp_Dongle_Info
00036 {
00037 public:
00038     enum
00039     {
00040         TUNER_UNKNOWN = 0,
00041         TUNER_E4000,
00042         TUNER_FC0012,
00043         TUNER_FC0013,
00044         TUNER_FC2580,
00045         TUNER_R820T,
00046         TUNER_R828D
00047     };
00048
00049     Rtl_Tcp_Dongle_Info();
00050
00051     boost::system::error_code read(boost::asio::ip::tcp::socket &socket);
00052
00053     bool is_valid() const;
00054
00055     const char *get_type_name() const;
00056
00057     double clip_gain(int gain) const;
00058
00059     inline uint32_t get_tuner_type() const
00060     {
00061         return tuner_type_;
00062     }
00063
00064     inline uint32_t get_tuner_gain_count() const
00065     {
00066         return tuner_gain_count_;
00067     }
00068
00069 private:
00070     char magic_[4];
00071     uint32_t tuner_type_;
00072     uint32_t tuner_gain_count_;
00073 };
00074
00075
00076 /** \} */
00077 /** \} */

```

```
00078 #endif // GNSS_SDR_RTL_TCP_DONGLE_INFO_H
```

## 11.418 beidou\_b1i\_telemetry\_decoder.h File Reference

Interface of an adapter of a Beidou B1I NAV data decoder block to a [TelemetryDecoderInterface](#).

```
#include "beidou_b1i_telemetry_decoder_gs.h"
#include "gnss_satellite.h"
#include "gnss_synchro.h"
#include "telemetry_decoder_interface.h"
#include "tlm_conf.h"
#include <gnuradio/runtime_types.h>
#include <cstdint>
#include <string>
```

Include dependency graph for beidou\_b1i\_telemetry\_decoder.h:



### Classes

- class [BeidouB1iTelemetryDecoder](#)  
*This class implements a NAV data decoder for BEIDOU B1I.*

### 11.418.1 Detailed Description

Interface of an adapter of a Beidou B1I NAV data decoder block to a [TelemetryDecoderInterface](#).

#### Author

Damian Miralles, 2018. [dmiralles2009@gmail.com](mailto:dmiralles2009@gmail.com)

Sergi Segura, 2018. [sergi.segura.munoz\(at\)gmail.com](mailto:sergi.segura.munoz(at)gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [beidou\\_b1i\\_telemetry\\_decoder.h](#).

---

## 11.419 beidou\_b1i\_telemetry\_decoder.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file beidou_b1i_telemetry_decoder.h
00003  * \brief Interface of an adapter of a Beidou B1I NAV data decoder block
00004  * to a TelemetryDecoderInterface
00005  * \author Damian Miralles, 2018. dmiralles2009@gmail.com
00006  * \author Sergi Segura, 2018. sergi.segura.munoz(at)gmail.com
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019
00020 #ifndef GNSS_SDR_BEIDOU_B1I_TELEMETRY_DECODER_H
00021 #define GNSS_SDR_BEIDOU_B1I_TELEMETRY_DECODER_H
00022
```

```

00023 #include "beidou_bli_telemetry_decoder_gs.h"
00024 #include "gnss_satellite.h" // for Gnss_Satellite
00025 #include "gnss_synchro.h"
00026 #include "telemetry_decoder_interface.h"
00027 #include "tlm_conf.h"
00028 #include <gnuradio/runtime_types.h> // for basic_block_sptr, top_block_sptr
00029 #include <stddef> // for size_t
00030 #include <string>
00031
00032 /** \addtogroup Telemetry_Decoder
00033  * \{ */
00034 /** \addtogroup Telemetry_Decoder_adapters
00035  * \{ */
00036
00037
00038 class ConfigurationInterface;
00039
00040 /*!
00041  * \brief This class implements a NAV data decoder for BEIDOU B1I
00042  */
00043 class BeidouBliTelemetryDecoder : public TelemetryDecoderInterface
00044 {
00045 public:
00046     BeidouBliTelemetryDecoder(
00047         const ConfigurationInterface* configuration,
00048         const std::string& role,
00049         unsigned int in_streams,
00050         unsigned int out_streams);
00051
00052     void connect(gr::top_block_sptr top_block) override;
00053     void disconnect(gr::top_block_sptr top_block) override;
00054     gr::basic_block_sptr get_left_block() override;
00055     gr::basic_block_sptr get_right_block() override;
00056
00057     void set_satellite(const Gnss_Satellite& satellite) override;
00058
00059     inline std::string role() override
00060     {
00061         return role_;
00062     }
00063
00064     /*! Returns "BEIDOU_B1I_Telemetry_Decoder"
00065     inline std::string implementation() override
00066     {
00067         return "BEIDOU_B1I_Telemetry_Decoder";
00068     }
00069
00070     inline void set_channel(int channel) override { telemetry_decoder_>set_channel(channel); }
00071
00072     inline void reset() override
00073     {
00074         telemetry_decoder_>reset();
00075     }
00076
00077     inline size_t item_size() override
00078     {
00079         return sizeof(Gnss_Synchro);
00080     }
00081
00082 private:
00083     beidou_bli_telemetry_decoder_gs_sptr telemetry_decoder_;
00084     Gnss_Satellite satellite_;
00085     Tlm_Conf tlm_parameters_;
00086     std::string role_;
00087     unsigned int in_streams_;
00088     unsigned int out_streams_;
00089 };
00090
00091
00092 /** \} */
00093 /** \} */
00094 #endif

```

## 11.420 beidou\_b3i\_telemetry\_decoder.h File Reference

Interface of an adapter of a Beidou B3I NAV data decoder block to a [TelemetryDecoderInterface](#).

```

#include "beidou_b3i_telemetry_decoder_gs.h"
#include "gnss_satellite.h"
#include "gnss_synchro.h"
#include "telemetry_decoder_interface.h"
#include "tlm_conf.h"

```



```
#include <gnuradio/runtime_types.h>
#include <cstdint>
#include <string>
Include dependency graph for beidou_b3i_telemetry_decoder.h:
```



## Classes

- class [BeidouB3iTelemetryDecoder](#)

*This class implements a NAV data decoder for BEIDOU B1I.*

### 11.420.1 Detailed Description

Interface of an adapter of a Beidou B3I NAV data decoder block to a [TelemetryDecoderInterface](#).

#### Author

Damian Miralles, 2019. [dmiralles2009@gmail.com](mailto:dmiralles2009@gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [beidou\\_b3i\\_telemetry\\_decoder.h](#).

---

## 11.421 beidou\_b3i\_telemetry\_decoder.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file beidou_b3i_telemetry_decoder.h
00003  * \brief Interface of an adapter of a Beidou B3I NAV data decoder block
00004  * to a TelemetryDecoderInterface
00005  * \author Damian Miralles, 2019. dmiralles2009@gmail.com
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_BEIDOU_B3I_TELEMETRY_DECODER_H
00019 #define GNSS_SDR_BEIDOU_B3I_TELEMETRY_DECODER_H
00020
00021 #include "beidou_b3i_telemetry_decoder_gs.h"
00022 #include "gnss_satellite.h" // for Gnss_Satellite
00023 #include "gnss_synchro.h"
00024 #include "telemetry_decoder_interface.h"
00025 #include "tlm_conf.h"
00026 #include <gnuradio/runtime_types.h> // for basic_block_sptr, top_block_sptr
00027 #include <cstdint> // for size_t
00028 #include <string>
00029
00030
00031 /** \addtogroup Telemetry_Decoder
00032  * \{ */
00033 /** \addtogroup Telemetry_Decoder_adapters
00034  * \{ */
00035
00036
00037 class ConfigurationInterface;
00038
00039 /*!
00040  * \brief This class implements a NAV data decoder for BEIDOU B1I
00041  */
```

```

00042 class BeidouB3iTelemetryDecoder : public TelemetryDecoderInterface
00043 {
00044 public:
00045     BeidouB3iTelemetryDecoder(
00046         const ConfigurationInterface *configuration,
00047         const std::string &role, unsigned int in_streams,
00048         unsigned int out_streams);
00049
00050     void connect(gr::top_block_sptr top_block) override;
00051     void disconnect(gr::top_block_sptr top_block) override;
00052     gr::basic_block_sptr get_left_block() override;
00053     gr::basic_block_sptr get_right_block() override;
00054
00055     void set_satellite(const Gnss_Satellite &satellite) override;
00056
00057     inline std::string role() override { return role_; }
00058
00059     /*! Returns "BEIDOU_B3I_Telemetry_Decoder"
00060     inline std::string implementation() override
00061     {
00062         return "BEIDOU_B3I_Telemetry_Decoder";
00063     }
00064
00065     inline void set_channel(int channel) override
00066     {
00067         telemetry_decoder_>set_channel(channel);
00068     }
00069
00070     inline void reset() override
00071     {
00072         telemetry_decoder_>reset();
00073     }
00074
00075     inline size_t item_size() override { return sizeof(Gnss_Synchro); }
00076
00077 private:
00078     beidou_b3i_telemetry_decoder_gs_sptr telemetry_decoder_;
00079     Gnss_Satellite satellite_;
00080     Tlm_Conf tlm_parameters_;
00081     std::string role_;
00082     unsigned int in_streams_;
00083     unsigned int out_streams_;
00084 };
00085
00086 /** \} */
00087 /** \} */
00088 #endif

```

## 11.422 galileo\_e1b\_telemetry\_decoder.h File Reference

Interface of an adapter of a GALILEO E1B NAV data decoder block to a [TelemetryDecoderInterface](#).

```

#include "galileo_telemetry_decoder_gs.h"
#include "gnss_satellite.h"
#include "gnss_synchro.h"
#include "telemetry_decoder_interface.h"
#include "tlm_conf.h"
#include <gnuradio/runtime_types.h>
#include <cstdint>
#include <string>

```

Include dependency graph for galileo\_e1b\_telemetry\_decoder.h:



### Classes

- class [GalileoE1BTelemetryDecoder](#)

*This class implements a NAV data decoder for Galileo INAV frames in E1B radio link.*

### 11.422.1 Detailed Description

Interface of an adapter of a GALILEO E1B NAV data decoder block to a [TelemetryDecoderInterface](#).

#### Author

Javier Arribas 2013 jarribas(at)cttc.es, Mara Branzanti 2013. mara.branzanti(at)gmail.com

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [galileo\\_e1b\\_telemetry\\_decoder.h](#).

---

## 11.423 galileo\_e1b\_telemetry\_decoder.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file galileo_e1b_telemetry_decoder.h
00003  * \brief Interface of an adapter of a GALILEO E1B NAV data decoder block
00004  * to a TelemetryDecoderInterface
00005  * \author Javier Arribas 2013 jarribas(at)cttc.es,
00006  * Mara Branzanti 2013. mara.branzanti(at)gmail.com
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019
00020 #ifndef GNSS_SDR_GALILEO_E1B_TELEMETRY_DECODER_H
00021 #define GNSS_SDR_GALILEO_E1B_TELEMETRY_DECODER_H
00022
00023
00024 #include "galileo_telemetry_decoder_gs.h"
00025 #include "gnss_satellite.h"
00026 #include "gnss_synchro.h"
00027 #include "telemetry_decoder_interface.h"
00028 #include "tlm_conf.h"
00029 #include <gnuradio/runtime_types.h> // for basic_block_sptr, top_block_sptr
00030 #include <cstring> // for size_t
00031 #include <string>
00032
00033 /** \addtogroup Telemetry_Decoder
00034  * \{ */
00035 /** \addtogroup Telemetry_Decoder_adapters
00036  * \{ */
00037
00038
00039 class ConfigurationInterface;
00040
00041 /*!
00042  * \brief This class implements a NAV data decoder for Galileo INAV frames in E1B radio link
00043  */
00044 class GalileoE1BTelemetryDecoder : public TelemetryDecoderInterface
00045 {
00046 public:
00047     GalileoE1BTelemetryDecoder(
00048         const ConfigurationInterface* configuration,
00049         const std::string& role,
00050         unsigned int in_streams,
00051         unsigned int out_streams);
00052
00053     void connect(gr::top_block_sptr top_block) override;
00054     void disconnect(gr::top_block_sptr top_block) override;
00055     gr::basic_block_sptr get_left_block() override;
00056     gr::basic_block_sptr get_right_block() override;
00057
00058     void set_satellite(const Gnss_Satellite& satellite) override;
00059
00060     inline std::string role() override
00061     {
00062         return role_;
00063     }
00064
00065     /*!
00066     * \brief Returns "Galileo_E1B_Telemetry_Decoder"
00067     */

```

```

00068     inline std::string implementation() override
00069     {
00070         return "Galileo_E1B_Telemetry_Decoder";
00071     }
00072
00073     inline void set_channel(int channel) override { telemetry_decoder_>set_channel(channel); }
00074
00075     inline void reset() override
00076     {
00077         telemetry_decoder_>reset();
00078     }
00079
00080     inline size_t item_size() override
00081     {
00082         return sizeof(Gnss_Synchro);
00083     }
00084
00085 private:
00086     galileo_telemetry_decoder_gs_sptr telemetry_decoder_;
00087     Gnss_Satellite satellite_;
00088     Tlm_Conf tlm_parameters_;
00089     std::string role_;
00090     unsigned int in_streams_;
00091     unsigned int out_streams_;
00092 };
00093
00094
00095 /** \} */
00096 /** \} */
00097 #endif // GNSS_SDR_GALILEO_E1B_TELEMETRY_DECODER_H

```

## 11.424 galileo\_e5a\_telemetry\_decoder.h File Reference

Interface of an adapter of a GALILEO E5a FNAV data decoder block to a [TelemetryDecoderInterface](#).

```

#include "galileo_telemetry_decoder_gs.h"
#include "gnss_satellite.h"
#include "gnss_synchro.h"
#include "telemetry_decoder_interface.h"
#include "tlm_conf.h"
#include <gnuradio/runtime_types.h>
#include <cstdio>
#include <string>

```

Include dependency graph for galileo\_e5a\_telemetry\_decoder.h:



### Classes

- class [GalileoE5aTelemetryDecoder](#)

*This class implements a NAV data decoder for Galileo INAV frames in E1B radio link.*

### 11.424.1 Detailed Description

Interface of an adapter of a GALILEO E5a FNAV data decoder block to a [TelemetryDecoderInterface](#).

#### Author

Marc Sales, 2014. marcsales92(at)gmail.com \based on work from:

- Javier Arribas, 2011. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [galileo\\_e5a\\_telemetry\\_decoder.h](#).

---

## 11.425 galileo\_e5a\_telemetry\_decoder.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file galileo_e5a_telemetry_decoder.h
00003  * \brief Interface of an adapter of a GALILEO E5a FNAV data decoder block
00004  * to a TelemetryDecoderInterface
00005  * \author Marc Sales, 2014. marcsales92(at)gmail.com
00006  * \based on work from:
00007  *     <ul>
00008  *         <li> Javier Arribas, 2011. jarribas(at)cttc.es
00009  *     </ul>
00010  *
00011  *
00012  * -----
00013  *
00014  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00015  * This file is part of GNSS-SDR.
00016  *
00017  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00018  * SPDX-License-Identifier: GPL-3.0-or-later
00019  *
00020  * -----
00021  */
00022
00023 #ifndef GNSS_SDR_GALILEO_E5A_TELEMETRY_DECODER_H
00024 #define GNSS_SDR_GALILEO_E5A_TELEMETRY_DECODER_H
00025
00026 #include "galileo_telemetry_decoder_gs.h"
00027 #include "gnss_satellite.h" // for Gnss_Satellite
00028 #include "gnss_synchro.h"
00029 #include "telemetry_decoder_interface.h"
00030 #include "tlm_conf.h"
00031 #include <gnuradio/runtime_types.h> // for basic_block_sptr, top_block_sptr
00032 #include <cstring> // for size_t
00033 #include <string>
00034
00035 /** \addtogroup Telemetry_Decoder
00036  * \{ */
00037 /** \addtogroup Telemetry_Decoder_adapters
00038  * \{ */
00039
00040
00041 class ConfigurationInterface;
00042
00043 /*!
00044 * \brief This class implements a NAV data decoder for Galileo INAV frames in E1B radio link
00045 */
00046 class GalileoE5aTelemetryDecoder : public TelemetryDecoderInterface
00047 {
00048 public:
00049     GalileoE5aTelemetryDecoder(
00050         const ConfigurationInterface* configuration,
00051         const std::string& role,
00052         unsigned int in_streams,
00053         unsigned int out_streams);
00054
00055     void connect(gr::top_block_sptr top_block) override;
00056     void disconnect(gr::top_block_sptr top_block) override;
00057     gr::basic_block_sptr get_left_block() override;
00058     gr::basic_block_sptr get_right_block() override;
00059
00060     void set_satellite(const Gnss_Satellite& satellite) override;
00061
00062     inline std::string role() override
00063     {
00064         return role_;
00065     }
00066
00067     /*!
00068     * \brief Returns "Galileo_E5a_Telemetry_Decoder"
00069     */
00070     inline std::string implementation() override
00071     {
00072         return "Galileo_E5A_Telemetry_Decoder";
00073     }
00074
00075     inline void set_channel(int channel) override { telemetry_decoder_>set_channel(channel); }
00076
00077     inline void reset() override
00078     {
00079         telemetry_decoder_>reset();
00080     }
00081
00082     inline size_t item_size() override
00083     {

```

```

00084         return sizeof(Gnss_Synchro);
00085     }
00086
00087 private:
00088     galileo_telemetry_decoder_gs_sptr telemetry_decoder_;
00089     Gnss_Satellite satellite_;
00090     Tlm_Conf tlm_parameters_;
00091     std::string role_;
00092     unsigned int in_streams_;
00093     unsigned int out_streams_;
00094 };
00095
00096
00097 /** \} */
00098 /** \} */
00099 #endif // GNSS_SDR_GALILEO_E5A_TELEMETRY_DECODER_H

```

## 11.426 galileo\_e5b\_telemetry\_decoder.h File Reference

Interface of an adapter of a GALILEO E5B NAV data decoder block to a [TelemetryDecoderInterface](#).

```

#include "galileo_telemetry_decoder_gs.h"
#include "gnss_satellite.h"
#include "gnss_synchro.h"
#include "telemetry_decoder_interface.h"
#include "tlm_conf.h"
#include <gnuradio/runtime_types.h>
#include <cstdint>
#include <string>

```

Include dependency graph for galileo\_e5b\_telemetry\_decoder.h:



### Classes

- class [GalileoE5bTelemetryDecoder](#)

*This class implements a NAV data decoder for Galileo INAV frames in E5b radio link.*

### 11.426.1 Detailed Description

Interface of an adapter of a GALILEO E5B NAV data decoder block to a [TelemetryDecoderInterface](#).

#### Author

Piyush Gupta 2020 [piyush04111999@gmail.com](mailto:piyush04111999@gmail.com).

#### Note

Code added as part of GSoC 2020 Program.

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [galileo\\_e5b\\_telemetry\\_decoder.h](#).

---

## 11.427 galileo\_e5b\_telemetry\_decoder.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file galileo_e5b_telemetry_decoder.h
00003  * \brief Interface of an adapter of a GALILEO E5B NAV data decoder block
00004  * to a TelemetryDecoderInterface
00005  * \author Piyush Gupta 2020 piyush04111999@gmail.com.

```

```

00006  * \note Code added as part of GSoC 2020 Program.
00007  *
00008  *
00009  * -----
00010  *
00011  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00012  * This file is part of GNSS-SDR.
00013  *
00014  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00015  * SPDX-License-Identifier: GPL-3.0-or-later
00016  *
00017  * -----
00018  */
00019
00020
00021 #ifndef GNSS_SDR_GALILEO_E5B_TELEMETRY_DECODER_H
00022 #define GNSS_SDR_GALILEO_E5B_TELEMETRY_DECODER_H
00023
00024 #include "galileo_telemetry_decoder_gs.h"
00025 #include "gnss_satellite.h"
00026 #include "gnss_synchro.h"
00027 #include "telemetry_decoder_interface.h"
00028 #include "tlm_conf.h"
00029 #include <gnuradio/runtime_types.h> // for basic_block_sptr, top_block_sptr
00030 #include <stddef> // for size_t
00031 #include <string>
00032
00033 /** \addtogroup Telemetry_Decoder
00034  * \{ */
00035 /** \addtogroup Telemetry_Decoder_adapters
00036  * \{ */
00037
00038
00039 class ConfigurationInterface;
00040
00041 /*!
00042  * \brief This class implements a NAV data decoder for Galileo INAV frames in E5b radio link
00043  */
00044 class GalileoE5bTelemetryDecoder : public TelemetryDecoderInterface
00045 {
00046 public:
00047     GalileoE5bTelemetryDecoder(
00048         const ConfigurationInterface* configuration,
00049         const std::string& role,
00050         unsigned int in_streams,
00051         unsigned int out_streams);
00052
00053     /*!
00054      * \brief Returns "Galileo_E5b_Telemetry_Decoder"
00055      */
00056     inline std::string implementation() override
00057     {
00058         return "Galileo_E5b_Telemetry_Decoder";
00059     }
00060
00061     /*!
00062      * \brief Connect
00063      */
00064     void connect(gr::top_block_sptr top_block) override;
00065
00066     /*!
00067      * \brief Disconnect
00068      */
00069     void disconnect(gr::top_block_sptr top_block) override;
00070
00071     /*!
00072      * \brief Get left block
00073      */
00074     gr::basic_block_sptr get_left_block() override;
00075
00076     /*!
00077      * \brief Get right block
00078      */
00079     gr::basic_block_sptr get_right_block() override;
00080
00081     void set_satellite(const Gnss_Satellite& satellite) override;
00082
00083     inline std::string role() override
00084     {
00085         return role_;
00086     }
00087
00088     inline void set_channel(int channel) override { telemetry_decoder_>set_channel(channel); }
00089
00090     inline void reset() override
00091     {
00092         telemetry_decoder_>reset();

```

```

00093     }
00094
00095     inline size_t item_size() override
00096     {
00097         return sizeof(Gnss_Synchro);
00098     }
00099
00100 private:
00101     galileo_telemetry_decoder_gs_sptr telemetry_decoder_;
00102     Gnss_Satellite satellite_;
00103     Tlm_Conf tlm_parameters_;
00104     std::string role_;
00105     unsigned int in_streams_;
00106     unsigned int out_streams_;
00107 };
00108
00109
00110 /** \} */
00111 /** \} */
00112 #endif // GNSS_SDR_GALILEO_E5B_TELEMETRY_DECODER_H

```

## 11.428 galileo\_e6\_telemetry\_decoder.h File Reference

Interface of an adapter of a GALILEO E6 CNAV data decoder block to a [TelemetryDecoderInterface](#).

```

#include "galileo_telemetry_decoder_gs.h"
#include "gnss_satellite.h"
#include "gnss_synchro.h"
#include "telemetry_decoder_interface.h"
#include "tlm_conf.h"
#include <gnuradio/runtime_types.h>
#include <cstddef>
#include <string>

```

Include dependency graph for galileo\_e6\_telemetry\_decoder.h:



### Classes

- class [GalileoE6TelemetryDecoder](#)

*This class implements a NAV data decoder for Galileo CNAV frames in E6 radio link.*

### 11.428.1 Detailed Description

Interface of an adapter of a GALILEO E6 CNAV data decoder block to a [TelemetryDecoderInterface](#).

#### Author

Carles Fernandez-Prades, 2020 [cfernandez@cttc.es](mailto:cfernandez@cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [galileo\\_e6\\_telemetry\\_decoder.h](#).

## 11.429 galileo\_e6\_telemetry\_decoder.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file galileo_e6_telemetry_decoder.h
00003  * \brief Interface of an adapter of a GALILEO E6 CNAV data decoder block
00004  * to a TelemetryDecoderInterface
00005  * \author Carles Fernandez-Prades, 2020 cfernandez@cttc.es
00006  *
00007  *

```



```

00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019
00020 #ifndef GNSS_SDR_GALILEO_E6_TELEMETRY_DECODER_H
00021 #define GNSS_SDR_GALILEO_E6_TELEMETRY_DECODER_H
00022
00023 #include "galileo_telemetry_decoder_gs.h"
00024 #include "gnss_satellite.h"
00025 #include "gnss_synchro.h"
00026 #include "telemetry_decoder_interface.h"
00027 #include "tlm_conf.h"
00028 #include <gnuradio/runtime_types.h> // for basic_block_sptr, top_block_sptr
00029 #include <cstdlib> // for size_t
00030 #include <string>
00031
00032 /** \addtogroup Telemetry_Decoder
00033  * \{ */
00034 /** \addtogroup Telemetry_Decoder_adapters
00035  * \{ */
00036
00037
00038 class ConfigurationInterface;
00039
00040 /*!
00041  * \brief This class implements a NAV data decoder for Galileo CNAV frames in E6 radio link
00042  */
00043 class GalileoE6TelemetryDecoder : public TelemetryDecoderInterface
00044 {
00045 public:
00046     GalileoE6TelemetryDecoder(
00047         const ConfigurationInterface* configuration,
00048         const std::string& role,
00049         unsigned int in_streams,
00050         unsigned int out_streams);
00051
00052     /*!
00053      * \brief Returns "Galileo_E6_Telemetry_Decoder"
00054      */
00055     inline std::string implementation() override
00056     {
00057         return "Galileo_E6_Telemetry_Decoder";
00058     }
00059
00060     /*!
00061      * \brief Connect
00062      */
00063     void connect(gr::top_block_sptr top_block) override;
00064
00065     /*!
00066      * \brief Disconnect
00067      */
00068     void disconnect(gr::top_block_sptr top_block) override;
00069
00070     /*!
00071      * \brief Get left block
00072      */
00073     gr::basic_block_sptr get_left_block() override;
00074
00075     /*!
00076      * \brief Get right block
00077      */
00078     gr::basic_block_sptr get_right_block() override;
00079
00080     void set_satellite(const Gnss_Satellite& satellite) override;
00081
00082     inline std::string role() override
00083     {
00084         return role_;
00085     }
00086
00087     inline void set_channel(int channel) override { telemetry_decoder_>set_channel(channel); }
00088
00089     inline void reset() override
00090     {
00091         telemetry_decoder_>reset();
00092     }
00093
00094     inline size_t item_size() override

```

```

00095     {
00096         return sizeof(Gnss_Synchro);
00097     }
00098
00099 private:
00100     galileo_telemetry_decoder_gs_sptr telemetry_decoder_;
00101     Gnss_Satellite satellite_;
00102     Tlm_Conf tlm_parameters_;
00103     std::string role_;
00104     unsigned int in_streams_;
00105     unsigned int out_streams_;
00106 };
00107
00108
00109 /** \} */
00110 /** \} */
00111 #endif // GNSS_SDR_GALILEO_E6_TELEMETRY_DECODER_H

```

## 11.430 glonass\_l1\_ca\_telemetry\_decoder.h File Reference

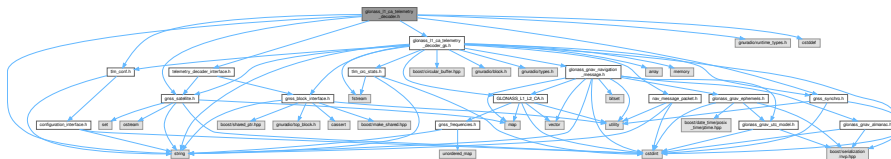
Interface of an adapter of a GLONASS L1 C/A NAV data decoder block to a [TelemetryDecoderInterface](#).

```

#include "glonass_l1_ca_telemetry_decoder_gs.h"
#include "gnss_satellite.h"
#include "gnss_synchro.h"
#include "telemetry_decoder_interface.h"
#include "tlm_conf.h"
#include <gnuradio/runtime_types.h>
#include <cstdint>
#include <string>

```

Include dependency graph for glonass\_l1\_ca\_telemetry\_decoder.h:



### Classes

- class [GlonassL1CaTelemetryDecoder](#)

*This class implements a NAV data decoder for GLONASS L1 C/A.*

### 11.430.1 Detailed Description

Interface of an adapter of a GLONASS L1 C/A NAV data decoder block to a [TelemetryDecoderInterface](#).

#### Note

Code added as part of GSoC 2017 program

#### Author

Damian Miralles, 2017. [dmiralles2009\(at\)gmail.com](mailto:dmiralles2009(at)gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [glonass\\_l1\\_ca\\_telemetry\\_decoder.h](#).

---

## 11.431 glonass\_l1\_ca\_telemetry\_decoder.h

[Go to the documentation of this file.](#)

```
00001 /*!
```

```

00002  * \file glonass_l1_ca_telemetry_decoder.h
00003  * \brief Interface of an adapter of a GLONASS L1 C/A NAV data decoder block
00004  * to a TelemetryDecoderInterface
00005  * \note Code added as part of GSOC 2017 program
00006  * \author Damian Miralles, 2017. dmiralles2009(at)gmail.com
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019
00020 #ifndef GNSS_SDR_GLOMSS_L1_CA_TELEMETRY_DECODER_H
00021 #define GNSS_SDR_GLOMSS_L1_CA_TELEMETRY_DECODER_H
00022
00023 #include "glonass_l1_ca_telemetry_decoder_gs.h"
00024 #include "gnss_satellite.h" // for Gnss_Satellite
00025 #include "gnss_synchro.h"
00026 #include "telemetry_decoder_interface.h"
00027 #include "tlm_conf.h"
00028 #include <gnuradio/runtime_types.h> // for basic_block_sptr, top_block_sptr
00029 #include <stddef> // for size_t
00030 #include <string>
00031
00032 /** \addtogroup Telemetry_Decoder
00033  * \{ */
00034 /** \addtogroup Telemetry_Decoder_adapters
00035  * \{ */
00036
00037
00038 class ConfigurationInterface;
00039
00040 /*!
00041  * \brief This class implements a NAV data decoder for GLONASS L1 C/A
00042  */
00043 class GlonassL1CaTelemetryDecoder : public TelemetryDecoderInterface
00044 {
00045 public:
00046     GlonassL1CaTelemetryDecoder(
00047         const ConfigurationInterface* configuration,
00048         const std::string& role,
00049         unsigned int in_streams,
00050         unsigned int out_streams);
00051
00052     void connect(gr::top_block_sptr top_block) override;
00053     void disconnect(gr::top_block_sptr top_block) override;
00054     gr::basic_block_sptr get_left_block() override;
00055     gr::basic_block_sptr get_right_block() override;
00056     void set_satellite(const Gnss_Satellite& satellite) override;
00057
00058     inline void set_channel(int channel) override { telemetry_decoder_>set_channel(channel); }
00059
00060     inline std::string role() override
00061     {
00062         return role_;
00063     }
00064
00065     /*! Returns "GLONASS_L1_CA_Telemetry_Decoder"
00066     inline std::string implementation() override
00067     {
00068         return "GLONASS_L1_CA_Telemetry_Decoder";
00069     }
00070
00071     inline void reset() override
00072     {
00073         telemetry_decoder_>reset();
00074     }
00075
00076     inline size_t item_size() override
00077     {
00078         return sizeof(Gnss_Synchro);
00079     }
00080
00081 private:
00082     glonass_l1_ca_telemetry_decoder_gs_sptr telemetry_decoder_;
00083     Gnss_Satellite satellite_;
00084     Tlm_Conf tlm_parameters_;
00085     std::string role_;
00086     unsigned int in_streams_;
00087     unsigned int out_streams_;
00088 };

```



```

00020 #define GNSS_SDR_GLONASS_L2_CA_TELEMETRY_DECODER_H
00021
00022 #include "glonass_l2_ca_telemetry_decoder_gs.h"
00023 #include "gnss_satellite.h" // for Gnss_Satellite
00024 #include "gnss_synchro.h"
00025 #include "telemetry_decoder_interface.h"
00026 #include "tlm_conf.h"
00027 #include <gnuradio/runtime_types.h> // for basic_block_sptr, top_block_sptr
00028 #include <cstdint> // for size_t
00029 #include <string>
00030
00031 /** \addtogroup Telemetry_Decoder
00032 * \{ */
00033 /** \addtogroup Telemetry_Decoder_adapters
00034 * \{ */
00035
00036
00037 class ConfigurationInterface;
00038
00039 /*!
00040 * \brief This class implements a NAV data decoder for GLONASS L2 C/A
00041 */
00042 class GlonassL2CaTelemetryDecoder : public TelemetryDecoderInterface
00043 {
00044 public:
00045     GlonassL2CaTelemetryDecoder(
00046         const ConfigurationInterface* configuration,
00047         const std::string& role,
00048         unsigned int in_streams,
00049         unsigned int out_streams);
00050
00051     void connect(gr::top_block_sptr top_block) override;
00052     void disconnect(gr::top_block_sptr top_block) override;
00053     gr::basic_block_sptr get_left_block() override;
00054     gr::basic_block_sptr get_right_block() override;
00055     void set_satellite(const Gnss_Satellite& satellite) override;
00056
00057     inline void set_channel(int channel) override { telemetry_decoder_>set_channel(channel); }
00058
00059     inline std::string role() override
00060     {
00061         return role_;
00062     }
00063
00064     /*! Returns "GLONASS_L2_CA_Telemetry_Decoder"
00065     inline std::string implementation() override
00066     {
00067         return "GLONASS_L2_CA_Telemetry_Decoder";
00068     }
00069
00070     inline void reset() override
00071     {
00072         telemetry_decoder_>reset();
00073     }
00074
00075     inline size_t item_size() override
00076     {
00077         return sizeof(Gnss_Synchro);
00078     }
00079
00080 private:
00081     glonass_l2_ca_telemetry_decoder_gs_sptr telemetry_decoder_;
00082     Gnss_Satellite satellite_;
00083     Tlm_Conf tlm_parameters_;
00084     std::string role_;
00085     unsigned int in_streams_;
00086     unsigned int out_streams_;
00087 };
00088
00089
00090 /** \} */
00091 /** \} */
00092 #endif // GNSS_SDR_GLONASS_L2_CA_TELEMETRY_DECODER_H

```

## 11.434 gps\_l1\_ca\_telemetry\_decoder.h File Reference

Interface of an adapter of a GPS L1 C/A NAV data decoder block to a [TelemetryDecoderInterface](#).

```

#include "gnss_satellite.h"
#include "gnss_synchro.h"
#include "gps_l1_ca_telemetry_decoder_gs.h"
#include "telemetry_decoder_interface.h"
#include "tlm_conf.h"

```

```
#include <gnuradio/runtime_types.h>
#include <cstddef>
#include <string>
Include dependency graph for gps_l1_ca_telemetry_decoder.h:
```



## Classes

- class [GpsL1CaTelemetryDecoder](#)

*This class implements a NAV data decoder for GPS L1 C/A.*

### 11.434.1 Detailed Description

Interface of an adapter of a GPS L1 C/A NAV data decoder block to a [TelemetryDecoderInterface](#).

#### Author

Javier Arribas, 2011. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gps\\_l1\\_ca\\_telemetry\\_decoder.h](#).

---

## 11.435 gps\_l1\_ca\_telemetry\_decoder.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file gps_l1_ca_telemetry_decoder.h
00003  * \brief Interface of an adapter of a GPS L1 C/A NAV data decoder block
00004  * to a TelemetryDecoderInterface
00005  * \author Javier Arribas, 2011. jarribas(at)cttc.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018
00019 #ifndef GNSS_SDR_GPS_L1_CA_TELEMETRY_DECODER_H
00020 #define GNSS_SDR_GPS_L1_CA_TELEMETRY_DECODER_H
00021
00022 #include "gnss_satellite.h" // for Gnss_Satellite
00023 #include "gnss_synchro.h"
00024 #include "gps_l1_ca_telemetry_decoder_gs.h"
00025 #include "telemetry_decoder_interface.h"
00026 #include "tlm_conf.h"
00027 #include <gnuradio/runtime_types.h> // for basic_block_sptr, top_block_sptr
00028 #include <cstddef> // for size_t
00029 #include <string>
00030
00031 /** \addtogroup Telemetry_Decoder Telemetry Decoder
00032  * Classes for the decoding of GNSS Navigation messages.
00033  * \{ */
00034 /** \addtogroup Telemetry_Decoder_adapters telemetry_decoder_adapters
00035  * Wrap GNU Radio blocks for the decoding of GNSS Navigation messages with a
00036  * TelemetryDecoderInterface
00037  * \{ */
00038
00039
00040 class ConfigurationInterface;
00041
```

```

00042  /*!
00043   * \brief This class implements a NAV data decoder for GPS L1 C/A
00044   */
00045  class GpsL1CaTelemetryDecoder : public TelemetryDecoderInterface
00046  {
00047  public:
00048      GpsL1CaTelemetryDecoder(
00049          const ConfigurationInterface* configuration,
00050          const std::string& role,
00051          unsigned int in_streams,
00052          unsigned int out_streams);
00053
00054      void connect(gr::top_block_sptr top_block) override;
00055      void disconnect(gr::top_block_sptr top_block) override;
00056      gr::basic_block_sptr get_left_block() override;
00057      gr::basic_block_sptr get_right_block() override;
00058
00059      void set_satellite(const Gnss_Satellite& satellite) override;
00060
00061      inline std::string role() override
00062      {
00063          return role_;
00064      }
00065
00066      ///! Returns "GPS_L1_CA_Telemetry_Decoder"
00067      inline std::string implementation() override
00068      {
00069          return "GPS_L1_CA_Telemetry_Decoder";
00070      }
00071
00072      inline void set_channel(int channel) override { telemetry_decoder_>set_channel(channel); }
00073
00074      inline void reset() override
00075      {
00076          telemetry_decoder_>reset();
00077      }
00078
00079      inline size_t item_size() override
00080      {
00081          return sizeof(Gnss_Synchro);
00082      }
00083
00084  private:
00085      gps_l1_ca_telemetry_decoder_gs_sptr telemetry_decoder_;
00086      Gnss_Satellite satellite_;
00087      Tlm_Conf tlm_parameters_;
00088      std::string role_;
00089      unsigned int in_streams_;
00090      unsigned int out_streams_;
00091  };
00092
00093  /** \} */
00094  /** \} */
00095  /** \} */
00096  #endif // GNSS_SDR_GPS_L1_CA_TELEMETRY_DECODER_H

```

## 11.436 gps\_l2c\_telemetry\_decoder.h File Reference

Interface of an adapter of a GPS L2C (CNAV) data decoder block to a [TelemetryDecoderInterface](#).

```

#include "gnss_satellite.h"
#include "gnss_synchro.h"
#include "gps_l2c_telemetry_decoder_gs.h"
#include "telemetry_decoder_interface.h"
#include "tlm_conf.h"
#include <gnuradio/runtime_types.h>
#include <cstdint>
#include <string>

```

Include dependency graph for gps\_l2c\_telemetry\_decoder.h:



## Classes

- class [GpsL2CTelemetryDecoder](#)

*This class implements a NAV data decoder for GPS L2 M.*

### 11.436.1 Detailed Description

Interface of an adapter of a GPS L2C (CNAV) data decoder block to a [TelemetryDecoderInterface](#).

#### Author

Javier Arribas, 2015. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [gps\\_l2c\\_telemetry\\_decoder.h](#).

---

### 11.437 `gps_l2c_telemetry_decoder.h`

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file gps_l2c_telemetry_decoder.h
00003  * \brief Interface of an adapter of a GPS L2C (CNAV) data decoder block
00004  * to a TelemetryDecoderInterface
00005  * \author Javier Arribas, 2015. jarribas(at)cttc.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018
00019 #ifndef GNSS_SDR_GPS_L2C_TELEMETRY_DECODER_H
00020 #define GNSS_SDR_GPS_L2C_TELEMETRY_DECODER_H
00021
00022 #include "gnss_satellite.h" // for Gnss_Satellite
00023 #include "gnss_synchro.h"
00024 #include "gps_l2c_telemetry_decoder_gs.h"
00025 #include "telemetry_decoder_interface.h"
00026 #include "tlm_conf.h"
00027 #include <gnuradio/runtime_types.h> // for basic_block_sptr, top_block_sptr
00028 #include <cstdlib> // for size_t
00029 #include <string>
00030
00031
00032 /** \addtogroup Telemetry_Decoder
00033  * \{ */
00034 /** \addtogroup Telemetry_Decoder_adapters
00035  * \{ */
00036
00037
00038 class ConfigurationInterface;
00039
00040 /*!
00041 * \brief This class implements a NAV data decoder for GPS L2 M
00042 */
00043 class GpsL2CTelemetryDecoder : public TelemetryDecoderInterface
00044 {
00045 public:
00046     GpsL2CTelemetryDecoder(
00047         const ConfigurationInterface* configuration,
00048         const std::string& role,
00049         unsigned int in_streams,
00050         unsigned int out_streams);
00051
00052     inline std::string role() override
00053     {
00054         return role_;
00055     }
00056
00057     //! Returns "GPS_L2C_Telemetry_Decoder"
00058     inline std::string implementation() override
00059     {

```



```

00060         return "GPS_L2C_Telemetry_Decoder";
00061     }
00062
00063     void connect(gr::top_block_sptr top_block) override;
00064     void disconnect(gr::top_block_sptr top_block) override;
00065     gr::basic_block_sptr get_left_block() override;
00066     gr::basic_block_sptr get_right_block() override;
00067
00068     void set_satellite(const Gnss_Satellite& satellite) override;
00069
00070     inline void set_channel(int channel) override { telemetry_decoder_>set_channel(channel); }
00071
00072     inline void reset() override
00073     {
00074         telemetry_decoder_>reset();
00075     }
00076
00077     inline size_t item_size() override
00078     {
00079         return sizeof(Gnss_Synchro);
00080     }
00081
00082 private:
00083     gps_l2c_telemetry_decoder_gs_sptr telemetry_decoder_;
00084     Gnss_Satellite satellite_;
00085     Tlm_Conf tlm_parameters_;
00086     std::string role_;
00087     unsigned int in_streams_;
00088     unsigned int out_streams_;
00089 };
00090
00091
00092 /** \} */
00093 /** \} */
00094 #endif // GNSS_SDR_GPS_L2C_TELEMETRY_DECODER_H

```

## 11.438 gps\_l5\_telemetry\_decoder.h File Reference

Interface of an adapter of a GPS L5 (CNAV) data decoder block to a [TelemetryDecoderInterface](#).

```

#include "gnss_satellite.h"
#include "gnss_synchro.h"
#include "gps_l5_telemetry_decoder_gs.h"
#include "telemetry_decoder_interface.h"
#include "tlm_conf.h"
#include <gnuradio/runtime_types.h>
#include <cstdint>
#include <string>

```

Include dependency graph for gps\_l5\_telemetry\_decoder.h:



### Classes

- class [GpsL5TelemetryDecoder](#)

*This class implements a NAV data decoder for GPS L5.*

### 11.438.1 Detailed Description

Interface of an adapter of a GPS L5 (CNAV) data decoder block to a [TelemetryDecoderInterface](#).

#### Author

Antonio Ramos, 2017. antonio.ramos(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [gps\\_l5\\_telemetry\\_decoder.h](#).

---

## 11.439 gps\_l5\_telemetry\_decoder.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file gps_l5_telemetry_decoder.h
00003  * \brief Interface of an adapter of a GPS L5 (CNAV) data decoder block
00004  * to a TelemetryDecoderInterface
00005  * \author Antonio Ramos, 2017. antonio.ramos(at)cttc.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018
00019 #ifndef GNSS_SDR_GPS_L5_TELEMETRY_DECODER_H
00020 #define GNSS_SDR_GPS_L5_TELEMETRY_DECODER_H
00021
00022
00023 #include "gnss_satellite.h"
00024 #include "gnss_synchro.h"
00025 #include "gps_l5_telemetry_decoder_gs.h"
00026 #include "telemetry_decoder_interface.h"
00027 #include "tlm_conf.h"
00028 #include <gnuradio/runtime_types.h> // for basic_block_sptr, top_block_sptr
00029 #include <cstdint> // for size_t
00030 #include <string>
00031
00032 /** \addtogroup Telemetry_Decoder
00033  * \{ */
00034 /** \addtogroup Telemetry_Decoder_adapters
00035  * \{ */
00036
00037
00038 class ConfigurationInterface;
00039
00040 /*!
00041 * \brief This class implements a NAV data decoder for GPS L5
00042 */
00043 class GpsL5TelemetryDecoder : public TelemetryDecoderInterface
00044 {
00045 public:
00046     GpsL5TelemetryDecoder(
00047         const ConfigurationInterface* configuration,
00048         const std::string& role,
00049         unsigned int in_streams,
00050         unsigned int out_streams);
00051
00052     inline std::string role() override
00053     {
00054         return role_;
00055     }
00056
00057     //! Returns "GPS_L5_Telemetry_Decoder"
00058     inline std::string implementation() override
00059     {
00060         return "GPS_L5_Telemetry_Decoder";
00061     }
00062
00063     void connect(gr::top_block_sptr top_block) override;
00064     void disconnect(gr::top_block_sptr top_block) override;
00065     gr::basic_block_sptr get_left_block() override;
00066     gr::basic_block_sptr get_right_block() override;
00067
00068     void set_satellite(const Gnss_Satellite& satellite) override;
00069
00070     inline void set_channel(int channel) override { telemetry_decoder->set_channel(channel); }
00071
00072     inline void reset() override
00073     {
00074         telemetry_decoder->reset();
00075     }
00076
00077     inline size_t item_size() override
00078     {
00079         return sizeof(Gnss_Synchro);
00080     }
00081
00082 private:
00083     gps_l5_telemetry_decoder_gs_sptr telemetry_decoder_;

```

```

00084     Gnss_Satellite satellite_;
00085     Tlm_Conf tlm_parameters_;
00086     std::string dump_filename_;
00087     std::string role_;
00088     unsigned int in_streams_;
00089     unsigned int out_streams_;
00090 };
00091
00092
00093 /** \} */
00094 /** \} */
00095 #endif // GNSS_SDR_GPS_L5_TELEMETRY_DECODER_H

```

## 11.440 sbas\_l1\_telemetry\_decoder.h File Reference

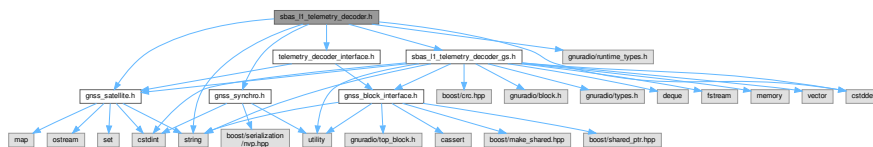
Interface of an adapter of a SBAS telemetry data decoder block to a [TelemetryDecoderInterface](#).

```

#include "gnss_satellite.h"
#include "gnss_synchro.h"
#include "sbas_l1_telemetry_decoder_gs.h"
#include "telemetry_decoder_interface.h"
#include <gnuradio/runtime_types.h>
#include <cstdio>
#include <string>

```

Include dependency graph for sbas\_l1\_telemetry\_decoder.h:



### Classes

- class [SbasL1TelemetryDecoder](#)

*This class implements a NAV data decoder for SBAS frames in L1 radio link.*

### 11.440.1 Detailed Description

Interface of an adapter of a SBAS telemetry data decoder block to a [TelemetryDecoderInterface](#).

#### Author

Daniel Fehr 2013. [daniel.co\(at\)bluewin.ch](mailto:daniel.co(at)bluewin.ch)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [sbas\\_l1\\_telemetry\\_decoder.h](#).

---

## 11.441 sbas\_l1\_telemetry\_decoder.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file sbas_l1_telemetry_decoder.h
00003  * \brief Interface of an adapter of a SBAS telemetry data decoder block
00004  * to a TelemetryDecoderInterface
00005  * \author Daniel Fehr 2013. daniel.co\(at\)bluewin.ch
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later

```

```

00014  *
00015  * -----
00016  */
00017
00018
00019 #ifndef GNSS_SDR_SBAS_L1_TELEMETRY_DECODER_H
00020 #define GNSS_SDR_SBAS_L1_TELEMETRY_DECODER_H
00021
00022
00023 #include "gnss_satellite.h" // for Gnss_Satellite
00024 #include "gnss_synchro.h"
00025 #include "sbas_l1_telemetry_decoder_gs.h"
00026 #include "telemetry_decoder_interface.h"
00027 #include <gnuradio/runtime_types.h> // for basic_block_sptr, top_block_sptr
00028 #include <cstdint> // for size_t
00029 #include <string>
00030
00031 /** \addtogroup Telemetry_Decoder
00032  * \{ */
00033 /** \addtogroup Telemetry_Decoder_adapters
00034  * \{ */
00035
00036 class ConfigurationInterface;
00037
00038 /*!
00039  * \brief This class implements a NAV data decoder for SBAS frames in L1 radio link
00040  */
00041 class SbasL1TelemetryDecoder : public TelemetryDecoderInterface
00042 {
00043 public:
00044     SbasL1TelemetryDecoder(
00045         const ConfigurationInterface* configuration,
00046         const std::string& role,
00047         unsigned int in_streams,
00048         unsigned int out_streams);
00049
00050     inline std::string role() override
00051     {
00052         return role_;
00053     }
00054
00055     /*!
00056     * \brief Returns "SBAS_L1_Telemetry_Decoder"
00057     */
00058     inline std::string implementation() override
00059     {
00060         return "SBAS_L1_Telemetry_Decoder";
00061     }
00062
00063     void connect(gr::top_block_sptr top_block) override;
00064     void disconnect(gr::top_block_sptr top_block) override;
00065     gr::basic_block_sptr get_left_block() override;
00066     gr::basic_block_sptr get_right_block() override;
00067
00068     void set_satellite(const Gnss_Satellite& satellite) override;
00069
00070     inline void set_channel(int channel) override { telemetry_decoder_>set_channel(channel); }
00071
00072     inline void reset() override
00073     {
00074         telemetry_decoder_>reset();
00075     }
00076
00077     inline size_t item_size() override
00078     {
00079         return sizeof(Gnss_Synchro);
00080     }
00081
00082 private:
00083     sbas_l1_telemetry_decoder_gs_sptr telemetry_decoder_;
00084     Gnss_Satellite satellite_;
00085     std::string dump_filename_;
00086     std::string role_;
00087     unsigned int in_streams_;
00088     unsigned int out_streams_;
00089     bool dump_;
00090 };
00091
00092
00093 /** \} */
00094 /** \} */
00095 #endif // GNSS_SDR_SBAS_L1_TELEMETRY_DECODER_H

```

## 11.442 beidou\_b1i\_telemetry\_decoder\_gs.h File Reference

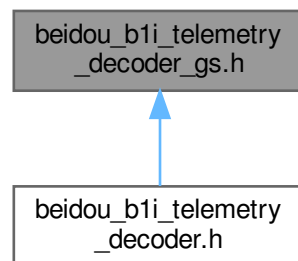
Implementation of a BEIDOU B1I DNAV data decoder block.

```
#include "beidou_dnav_navigation_message.h"
#include "gnss_block_interface.h"
#include "gnss_satellite.h"
#include "nav_message_packet.h"
#include "tlm_conf.h"
#include "tlm_crc_stats.h"
#include <boost/circular_buffer.hpp>
#include <gnuradio/block.h>
#include <gnuradio/types.h>
#include <array>
#include <cstdint>
#include <fstream>
#include <memory>
#include <string>
```

Include dependency graph for beidou\_b1i\_telemetry\_decoder\_gs.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [beidou\\_b1i\\_telemetry\\_decoder\\_gs](#)

*This class implements a block that decodes the BeiDou DNAV data.*

### Typedefs

- using [beidou\\_b1i\\_telemetry\\_decoder\\_gs\\_sptr](#) = `gnss_shared_ptr<beidou\_b1i\_telemetry\_decoder\_gs>`

### Functions

- `beidou_b1i_telemetry_decoder_gs_sptr beidou\_b1i\_make\_telemetry\_decoder\_gs (const Gnss\_Satellite &satellite, const Tlm\_Conf &conf)`

### 11.442.1 Detailed Description

Implementation of a BEIDOU B1I DNAV data decoder block.  
Code added as part of GSoC 2018 program.

#### Author

Damian Miralles, 2018. dmiralles2009(at)gmail.com

Sergi Segura, 2018. sergi.segura.munoz(at)gmail.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [beidou\\_b1i\\_telemetry\\_decoder\\_gs.h](#).

---

### 11.443 beidou\_b1i\_telemetry\_decoder\_gs.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file beidou_b1i_telemetry_decoder_gs.h
00003  * \brief Implementation of a BEIDOU B1I DNAV data decoder block
00004  * \details Code added as part of GSoC 2018 program.
00005  * \author Damian Miralles, 2018. dmiralles2009(at)gmail.com
00006  * \author Sergi Segura, 2018. sergi.segura.munoz(at)gmail.es
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019 #ifndef GNSS_SDR_BEIDOU_B1I_TELEMETRY_DECODER_GS_H
00020 #define GNSS_SDR_BEIDOU_B1I_TELEMETRY_DECODER_GS_H
00021
00022
00023 #include "beidou_dnav_navigation_message.h"
00024 #include "gnss_block_interface.h"
00025 #include "gnss_satellite.h"
00026 #include "nav_message_packet.h"
00027 #include "tlm_conf.h"
00028 #include "tlm_crc_stats.h"
00029 #include <boost/circular_buffer.hpp>
00030 #include <gnuradio/block.h> // for block
00031 #include <gnuradio/types.h> // for gr_vector_const_void_star
00032 #include <array>
00033 #include <stdint>
00034 #include <fstream>
00035 #include <memory> // for std::unique_ptr
00036 #include <string>
00037
00038 /** \addtogroup Telemetry_Decoder
00039  * \{ */
00040 /** \addtogroup Telemetry_Decoder_gnuradio_blocks
00041  * \{ */
00042
00043
00044 class beidou_b1i_telemetry_decoder_gs;
00045
00046 using beidou_b1i_telemetry_decoder_gs_sptr = gnss_shared_ptr<beidou_b1i_telemetry_decoder_gs>;
00047
00048 beidou_b1i_telemetry_decoder_gs_sptr beidou_b1i_make_telemetry_decoder_gs(
00049     const Gnss_Satellite &satellite,
00050     const Tlm_Conf &conf);
00051
00052
00053  /*!
00054  * \brief This class implements a block that decodes the BeiDou DNAV data.
00055  * \note Code added as part of GSoC 2018 program
00056  */
00057 class beidou_b1i_telemetry_decoder_gs : public gr::block
00058 {
00059 public:
00060     ~beidou_b1i_telemetry_decoder_gs() override; //!< Class destructor
00061     void set_satellite(const Gnss_Satellite &satellite); //!< Set satellite PRN
00062     void set_channel(int channel); //!< Set receiver's channel
00063     void reset();

```

```

00064
00065     /*!
00066     * \brief This is where all signal processing takes place
00067     */
00068     int general_work(int noutput_items, gr_vector_int &ninput_items,
00069                     gr_vector_const_void_star &input_items, gr_vector_void_star &output_items) override;
00070
00071 private:
00072     friend beidou_bli_telemetry_decoder_gs_sptr beidou_bli_make_telemetry_decoder_gs(
00073         const Gnss_Satellite &satellite,
00074         const Tlm_Conf &conf);
00075
00076     beidou_bli_telemetry_decoder_gs(const Gnss_Satellite &satellite, const Tlm_Conf &conf);
00077
00078     void decode_subframe(float *symbols, double cn0);
00079     void decode_word(int32_t word_counter, const float *enc_word_symbols, int32_t *dec_word_symbols);
00080     void decode_bch15_11_01(const int32_t *bits, std::array<int32_t, 15> &decbits);
00081
00082     // Preamble decoding
00083     std::array<int32_t, BEIDOU_DNAV_PREAMBLE_LENGTH_SYMBOLS> d_preamble_samples{};
00084
00085     std::array<float, BEIDOU_DNAV_PREAMBLE_PERIOD_SYMBOLS> d_subframe_symbols{};
00086
00087     // Storage for incoming data
00088     boost::circular_buffer<float> d_symbol_history;
00089
00090     // Navigation Message variable
00091     Beidou_Dnav_Navigation_Message d_nav;
00092
00093     Nav_Message_Packet d_nav_msg_packet;
00094     std::unique_ptr<Tlm_CRC_Stats> d_Tlm_CRC_Stats;
00095
00096     // Satellite Information and logging capacity
00097     Gnss_Satellite d_satellite;
00098     std::string d_dump_filename;
00099     std::ofstream d_dump_file;
00100
00101     uint64_t d_sample_counter; // Sample counter as an index (1,2,3,..etc) indicating number of
samples processed
00102     uint64_t d_preamble_index; // Index of sample number where preamble was found
00103
00104     int32_t d_channel;
00105     int32_t d_symbols_per_preamble;
00106     int32_t d_samples_per_preamble;
00107     int32_t d_preamble_period_samples;
00108     int32_t d_CRC_error_counter; // Number of failed CRC operations
00109     uint32_t d_required_symbols;
00110     uint32_t d_stat; // Status of decoder
00111
00112     // Values to populate gnss synchronization structure
00113     uint64_t d_last_valid_preamble;
00114     uint32_t d_symbol_duration_ms;
00115     uint32_t d_TOW_at_Preamble_ms;
00116     uint32_t d_TOW_at_current_symbol_ms;
00117
00118     bool d_flag_SOW_set; // Indicates when time of week is set
00119     bool d_flag_frame_sync; // Indicate when a frame sync is achieved
00120     bool d_flag_preamble; // Flag indicating when preamble was found
00121
00122     bool d_flag_valid_word;
00123     bool d_sent_tlm_failed_msg;
00124     bool d_dump;
00125     bool d_dump_mat;
00126     bool d_remove_dat;
00127     bool d_enable_navdata_monitor;
00128     bool d_dump_crc_stats;
00129 };
00130
00131
00132 /** @} */
00133 /** @} */
00134 #endif // GNSS_SDR_BEIDOU_B1I_TELEMETRY_DECODER_GS_H

```

## 11.444 beidou\_b3i\_telemetry\_decoder\_gs.h File Reference

Implementation of a BEIDOU B3I DNAV data decoder block.

```

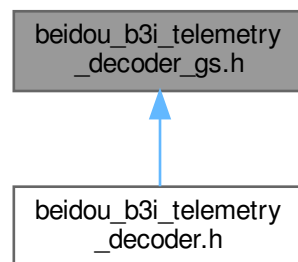
#include "beidou_dnav_navigation_message.h"
#include "gnss_block_interface.h"
#include "gnss_satellite.h"
#include "nav_message_packet.h"
#include "tlm_conf.h"

```

```
#include "tlm_crc_stats.h"
#include <boost/circular_buffer.hpp>
#include <gnuradio/block.h>
#include <gnuradio/types.h>
#include <array>
#include <cstdint>
#include <fstream>
#include <memory>
#include <string>
Include dependency graph for beidou_b3i_telemetry_decoder_gs.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [beidou\\_b3i\\_telemetry\\_decoder\\_gs](#)  
*This class implements a block that decodes the BeiDou DNAV data.*

## Typedefs

- using [beidou\\_b3i\\_telemetry\\_decoder\\_gs\\_sptr](#)

## Functions

- [beidou\\_b3i\\_telemetry\\_decoder\\_gs\\_sptr](#) **beidou\_b3i\_make\_telemetry\_decoder\_gs** (const [Gnss\\_Satellite](#) &satellite, const [Tlm\\_Conf](#) &conf)

### 11.444.1 Detailed Description

Implementation of a BEIDOU B3I DNAV data decoder block.

Author

Damian Miralles, 2019. [dmiralles2009@gmail.com](mailto:dmiralles2009@gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [beidou\\_b3i\\_telemetry\\_decoder\\_gs.h](#).

---



## 11.445 beidou\_b3i\_telemetry\_decoder\_gs.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file beidou_b3i_telemetry_decoder_gs.h
00003  * \brief Implementation of a BEIDOU B3I DNAV data decoder block
00004  * \author Damian Miralles, 2019. dmiralles2009(at)gmail.com
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_BEIDOU_B3I_TELEMETRY_DECODER_GS_H
00018 #define GNSS_SDR_BEIDOU_B3I_TELEMETRY_DECODER_GS_H
00019
00020 #include "beidou_dnav_navigation_message.h"
00021 #include "gnss_block_interface.h"
00022 #include "gnss_satellite.h"
00023 #include "nav_message_packet.h"
00024 #include "tlm_conf.h"
00025 #include "tlm_crc_stats.h"
00026 #include <boost/circular_buffer.hpp>
00027 #include <gnuradio/block.h> // for block
00028 #include <gnuradio/types.h> // for gr_vector_const_void_star
00029 #include <array>
00030 #include <cstdint>
00031 #include <fstream>
00032 #include <memory> // for std::unique_ptr
00033 #include <string>
00034
00035
00036 /** \addtogroup Telemetry_Decoder
00037  * \{ */
00038 /** \addtogroup Telemetry_Decoder_gnuradio_blocks
00039  * \{ */
00040
00041
00042 class beidou_b3i_telemetry_decoder_gs;
00043
00044 using beidou_b3i_telemetry_decoder_gs_sptr =
00045     gnss_shared_ptr<beidou_b3i_telemetry_decoder_gs>;
00046
00047 beidou_b3i_telemetry_decoder_gs_sptr beidou_b3i_make_telemetry_decoder_gs(
00048     const Gnss_Satellite &satellite,
00049     const Tlm_Conf &conf);
00050
00051 /*!
00052  * \brief This class implements a block that decodes the BeiDou DNAV data.
00053  */
00054 class beidou_b3i_telemetry_decoder_gs : public gr::block
00055 {
00056 public:
00057     ~beidou_b3i_telemetry_decoder_gs() override;           //!< Class destructor
00058     void set_satellite(const Gnss_Satellite &satellite);    //!< Set satellite PRN
00059     void set_channel(int channel);                          //!< Set receiver's channel
00060     void reset();
00061
00062     /*!
00063     * \brief This is where all signal processing takes place
00064     */
00065     int general_work(int noutput_items, gr_vector_int &ninput_items,
00066         gr_vector_const_void_star &input_items,
00067         gr_vector_void_star &output_items) override;
00068
00069 private:
00070     friend beidou_b3i_telemetry_decoder_gs_sptr beidou_b3i_make_telemetry_decoder_gs(
00071         const Gnss_Satellite &satellite,
00072         const Tlm_Conf &conf);
00073
00074     beidou_b3i_telemetry_decoder_gs(const Gnss_Satellite &satellite, const Tlm_Conf &conf);
00075
00076     void decode_subframe(float *symbols, double cn0);
00077     void decode_word(int32_t word_counter, const float *enc_word_symbols,
00078         int32_t *dec_word_symbols);
00079     void decode_bch15_11_01(const int32_t *bits, std::array<int32_t, 15> &decbits);
00080
00081     // Preamble decoding
00082     std::array<int32_t, BEIDOU_DNAV_PREAMBLE_LENGTH_SYMBOLS> d_preamble_samples{};
00083     std::array<float, BEIDOU_DNAV_PREAMBLE_PERIOD_SYMBOLS> d_subframe_symbols{};

```

```

00084
00085 // Storage for incoming data
00086 boost::circular_buffer<float> d_symbol_history;
00087
00088 // Navigation Message variable
00089 Beidou_Dnav_Navigation_Message d_nav;
00090 Gnss_Satellite d_satellite;
00091
00092 Nav_Message_Packet d_nav_msg_packet;
00093 std::unique_ptr<Tlm_CRC_Stats> d_Tlm_CRC_Stats;
00094
00095 std::string d_dump_filename;
00096 std::ofstream d_dump_file;
00097
00098 uint64_t d_sample_counter; // Sample counter as an index (1,2,3,..etc) indicating number of
samples processed
00099 uint64_t d_preamble_index; // Index of sample number where preamble was found
00100 uint32_t d_required_symbols;
00101 uint32_t d_stat; // Status of decoder
00102
00103 int32_t d_channel;
00104 int32_t d_CRC_error_counter; // Number of failed CRC operations
00105 int32_t d_symbols_per_preamble;
00106 int32_t d_samples_per_preamble;
00107 int32_t d_preamble_period_samples;
00108
00109 // Values to populate gnss synchronization structure
00110 uint64_t d_last_valid_preamble;
00111 uint32_t d_symbol_duration_ms;
00112 uint32_t d_TOW_at_Preamble_ms;
00113 uint32_t d_TOW_at_current_symbol_ms;
00114
00115 bool d_flag_SOW_set; // Indicates when time of week is set
00116 bool d_flag_frame_sync; // Indicate when a frame sync is achieved
00117 bool d_flag_preamble; // Flag indicating when preamble was found
00118 bool d_flag_valid_word;
00119 bool d_sent_tlm_failed_msg;
00120 bool d_dump;
00121 bool d_dump_mat;
00122 bool d_remove_dat;
00123 bool d_enable_navdata_monitor;
00124 bool d_dump_crc_stats;
00125 };
00126
00127
00128 /** \} */
00129 /** \} */
00130 #endif // GNSS_SDR_BEIDOU_B3I_TELEMETRY_DECODER_GS_H

```

## 11.446 galileo\_telemetry\_decoder\_gs.h File Reference

Implementation of a Galileo unified INAV and FNAV message demodulator block.

```

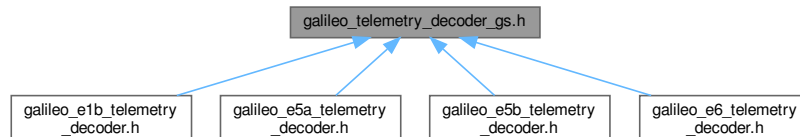
#include "galileo_cnav_message.h"
#include "galileo_fnav_message.h"
#include "galileo_inav_message.h"
#include "gnss_block_interface.h"
#include "gnss_satellite.h"
#include "gnss_time.h"
#include "nav_message_packet.h"
#include "tlm_conf.h"
#include <boost/circular_buffer.hpp>
#include <gnuradio/block.h>
#include <gnuradio/types.h>
#include <pmt/pmt.h>
#include <cstdint>
#include <fstream>
#include <memory>
#include <string>
#include <vector>

```

Include dependency graph for galileo\_telemetry\_decoder\_gs.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [galileo\\_telemetry\\_decoder\\_gs](#)  
*This class implements a block that decodes the INAV and FNAV data defined in Galileo ICD.*

## Typedefs

- using [galileo\\_telemetry\\_decoder\\_gs\\_sptr](#) = gnss\_shared\_ptr<[galileo\\_telemetry\\_decoder\\_gs](#)>

## Functions

- [galileo\\_telemetry\\_decoder\\_gs\\_sptr](#) **galileo\_make\_telemetry\_decoder\_gs** (const [Gnss\\_Satellite](#) &satellite, const [Tlm\\_Conf](#) &conf, int frame\_type)

## 11.446.1 Detailed Description

Implementation of a Galileo unified INAV and FNAV message demodulator block.

### Author

Javier Arribas 2018. jarribas(at)cttc.es

Carles Fernandez, 2021-2022. cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [galileo\\_telemetry\\_decoder\\_gs.h](#).

---

## 11.447 galileo\_telemetry\_decoder\_gs.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file galileo_telemetry_decoder_gs.h
00003  * \brief Implementation of a Galileo unified INAV and FNAV message demodulator
00004  * block
00005  * \author Javier Arribas 2018. jarribas(at)cttc.es
00006  * \author Carles Fernandez, 2021-2022. cfernandez(at)cttc.es
00007  *
00008  *
00009  * -----
00010  *
00011  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00012  * This file is part of GNSS-SDR.
00013  *
```

```

00014  * Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors)
00015  * SPDX-License-Identifier: GPL-3.0-or-later
00016  *
00017  * -----
00018  */
00019
00020
00021 #ifndef GNSS_SDR_GALILEO_TELEMETRY_DECODER_GS_H
00022 #define GNSS_SDR_GALILEO_TELEMETRY_DECODER_GS_H
00023
00024 #include "galileo_cnav_message.h" // for Galileo_Cnav_Message
00025 #include "galileo_fnav_message.h" // for Galileo_Fnav_Message
00026 #include "galileo_inav_message.h" // for Galileo_Inav_Message
00027 #include "gnss_block_interface.h" // for gnss_shared_ptr (adapts smart pointer type to GNU Radio
    version)
00028 #include "gnss_satellite.h" // for Gnss_Satellite
00029 #include "gnss_time.h" // for GnssTime
00030 #include "nav_message_packet.h" // for Nav_Message_Packet
00031 #include "tlm_conf.h" // for Tlm_Conf
00032 #include <boost/circular_buffer.hpp> // for boost::circular_buffer
00033 #include <gnuradio/block.h> // for block
00034 #include <gnuradio/types.h> // for gr_vector_const_void_star
00035 #include <pmt/pmt.h> // for pmt::pmt_t
00036 #include <cstdlib> // for int32_t, uint32_t
00037 #include <fstream> // for std::ofstream
00038 #include <memory> // for std::unique_ptr
00039 #include <string> // for std::string
00040 #include <vector> // for std::vector
00041
00042 /** \addtogroup Telemetry_Decoder
00043  * \{ */
00044 /** \addtogroup Telemetry_Decoder_gnuradio_blocks
00045  * \{ */
00046
00047 class Viterbi_Decoder; // forward declaration
00048 class Tlm_CRC_Stats; // forward declaration
00049 class galileo_telemetry_decoder_gs; // forward declaration
00050
00051 using galileo_telemetry_decoder_gs_sptr = gnss_shared_ptr<galileo_telemetry_decoder_gs>;
00052
00053 galileo_telemetry_decoder_gs_sptr galileo_make_telemetry_decoder_gs(
00054     const Gnss_Satellite &satellite,
00055     const Tlm_Conf &conf,
00056     int frame_type);
00057
00058 /*!
00059  * \brief This class implements a block that decodes the INAV and FNAV data defined in Galileo ICD
00060  */
00061 class galileo_telemetry_decoder_gs : public gr::block
00062 {
00063 public:
00064     ~galileo_telemetry_decoder_gs() override;
00065     void set_satellite(const Gnss_Satellite &satellite); //!< Set satellite PRN
00066     void set_channel(int32_t channel); //!< Set receiver's channel
00067     void reset();
00068
00069     /*!
00070     * \brief This is where all signal processing takes place
00071     */
00072     int general_work(int noutput_items, gr_vector_int &ninput_items,
00073         gr_vector_const_void_star &input_items, gr_vector_void_star &output_items) override;
00074 private:
00075     friend galileo_telemetry_decoder_gs_sptr galileo_make_telemetry_decoder_gs(
00076         const Gnss_Satellite &satellite,
00077         const Tlm_Conf &conf,
00078         int frame_type);
00079
00080     galileo_telemetry_decoder_gs(const Gnss_Satellite &satellite, const Tlm_Conf &conf, int
    frame_type);
00082
00083     void check_tlm_separation();
00084     void msg_handler_read_galileo_tow_map(const pmt::pmt_t &msg);
00085     void deinterleave(int32_t rows, int32_t cols, const float *in, float *out);
00086     void decode_INAV_word(float *page_part_symbols, int32_t frame_length, double cn0);
00087     void decode_FNAV_word(float *page_symbols, int32_t frame_length, double cn0);
00088     void decode_CNAV_word(uint64_t time_stamp, float *page_symbols, int32_t page_length, double cn0);
00089
00090     std::unique_ptr<Viterbi_Decoder> d_viterbi;
00091     std::vector<int32_t> d_preamble_samples;
00092     std::vector<float> d_page_part_symbols;
00093
00094     std::string d_dump_filename;
00095     std::ofstream d_dump_file;
00096
00097     boost::circular_buffer<float> d_symbol_history;
00098

```

```

00099     Gnss_Satellite d_satellite;
00100
00101     // navigation message vars
00102     Galileo_Cnav_Message d_cnav_nav;
00103     Galileo_Inav_Message d_inav_nav;
00104     Galileo_Fnav_Message d_fnav_nav;
00105
00106     Nav_Message_Packet d_nav_msg_packet;
00107     GnssTime d_current_timetag{};
00108
00109     std::unique_ptr<Tlm_CRC_Stats> d_Tlm_CRC_Stats;
00110
00111     double d_delta_t; // GPS-GALILEO time offset
00112
00113     uint64_t d_symbol_counter;
00114     uint64_t d_preamble_index;
00115     uint64_t d_last_valid_preamble;
00116     uint64_t d_received_sample_counter;
00117
00118     int32_t d_mm;
00119     int32_t d_codelength;
00120     int32_t d_datalength;
00121     int32_t d_frame_type;
00122     int32_t d_bits_per_preamble;
00123     int32_t d_samples_per_preamble;
00124     int32_t d_preamble_period_symbols;
00125     int32_t d_CRC_error_counter;
00126     int32_t d_channel;
00127     int32_t d_flag_even_word_arrived;
00128
00129     uint32_t d_PRN_code_period_ms;
00130     uint32_t d_required_symbols;
00131     uint32_t d_frame_length_symbols;
00132     uint32_t d_stat;
00133     uint32_t d_TOW_at_Preamble_ms;
00134     uint32_t d_TOW_at_current_symbol_ms;
00135     uint32_t d_max_symbols_without_valid_frame;
00136     uint32_t d_received_tow_ms;
00137
00138     char d_band; // This variable will store which band we are dealing with (Galileo E1 or E5b)
00139
00140     bool d_sent_tlm_failed_msg;
00141     bool d_flag_frame_sync;
00142     bool d_flag_PLL_180_deg_phase_locked;
00143     bool d_flag_preamble;
00144     bool d_dump;
00145     bool d_dump_mat;
00146     bool d_remove_dat;
00147     bool d_first_eph_sent;
00148     bool d_cnav_dummy_page;
00149     bool d_print_cnav_page;
00150     bool d_enable_navdata_monitor;
00151     bool d_dump_crc_stats;
00152     bool d_enable_reed_solomon_inav;
00153     bool d_valid_timetag;
00154     bool d_E6_TOW_set;
00155     bool d_there_are_e1_channels;
00156     bool d_there_are_e6_channels;
00157     bool d_use_ced;
00158 };
00159
00160
00161 /** \} */
00162 /** \} */
00163 #endif // GNSS_SDR_GALILEO_TELEMETRY_DECODER_GS_H

```

## 11.448 glonass\_l1\_ca\_telemetry\_decoder\_gs.h File Reference

Implementation of a GLONASS L1 C/A NAV data decoder block.

```

#include "GLONASS_L1_L2_CA.h"
#include "glonass_gnav_navigation_message.h"
#include "gnss_block_interface.h"
#include "gnss_satellite.h"
#include "gnss_synchro.h"
#include "nav_message_packet.h"
#include "tlm_conf.h"
#include "tlm_crc_stats.h"
#include <boost/circular_buffer.hpp>
#include <gnuradio/block.h>

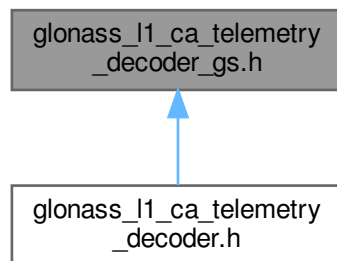
```

```
#include <gnuradio/types.h>
#include <array>
#include <stdint>
#include <fstream>
#include <memory>
#include <string>
```

Include dependency graph for `glonass_l1_ca_telemetry_decoder_gs.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [glonass\\_l1\\_ca\\_telemetry\\_decoder\\_gs](#)

*This class implements a block that decodes the GNAV data defined in GLONASS ICD v5.1.*

## Typedefs

- using [glonass\\_l1\\_ca\\_telemetry\\_decoder\\_gs\\_sptr](#) = `gnss_shared_ptr<glonass\_l1\_ca\_telemetry\_decoder\_gs>`

## Functions

- `glonass_l1_ca_telemetry_decoder_gs_sptr glonass\_l1\_ca\_make\_telemetry\_decoder\_gs (const Gnss\_Satellite &satellite, const Tlm\_Conf &conf)`

### 11.448.1 Detailed Description

Implementation of a GLONASS L1 C/A NAV data decoder block.

#### Note

Code added as part of GSoC 2017 program

#### Author

Damian Miralles, 2017. [dmiralles2009\(at\)gmail.com](mailto:dmiralles2009(at)gmail.com)K

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [glonass\\_l1\\_ca\\_telemetry\\_decoder\\_gs.h](#).

## 11.449 glonass\_l1\_ca\_telemetry\_decoder\_gs.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file glonass_l1_ca_telemetry_decoder_gs.h
00003  * \brief Implementation of a GLONASS L1 C/A NAV data decoder block
00004  * \note Code added as part of GSOC 2017 program
00005  * \author Damian Miralles, 2017. dmiralles2009(at)gmail.comK
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_GLOMSS_L1_CA_TELEMETRY_DECODER_GS_H
00019 #define GNSS_SDR_GLOMSS_L1_CA_TELEMETRY_DECODER_GS_H
00020
00021
00022 #include "GLONASS_L1_L2_CA.h"
00023 #include "glonass_gnav_navigation_message.h"
00024 #include "gnss_block_interface.h"
00025 #include "gnss_satellite.h"
00026 #include "gnss_synchro.h"
00027 #include "nav_message_packet.h"
00028 #include "t1m_conf.h"
00029 #include "t1m_crc_stats.h"
00030 #include <boost/circular_buffer.hpp>
00031 #include <gnuradio/block.h> // for block
00032 #include <gnuradio/types.h> // for gr_vector_const_void_star
00033 #include <array>
00034 #include <stdint>
00035 #include <fstream> // for ofstream
00036 #include <memory> // for std::unique_ptr
00037 #include <string>
00038
00039 /** \addtogroup Telemetry_Decoder
00040  * \{ */
00041 /** \addtogroup Telemetry_Decoder_gnuradio_blocks
00042  * \{ */
00043
00044
00045 class glonass_l1_ca_telemetry_decoder_gs;
00046
00047 using glonass_l1_ca_telemetry_decoder_gs_sptr = gnss_shared_ptr<glonass_l1_ca_telemetry_decoder_gs>;
00048
00049 glonass_l1_ca_telemetry_decoder_gs_sptr glonass_l1_ca_make_telemetry_decoder_gs(
00050     const Gnss_Satellite &satellite,
00051     const T1m_Conf &conf);
00052
00053 /*!
00054  * \brief This class implements a block that decodes the GNAV data defined in GLONASS ICD v5.1
00055  * \note Code added as part of GSOC 2017 program
00056  * \see <a
00057  href="http://russianspacesystems.ru/wp-content/uploads/2016/08/ICD_GLONASS_eng_v5.1.pdf">GLONASS
00058  ICD</a>
00059  *
00060  */
00061 class glonass_l1_ca_telemetry_decoder_gs : public gr::block
00062 {
00063 public:
00064     ~glonass_l1_ca_telemetry_decoder_gs() override; //!< Class destructor
00065     void set_satellite(const Gnss_Satellite &satellite); //!< Set satellite PRN
00066     void set_channel(int32_t channel); //!< Set receiver's channel
00067     inline void reset() {};
00068
00069     /*!
00070     * \brief This is where all signal processing takes place
00071     */
00072     int general_work(int noutput_items, gr_vector_int &ninput_items,
00073         gr_vector_const_void_star &input_items, gr_vector_void_star &output_items) override;
00074 private:
00075     friend glonass_l1_ca_telemetry_decoder_gs_sptr glonass_l1_ca_make_telemetry_decoder_gs(
00076         const Gnss_Satellite &satellite,
00077         const T1m_Conf &conf);
00078     glonass_l1_ca_telemetry_decoder_gs(const Gnss_Satellite &satellite, const T1m_Conf &conf);
00079     const std::array<uint16_t, GLONASS_GNAV_PREAMBLE_LENGTH_BITS>
00080         d_preambles_bits{GLONASS_GNAV_PREAMBLE};

```

```

00081
00082     const int32_t d_symbols_per_preamble = GLONASS_GNAV_PREAMBLE_LENGTH_SYMBOLS;
00083
00084     void decode_string(const double *symbols, int32_t frame_length, double cn0);
00085
00086     // Help with coherent tracking
00087
00088     // Preamble decoding
00089     std::array<int32_t, GLONASS_GNAV_PREAMBLE_LENGTH_SYMBOLS> d_preambles_symbols{};
00090
00091     // Storage for incoming data
00092     boost::circular_buffer<Gnss_Synchro> d_symbol_history;
00093
00094     // Navigation Message variable
00095     Glonass_Gnav_Navigation_Message d_nav;
00096
00097     Gnss_Satellite d_satellite;
00098
00099     Nav_Message_Packet d_nav_msg_packet;
00100     std::unique_ptr<Tlm_CRC_Stats> d_Tlm_CRC_Stats;
00101
00102     std::string d_dump_filename;
00103     std::ofstream d_dump_file;
00104
00105     double d_preamble_time_samples;
00106     double d_TOW_at_current_symbol;
00107
00108     // Variables for internal functionality
00109     uint64_t d_sample_counter; // Sample counter as an index (1,2,3,..etc) indicating number of
samples processed
00110     uint64_t d_preamble_index; // Index of sample number where preamble was found
00111
00112     uint32_t d_stat; // Status of decoder
00113     int32_t d_CRC_error_counter; // Number of failed CRC operations
00114     int32_t d_channel;
00115
00116     bool d_flag_frame_sync; // Indicate when a frame sync is achieved
00117     bool d_flag_preamble; // Flag indicating when preamble was found
00118     bool d_dump;
00119     bool d_dump_mat;
00120     bool d_remove_dat;
00121     bool d_enable_navdata_monitor;
00122     bool d_dump_crc_stats;
00123 };
00124
00125
00126 /** @} */
00127 /** @} */
00128 #endif // GNSS_SDR_GLOASS_L1_CA_TELEMETRY_DECODER_GS_H

```

## 11.450 glonass\_l2\_ca\_telemetry\_decoder\_gs.h File Reference

Implementation of a GLONASS L2 C/A NAV data decoder block.

```

#include "GLONASS_L1_L2_CA.h"
#include "glonass_gnav_navigation_message.h"
#include "gnss_block_interface.h"
#include "gnss_satellite.h"
#include "gnss_synchro.h"
#include "nav_message_packet.h"
#include "tlm_conf.h"
#include "tlm_crc_stats.h"
#include <boost/circular_buffer.hpp>
#include <gnuradio/block.h>
#include <gnuradio/types.h>
#include <array>
#include <stdint>
#include <fstream>
#include <memory>
#include <string>

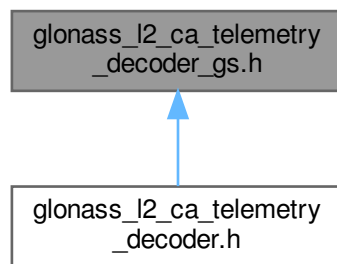
```



Include dependency graph for glonass\_l2\_ca\_telemetry\_decoder\_gs.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [glonass\\_l2\\_ca\\_telemetry\\_decoder\\_gs](#)

*This class implements a block that decodes the GNAV data defined in GLONASS ICD v5.1.*

## Typedefs

- using [glonass\\_l2\\_ca\\_telemetry\\_decoder\\_gs\\_sptr](#) = gnss\_shared\_ptr<[glonass\\_l2\\_ca\\_telemetry\\_decoder\\_gs](#)>

## Functions

- [glonass\\_l2\\_ca\\_telemetry\\_decoder\\_gs\\_sptr](#) **glonass\_l2\_ca\_make\_telemetry\_decoder\_gs** (const [Gnss\\_Satellite](#) &satellite, const [Tlm\\_Conf](#) &conf)

### 11.450.1 Detailed Description

Implementation of a GLONASS L2 C/A NAV data decoder block.

#### Author

Damian Miralles, 2018. [dmiralles2009\(at\)gmail.com](mailto:dmiralles2009(at)gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [glonass\\_l2\\_ca\\_telemetry\\_decoder\\_gs.h](#).

---

## 11.451 glonass\_l2\_ca\_telemetry\_decoder\_gs.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file glonass_l2_ca_telemetry_decoder_gs.h
00003  * \brief Implementation of a GLONASS L2 C/A NAV data decoder block
00004  * \author Damian Miralles, 2018. dmiralles2009\(at\)gmail.com
00005  *
```

```

00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_GLONASS_L2_CA_TELEMETRY_DECODER_GS_H
00018 #define GNSS_SDR_GLONASS_L2_CA_TELEMETRY_DECODER_GS_H
00019
00020
00021 #include "GLONASS_L1_L2_CA.h"
00022 #include "glonass_gnav_navigation_message.h"
00023 #include "gnss_block_interface.h"
00024 #include "gnss_satellite.h"
00025 #include "gnss_synchro.h"
00026 #include "nav_message_packet.h"
00027 #include "tlm_conf.h"
00028 #include "tlm_crc_stats.h"
00029 #include <boost/circular_buffer.hpp>
00030 #include <gnuradio/block.h>
00031 #include <gnuradio/types.h> // for gr_vector_const_void_star
00032 #include <array>
00033 #include <cstdint>
00034 #include <fstream>
00035 #include <memory> // for std::unique_ptr
00036 #include <string>
00037
00038 /** \addtogroup Telemetry_Decoder
00039  * \{ */
00040 /** \addtogroup Telemetry_Decoder_gnuradio_blocks
00041  * \{ */
00042
00043
00044 class glonass_l2_ca_telemetry_decoder_gs;
00045
00046 using glonass_l2_ca_telemetry_decoder_gs_sptr = gnss_shared_ptr<glonass_l2_ca_telemetry_decoder_gs>;
00047
00048 glonass_l2_ca_telemetry_decoder_gs_sptr glonass_l2_ca_make_telemetry_decoder_gs(
00049     const Gnss_Satellite &satellite,
00050     const Tlm_Conf &conf);
00051
00052 /*!
00053  * \brief This class implements a block that decodes the GNAV data defined in GLONASS ICD v5.1
00054  * \see <a
00055     href="http://russianspacesystems.ru/wp-content/uploads/2016/08/ICD_GLONASS_eng_v5.1.pdf">GLONASS
00056     ICD</a>
00057  *
00058  */
00057 class glonass_l2_ca_telemetry_decoder_gs : public gr::block
00058 {
00059 public:
00060     ~glonass_l2_ca_telemetry_decoder_gs() override; //!< Class destructor
00061     void set_satellite(const Gnss_Satellite &satellite); //!< Set satellite PRN
00062     void set_channel(int32_t channel); //!< Set receiver's channel
00063     inline void reset() {};
00064
00065     /*!
00066     * \brief This is where all signal processing takes place
00067     */
00068     int general_work(int noutput_items, gr_vector_int &ninput_items,
00069         gr_vector_const_void_star &input_items, gr_vector_void_star &output_items) override;
00070
00071 private:
00072     friend glonass_l2_ca_telemetry_decoder_gs_sptr glonass_l2_ca_make_telemetry_decoder_gs(
00073         const Gnss_Satellite &satellite,
00074         const Tlm_Conf &conf);
00075
00076     glonass_l2_ca_telemetry_decoder_gs(const Gnss_Satellite &satellite, const Tlm_Conf &conf);
00077
00078     const std::array<uint16_t, GLONASS_GNAV_PREAMBLE_LENGTH_BITS>
00079     d_preambles_bits{GLONASS_GNAV_PREAMBLE};
00080
00081     const int32_t d_symbols_per_preamble = GLONASS_GNAV_PREAMBLE_LENGTH_SYMBOLS;
00082
00083     void decode_string(const double *symbols, int32_t frame_length, double cn0);
00084
00085     // Storage for incoming data
00086     boost::circular_buffer<Gnss_Synchro> d_symbol_history;
00087
00088     std::array<int32_t, GLONASS_GNAV_PREAMBLE_LENGTH_SYMBOLS> d_preambles_symbols{};
00089
00090     // Navigation Message variable

```

```

00090     Glonass_Gnav_Navigation_Message d_nav;
00091     Gnss_Satellite d_satellite;
00092
00093     Nav_Message_Packet d_nav_msg_packet;
00094     std::unique_ptr<Tlm_CRC_Stats> d_Tlm_CRC_Stats;
00095
00096     std::string d_dump_filename;
00097     std::ofstream d_dump_file;
00098
00099     double d_preamble_time_samples;
00100     double d_TOW_at_current_symbol;
00101
00102     uint64_t d_sample_counter; // Sample counter as an index (1,2,3,..etc) indicating number of
samples processed
00103     uint64_t d_preamble_index; // Index of sample number where preamble was found
00104     uint32_t d_stat;           // Status of decoder
00105
00106     int32_t d_CRC_error_counter; // Number of failed CRC operations
00107     int32_t d_channel;
00108
00109     bool d_flag_frame_sync; // Indicate when a frame sync is achieved
00110     bool d_flag_preamble;   // Flag indicating when preamble was found
00111     bool d_dump;
00112     bool d_dump_mat;
00113     bool d_remove_dat;
00114     bool d_enable_navdata_monitor;
00115     bool d_dump_crc_stats;
00116 };
00117
00118
00119 /** \} */
00120 /** \} */
00121 #endif // GNSS_SDR_GLOPASS_L2_CA_TELEMETRY_DECODER_GS_H

```

## 11.452 gps\_l1\_ca\_telemetry\_decoder\_gs.h File Reference

Interface of a NAV message demodulator block based on Kay Borre book MATLAB-based GPS receiver.

```

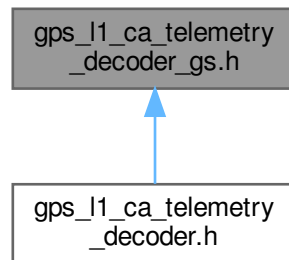
#include "GPS_L1_CA.h"
#include "gnss_block_interface.h"
#include "gnss_satellite.h"
#include "gnss_synchro.h"
#include "gnss_time.h"
#include "gps_navigation_message.h"
#include "nav_message_packet.h"
#include "tlm_conf.h"
#include "tlm_crc_stats.h"
#include <boost/circular_buffer.hpp>
#include <gnuradio/block.h>
#include <gnuradio/types.h>
#include <array>
#include <stdint>
#include <fstream>
#include <memory>
#include <string>

```

Include dependency graph for gps\_l1\_ca\_telemetry\_decoder\_gs.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gps\\_l1\\_ca\\_telemetry\\_decoder\\_gs](#)

*This class implements a block that decodes the NAV data defined in IS-GPS-200M.*

## Typedefs

- using [gps\\_l1\\_ca\\_telemetry\\_decoder\\_gs\\_ptr](#) = `gnss_shared_ptr<gps\_l1\_ca\_telemetry\_decoder\_gs>`

## Functions

- `gps_l1_ca_telemetry_decoder_gs_ptr gps\_l1\_ca\_make\_telemetry\_decoder\_gs (const Gnss\_Satellite &satellite, const Tlm\_Conf &conf)`

## 11.452.1 Detailed Description

Interface of a NAV message demodulator block based on Kay Borre book MATLAB-based GPS receiver.

### 11.452.1.1 autotoc\_md449

#### Author

Javier Arribas, 2011. [jarribas\(at\)cttc.es](mailto:jarribas(at)cttc.es)

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gps\\_l1\\_ca\\_telemetry\\_decoder\\_gs.h](#).

## 11.453 [gps\\_l1\\_ca\\_telemetry\\_decoder\\_gs.h](#)

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file gps_l1_ca_telemetry_decoder_gs.h
00003  * \brief Interface of a NAV message demodulator block based on
00004  * Kay Borre book MATLAB-based GPS receiver
00005  * \author Javier Arribas, 2011. jarribas\(at\)cttc.es
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
  
```

```

00015  */
00016
00017 #ifndef GNSS_SDR_GPS_L1_CA_TELEMETRY_DECODER_GS_H
00018 #define GNSS_SDR_GPS_L1_CA_TELEMETRY_DECODER_GS_H
00019 #include "GPS_L1_CA.h"
00020 #include "gnss_block_interface.h"
00021 #include "gnss_satellite.h"
00022 #include "gnss_synchro.h"
00023 #include "gnss_time.h" // for timetags produced by Tracking
00024 #include "gps_navigation_message.h"
00025 #include "nav_message_packet.h"
00026 #include "tlm_conf.h"
00027 #include "tlm_crc_stats.h"
00028 #include <boost/circular_buffer.hpp>
00029 #include <gnuradio/block.h> // for block
00030 #include <gnuradio/types.h> // for gr_vector_const_void_star
00031 #include <array> // for array
00032 #include <stdint> // for int32_t
00033 #include <fstream> // for ofstream
00034 #include <memory> // for std::unique_ptr
00035 #include <string> // for string
00036
00037 /** \addtogroup Telemetry_Decoder
00038  * \{ */
00039 /** \addtogroup Telemetry_Decoder_gnuradio_blocks telemetry_decoder_gr_blocks
00040  * GNU Radio blocks for the demodulation of GNSS navigation messages.
00041  * \{ */
00042
00043
00044 class gps_l1_ca_telemetry_decoder_gs;
00045
00046 using gps_l1_ca_telemetry_decoder_gs_sptr = gnss_shared_ptr<gps_l1_ca_telemetry_decoder_gs>;
00047
00048 gps_l1_ca_telemetry_decoder_gs_sptr gps_l1_ca_make_telemetry_decoder_gs(
00049     const Gnss_Satellite &satellite,
00050     const Tlm_Conf &conf);
00051
00052 /*!
00053  * \brief This class implements a block that decodes the NAV data defined in IS-GPS-200M
00054  */
00055 class gps_l1_ca_telemetry_decoder_gs : public gr::block
00056 {
00057 public:
00058     ~gps_l1_ca_telemetry_decoder_gs() override;
00059     void set_satellite(const Gnss_Satellite &satellite); //!< Set satellite PRN
00060     void set_channel(int channel); //!< Set receiver's channel
00061     void reset();
00062
00063     /*!
00064      * \brief This is where all signal processing takes place
00065      */
00066     int general_work(int noutput_items, gr_vector_int &ninput_items,
00067         gr_vector_const_void_star &input_items, gr_vector_void_star &output_items) override;
00068
00069 private:
00070     friend gps_l1_ca_telemetry_decoder_gs_sptr gps_l1_ca_make_telemetry_decoder_gs(
00071         const Gnss_Satellite &satellite,
00072         const Tlm_Conf &conf);
00073
00074     gps_l1_ca_telemetry_decoder_gs(const Gnss_Satellite &satellite, const Tlm_Conf &conf);
00075
00076     void check_tlm_separation();
00077     void frame_synchronization(const Gnss_Synchro &current_gs);
00078     bool is_PL_L180_deg_phase_locked();
00079     bool gps_word_parityCheck(uint32_t gpsword);
00080     bool decode_subframe(double cn0, bool flag_invert);
00081
00082     Gps_Navigation_Message d_nav;
00083     Gnss_Satellite d_satellite;
00084     Nav_Message_Packet d_nav_msg_packet;
00085     std::unique_ptr<Tlm_CRC_Stats> d_Tlm_CRC_Stats;
00086
00087     std::array<int32_t, GPS_CA_PREAMBLE_LENGTH_BITS> d_preamble_samples{};
00088
00089     std::string d_dump_filename;
00090     std::ofstream d_dump_file;
00091
00092     boost::circular_buffer<float> d_symbol_history;
00093
00094     uint64_t d_sample_counter;
00095     uint64_t d_preamble_index;
00096     uint64_t d_last_valid_preamble;
00097
00098     int32_t d_bits_per_preamble;
00099     int32_t d_samples_per_preamble;
00100     int32_t d_preamble_period_symbols;
00101     int32_t d_CRC_error_counter;

```

```

00102     int32_t d_channel;
00103
00104     uint32_t d_required_symbols;
00105     uint32_t d_prev_GPS_frame_4bytes;
00106     uint32_t d_max_symbols_without_valid_frame;
00107     uint32_t d_stat;
00108     uint32_t d_TOW_at_Preamble_ms;
00109     uint32_t d_TOW_at_current_symbol_ms;
00110
00111     bool d_flag_frame_sync;
00112     bool d_flag_preamble;
00113     bool d_sent_tlm_failed_msg;
00114     bool d_flag_PLL_180_deg_phase_locked;
00115     bool d_flag_TOW_set;
00116     bool d_dump;
00117     bool d_dump_mat;
00118     bool d_remove_dat;
00119     bool d_enable_navdata_monitor;
00120     bool d_dump_crc_stats;
00121 };
00122
00123
00124 /** \} */
00125 /** \} */
00126 #endif // GNSS_SDR_GPS_L1_CA_TELEMETRY_DECODER_GS_H

```

## 11.454 gps\_l2c\_telemetry\_decoder\_gs.h File Reference

Interface of a CNAV message demodulator block.

```

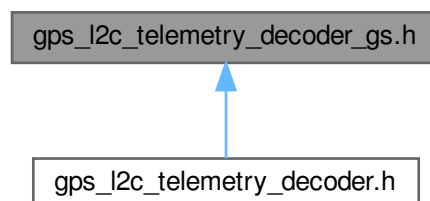
#include "gnss_block_interface.h"
#include "gnss_satellite.h"
#include "gps_cnav_navigation_message.h"
#include "nav_message_packet.h"
#include "tlm_conf.h"
#include "tlm_crc_stats.h"
#include <gnuradio/block.h>
#include <gnuradio/types.h>
#include <stdint>
#include <fstream>
#include <memory>
#include <string>
#include "cnav_msg.h"

```

Include dependency graph for gps\_l2c\_telemetry\_decoder\_gs.h:



This graph shows which files directly or indirectly include this file:



**Classes**

- class [gps\\_l2c\\_telemetry\\_decoder\\_gs](#)

*This class implements a block that decodes CNAV data defined in IS-GPS-200M.*

**Typedefs**

- using [gps\\_l2c\\_telemetry\\_decoder\\_gs\\_sptr](#) = gnss\_shared\_ptr<[gps\\_l2c\\_telemetry\\_decoder\\_gs](#)>

**Functions**

- [gps\\_l2c\\_telemetry\\_decoder\\_gs\\_sptr](#) [gps\\_l2c\\_make\\_telemetry\\_decoder\\_gs](#) (const [Gnss\\_Satellite](#) &satellite, const [Tlm\\_Conf](#) &conf)

**11.454.1 Detailed Description**

Interface of a CNAV message demodulator block.

**Author**

Javier Arribas, 2015. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gps\\_l2c\\_telemetry\\_decoder\\_gs.h](#).

---

**11.455 gps\_l2c\_telemetry\_decoder\_gs.h**

[Go to the documentation of this file.](#)

```

00001  /*!
00002   * \file gps_l2c_telemetry_decoder_gs.h
00003   * \brief Interface of a CNAV message demodulator block
00004   * \author Javier Arribas, 2015. jarribas(at)cttc.es
00005   *
00006   * -----
00007   *
00008   * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009   * This file is part of GNSS-SDR.
00010   *
00011   * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012   * SPDX-License-Identifier: GPL-3.0-or-later
00013   *
00014   * -----
00015   */
00016
00017 #ifndef GNSS_SDR_GPS_L2C_TELEMETRY_DECODER_GS_H
00018 #define GNSS_SDR_GPS_L2C_TELEMETRY_DECODER_GS_H
00019
00020
00021 #include "gnss_block_interface.h"
00022 #include "gnss_satellite.h"
00023 #include "gps_cnav_navigation_message.h"
00024 #include "nav_message_packet.h"
00025 #include "tlm_conf.h"
00026 #include "tlm_crc_stats.h"
00027 #include <gnuradio/block.h>
00028 #include <gnuradio/types.h> // for gr_vector_const_void_star
00029 #include <cstdint>
00030 #include <fstream>
00031 #include <memory> // for std::unique_ptr
00032 #include <string>
00033
00034 extern "C"
00035 {
00036     #include "cnav_msg.h"
00037 }
00038
00039 /** \addtogroup Telemetry_Decoder
00040  *  \{ */
00041 /** \addtogroup Telemetry_Decoder_gnuradio_blocks
00042  *  \{ */
00043
00044
00045 class gps_l2c_telemetry_decoder_gs;
00046

```

```

00047 using gps_l2c_telemetry_decoder_gs_sptr = gnss_shared_ptr<gps_l2c_telemetry_decoder_gs>;
00048
00049 gps_l2c_telemetry_decoder_gs_sptr gps_l2c_make_telemetry_decoder_gs(
00050     const Gnss_Satellite &satellite,
00051     const Tlm_Conf &conf);
00052
00053 /*!
00054  * \brief This class implements a block that decodes CNAV data defined in IS-GPS-200M
00055  */
00056 class gps_l2c_telemetry_decoder_gs : public gr::block
00057 {
00058 public:
00059     ~gps_l2c_telemetry_decoder_gs() override;
00060     void set_satellite(const Gnss_Satellite &satellite); //!< Set satellite PRN
00061     void set_channel(int32_t channel); //!< Set receiver's channel
00062     void reset();
00063
00064     /*!
00065     * \brief This is where all signal processing takes place
00066     */
00067     int general_work(int noutput_items, gr_vector_int &ninput_items,
00068         gr_vector_const_void_star &input_items, gr_vector_void_star &output_items) override;
00069 private:
00070     friend gps_l2c_telemetry_decoder_gs_sptr gps_l2c_make_telemetry_decoder_gs(
00071         const Gnss_Satellite &satellite,
00072         const Tlm_Conf &conf);
00073
00074     gps_l2c_telemetry_decoder_gs(const Gnss_Satellite &satellite, const Tlm_Conf &conf);
00075
00076     Gnss_Satellite d_satellite;
00077
00078     cnav_msg_decoder_t d_cnav_decoder{};
00079
00080     Gps_CNAV_Navigation_Message d_CNAV_Message;
00081
00082     Nav_Message_Packet d_nav_msg_packet;
00083     std::unique_ptr<Tlm_CRC_Stats> d_Tlm_CRC_Stats;
00084
00085     std::string d_dump_filename;
00086     std::ofstream d_dump_file;
00087
00088     double d_TOW_at_current_symbol;
00089     double d_TOW_at_Preamble;
00090
00091     uint64_t d_sample_counter;
00092     uint64_t d_last_valid_preamble;
00093
00094     int32_t d_channel;
00095
00096     uint32_t d_max_symbols_without_valid_frame;
00097
00098     bool d_dump;
00099     bool d_sent_tlm_failed_msg;
00100     bool d_flag_PLL_180_deg_phase_locked;
00101     bool d_flag_valid_word;
00102     bool d_dump_mat;
00103     bool d_remove_dat;
00104     bool d_enable_navdata_monitor;
00105     bool d_dump_crc_stats;
00106 };
00107
00108
00109
00110 /** \} */
00111 /** \} */
00112 #endif // GNSS_SDR_GPS_L2C_TELEMETRY_DECODER_GS_H

```

## 11.456 gps\_l5\_telemetry\_decoder\_gs.h File Reference

Interface of a CNAV message demodulator block.

```

#include "GPS_L5.h"
#include "gnss_block_interface.h"
#include "gnss_satellite.h"
#include "gps_cnav_navigation_message.h"
#include "nav_message_packet.h"
#include "tlm_conf.h"
#include "tlm_crc_stats.h"
#include <boost/circular_buffer.hpp>
#include <gnuradio/block.h>

```

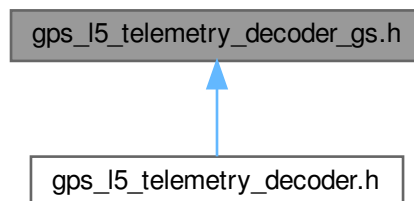


```
#include <gnuradio/types.h>
#include <stdint>
#include <fstream>
#include <memory>
#include <string>
#include "cnav_msg.h"
```

Include dependency graph for `gps_l5_telemetry_decoder_gs.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [gps\\_l5\\_telemetry\\_decoder\\_gs](#)  
*This class implements a GPS L5 Telemetry decoder.*

## Typedefs

- using [gps\\_l5\\_telemetry\\_decoder\\_gs\\_sptr](#) = `gnss_shared_ptr<gps\_l5\_telemetry\_decoder\_gs>`

## Functions

- `gps_l5_telemetry_decoder_gs_sptr gps\_l5\_make\_telemetry\_decoder\_gs (const Gnss\_Satellite &satellite, const Tlm\_Conf &conf)`

## 11.456.1 Detailed Description

Interface of a CNAV message demodulator block.

### 11.456.1.1 autotoc\_md453

#### Author

Antonio Ramos, 2017. antonio.ramos(at)cttc.es

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [gps\\_l5\\_telemetry\\_decoder\\_gs.h](#).

## 11.457 gps\_l5\_telemetry\_decoder\_gs.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file gps_l5_telemetry_decoder_gs.h
00003  * \brief Interface of a CNAV message demodulator block
00004  * \author Antonio Ramos, 2017. antonio.ramos(at)cttc.es
00005  * -----
00006  *
00007  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00008  * This file is part of GNSS-SDR.
00009  *
00010  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00011  * SPDX-License-Identifier: GPL-3.0-or-later
00012  *
00013  * -----
00014  */
00015
00016 #ifndef GNSS_SDR_GPS_L5_TELEMETRY_DECODER_GS_H
00017 #define GNSS_SDR_GPS_L5_TELEMETRY_DECODER_GS_H
00018
00019
00020 #include "GPS_L5.h" // for GPS_L5I_NH_CODE_LENGTH
00021 #include "gnss_block_interface.h"
00022 #include "gnss_satellite.h" // for Gnss_Satellite
00023 #include "gps_cnav_navigation_message.h" // for Gps_CNAV_Navigation_Message
00024 #include "nav_message_packet.h"
00025 #include "tlm_conf.h"
00026 #include "tlm_crc_stats.h"
00027 #include <boost/circular_buffer.hpp>
00028 #include <gnuradio/block.h>
00029 #include <gnuradio/types.h> // for gr_vector_const_void_star
00030 #include <stdint>
00031 #include <fstream>
00032 #include <memory> // for std::unique_ptr
00033 #include <string>
00034
00035 extern "C"
00036 {
00037 #include "cnav_msg.h"
00038 }
00039
00040 /** \addtogroup Telemetry_Decoder
00041  * \{ */
00042 /** \addtogroup Telemetry_Decoder_gnuradio_blocks
00043  * \{ */
00044
00045
00046 class gps_l5_telemetry_decoder_gs;
00047
00048 using gps_l5_telemetry_decoder_gs_sptr = gnss_shared_ptr<gps_l5_telemetry_decoder_gs>;
00049
00050 gps_l5_telemetry_decoder_gs_sptr gps_l5_make_telemetry_decoder_gs(
00051     const Gnss_Satellite &satellite,
00052     const Tlm_Conf &conf);
00053
00054 /*!
00055  * \brief This class implements a GPS L5 Telemetry decoder
00056  *
00057  */
00058 class gps_l5_telemetry_decoder_gs : public gr::block
00059 {
00060 public:
00061     ~gps_l5_telemetry_decoder_gs() override;
00062     void set_satellite(const Gnss_Satellite &satellite); //!< Set satellite PRN
00063     void set_channel(int32_t channel); //!< Set receiver's channel
00064     void reset();
00065     int general_work(int noutput_items, gr_vector_int &ninput_items,
00066         gr_vector_const_void_star &input_items, gr_vector_void_star &output_items) override;
00067 private:
00068     friend gps_l5_telemetry_decoder_gs_sptr gps_l5_make_telemetry_decoder_gs(
00069         const Gnss_Satellite &satellite,
00070         const Tlm_Conf &conf);
00071
00072     gps_l5_telemetry_decoder_gs(const Gnss_Satellite &satellite, const Tlm_Conf &conf);
00073
00074     cnav_msg_decoder_t d_cnav_decoder{};
00075
00076     Gnss_Satellite d_satellite;
00077
00078     Gps_CNAV_Navigation_Message d_CNAV_Message;
00079
00080     Nav_Message_Packet d_nav_msg_packet;
00081     std::unique_ptr<Tlm_CRC_Stats> d_Tlm_CRC_Stats;
00082
00083

```

```

00084     std::string d_dump_filename;
00085     std::ofstream d_dump_file;
00086
00087     uint64_t d_sample_counter;
00088     uint64_t d_last_valid_preamble;
00089
00090     int32_t d_channel;
00091
00092     uint32_t d_TOW_at_current_symbol_ms;
00093     uint32_t d_TOW_at_Preamble_ms;
00094     uint32_t d_max_symbols_without_valid_frame;
00095
00096     bool d_flag_PLL_180_deg_phase_locked;
00097     bool d_flag_valid_word;
00098     bool d_sent_tlm_failed_msg;
00099     bool d_dump;
00100     bool d_dump_mat;
00101     bool d_remove_dat;
00102     bool d_enable_navdata_monitor;
00103     bool d_dump_crc_stats;
00104 };
00105
00106
00107 /** \} */
00108 /** \} */
00109 #endif // GNSS_SDR_GPS_L5_TELEMETRY_DECODER_GS_H

```

## 11.458 sbas\_l1\_telemetry\_decoder\_gs.h File Reference

Interface of a SBAS telemetry data decoder block.

```

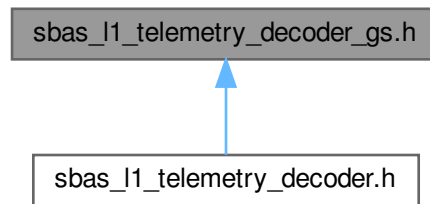
#include "gnss_block_interface.h"
#include "gnss_satellite.h"
#include <boost/crc.hpp>
#include <gnuradio/block.h>
#include <gnuradio/types.h>
#include <cstdint>
#include <cstdint>
#include <deque>
#include <fstream>
#include <memory>
#include <string>
#include <utility>
#include <vector>

```

Include dependency graph for sbas\_l1\_telemetry\_decoder\_gs.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [sbas\\_l1\\_telemetry\\_decoder\\_gs](#)

*This class implements a block that decodes the SBAS integrity and corrections data defined in RTCA MOPS DO-229.*

## Typedefs

- using [sbas\\_l1\\_telemetry\\_decoder\\_gs\\_ptr](#) = gnss\_shared\_ptr<[sbas\\_l1\\_telemetry\\_decoder\\_gs](#)>

## Functions

- [sbas\\_l1\\_telemetry\\_decoder\\_gs\\_ptr](#) **sbas\_l1\_make\_telemetry\_decoder\_gs** (const [Gnss\\_Satellite](#) &satellite, bool dump)

## 11.458.1 Detailed Description

Interface of a SBAS telemetry data decoder block.

### Author

Daniel Fehr 2013. [daniel.co\(at\)bluewin.ch](mailto:daniel.co(at)bluewin.ch)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [sbas\\_l1\\_telemetry\\_decoder\\_gs.h](#).

---

## 11.459 sbas\_l1\_telemetry\_decoder\_gs.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002   * \file sbas_l1_telemetry_decoder_gs.h
00003   * \brief Interface of a SBAS telemetry data decoder block
00004   * \author Daniel Fehr 2013. daniel.co\(at\)bluewin.ch
00005   *
00006   * -----
00007   *
00008   * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009   * This file is part of GNSS-SDR.
00010   *
00011   * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012   * SPDX-License-Identifier: GPL-3.0-or-later
00013   *
00014   * -----
00015   */
00016
00017 #ifndef GNSS_SDR_SBAS_L1_TELEMETRY_DECODER_GS_H
00018 #define GNSS_SDR_SBAS_L1_TELEMETRY_DECODER_GS_H
00019
00020 #include "gnss_block_interface.h"
  
```

```

00021 #include "gnss_satellite.h"
00022 #include <boost/crc.hpp> // for crc_optimal
00023 #include <gnuradio/block.h>
00024 #include <gnuradio/types.h> // for gr_vector_const_void_star
00025 #include <cstdlib> // for size_t
00026 #include <cstdint>
00027 #include <deque>
00028 #include <fstream>
00029 #include <memory> // for std::shared_ptr
00030 #include <string>
00031 #include <utility> // for pair
00032 #include <vector>
00033
00034 /** \addtogroup Telemetry_Decoder
00035  * \{ */
00036 /** \addtogroup Telemetry_Decoder_gnuradio_blocks
00037  * \{ */
00038
00039
00040 class Viterbi_Decoder_Sbas;
00041
00042 class sbas_l1_telemetry_decoder_gs;
00043
00044 using sbas_l1_telemetry_decoder_gs_sptr = gnss_shared_ptr<sbas_l1_telemetry_decoder_gs>;
00045
00046 sbas_l1_telemetry_decoder_gs_sptr sbas_l1_make_telemetry_decoder_gs(
00047     const Gnss_Satellite &satellite,
00048     bool dump);
00049
00050 /*!
00051  * \brief This class implements a block that decodes the SBAS integrity and
00052  * corrections data defined in RTCA MOPS DO-229
00053  */
00054 class sbas_l1_telemetry_decoder_gs : public gr::block
00055 {
00056 public:
00057     ~sbas_l1_telemetry_decoder_gs() override;
00058     void set_satellite(const Gnss_Satellite &satellite); //!< Set satellite PRN
00059     void set_channel(int32_t channel); //!< Set receiver's channel
00060     inline void reset() {};
00061
00062     /*!
00063     * \brief This is where all signal processing takes place
00064     */
00065     int general_work(int noutput_items, gr_vector_int &ninput_items,
00066         gr_vector_const_void_star &input_items, gr_vector_void_star &output_items) override;
00067 private:
00068     friend sbas_l1_telemetry_decoder_gs_sptr sbas_l1_make_telemetry_decoder_gs(
00069         const Gnss_Satellite &satellite,
00070         bool dump);
00071     sbas_l1_telemetry_decoder_gs(const Gnss_Satellite &satellite, bool dump);
00072
00073     void viterbi_decoder(double *page_part_symbols, int32_t *page_part_bits);
00074     void align_samples();
00075
00076     static const int32_t D_SAMPLES_PER_SYMBOL = 2;
00077     static const int32_t D_SYMBOLS_PER_BIT = 2;
00078     static const int32_t D_BLOCK_SIZE_IN_BITS = 30;
00079
00080     bool d_dump;
00081     Gnss_Satellite d_satellite;
00082     int32_t d_channel;
00083
00084     std::string d_dump_filename;
00085     std::ofstream d_dump_file;
00086
00087     size_t d_block_size; //!< number of samples which are processed during one
00088     invocation of the algorithms
00089     std::vector<double> d_sample_buf; //!< input buffer holding the samples to be processed in one
00090     block
00091
00092     typedef std::pair<int32_t, std::vector<int32_t>> msg_candidate_int_t;
00093     typedef std::pair<int32_t, std::vector<uint8_t>> msg_candidate_char_t;
00094
00095     // helper class for sample alignment
00096     class Sample_Aligner
00097     {
00098     public:
00099         Sample_Aligner();
00100         void reset();
00101         /*
00102         * samples length must be a multiple of two
00103         * for block operation
00104         */
00105         bool get_symbols(const std::vector<double> &samples, std::vector<double> &symbols);

```

```

00106
00107     private:
00108         int32_t d_n_smpls_in_history{3};
00109         double d_iir_par{0.05};
00110         double d_corr_paired{};
00111         double d_corr_shifted{};
00112         bool d_aligned{};
00113         double d_past_sample{};
00114     } d_sample_aligner;
00115
00116     // helper class for symbol alignment and Viterbi decoding
00117     class Symbol_Aligner_And_Decoder
00118     {
00119     public:
00120         Symbol_Aligner_And_Decoder();
00121         void reset();
00122         bool get_bits(const std::vector<double> &symbols, std::vector<int32_t> &bits);
00123
00124     private:
00125         int32_t d_KK{7};
00126         std::shared_ptr<Viterbi_Decoder_Sbas> d_vd1;
00127         std::shared_ptr<Viterbi_Decoder_Sbas> d_vd2;
00128         double d_past_symbol{0};
00129     } d_symbol_aligner_and_decoder;
00130
00131
00132     // helper class for detecting the preamble and collect the corresponding message candidates
00133     class Frame_Detector
00134     {
00135     public:
00136         void reset();
00137         void get_frame_candidates(const std::vector<int32_t> &bits, std::vector<std::pair<int32_t,
std::vector<int32_t>> &msg_candidates);
00138
00139     private:
00140         std::deque<int32_t> d_buffer;
00141     } d_frame_detector;
00142
00143
00144     // helper class for checking the CRC of the message candidates
00145     class Crc_Verifier
00146     {
00147     public:
00148         void reset();
00149         void get_valid_frames(const std::vector<msg_candidate_int_t> &msg_candidates,
std::vector<msg_candidate_char_t> &valid_msgs);
00150
00151     private:
00152         typedef boost::crc_optimal<24, 0x1864CFBU, 0x0, 0x0, false, false> crc_24_q_type;
00153         crc_24_q_type d_checksum_agent;
00154         void zerropad_front_and_convert_to_bytes(const std::vector<int32_t> &msg_candidate,
std::vector<uint8_t> &bytes);
00155         void zerropad_back_and_convert_to_bytes(const std::vector<int32_t> &msg_candidate,
std::vector<uint8_t> &bytes);
00156     } d_crc_verifier;
00157 };
00158
00159
00160 /** \} */
00161 /** \} */
00162 #endif // GNSS_SDR_SBAS_L1_TELEMETRY_DECODER_GS_H

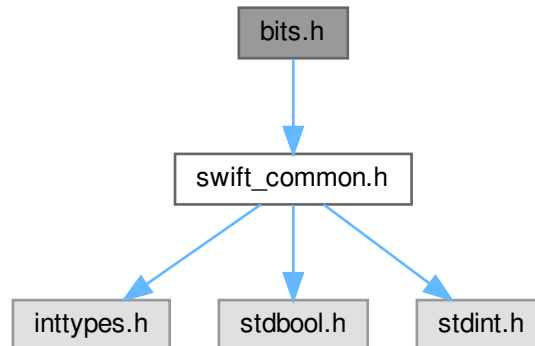
```

## 11.460 bits.h File Reference

Utilities for bit manipulation of the libswiftnav library.

```
#include "swift_common.h"
```

Include dependency graph for bits.h:



## Functions

- `uint8_t parity` (`uint32_t x`)
- `uint32_t getbitu` (`const uint8_t *buff`, `uint32_t pos`, `uint8_t len`)
- `int32_t getbits` (`const uint8_t *buff`, `uint32_t pos`, `uint8_t len`)
- `void setbitu` (`uint8_t *buff`, `uint32_t pos`, `uint32_t len`, `uint32_t data`)
- `void setbits` (`uint8_t *buff`, `uint32_t pos`, `uint32_t len`, `int32_t data`)
- `void bitcopy` (`void *dst`, `uint32_t dst_index`, `const void *src`, `uint32_t src_index`, `uint32_t count`)
- `void bitshl` (`void *buf`, `uint32_t size`, `uint32_t shift`)
- `uint8_t count_bits_u64` (`uint64_t v`, `uint8_t bv`)
- `uint8_t count_bits_u32` (`uint32_t v`, `uint8_t bv`)
- `uint8_t count_bits_u16` (`uint16_t v`, `uint8_t bv`)
- `uint8_t count_bits_u8` (`uint8_t v`, `uint8_t bv`)

### 11.460.1 Detailed Description

Utilities for bit manipulation of the libswiftnav library.

Author

Fergus Noble [fergus@swift-nav.com](mailto:fergus@swift-nav.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

This file was originally borrowed from libswiftnav <https://github.com/swift-nav/libswiftnav>, a portable C library implementing GNSS related functions and algorithms, and then modified by J. Arribas and C. Fernandez

Copyright (C) 2013, 2016 Swift Navigation Inc. Contact: Fergus Noble [fergus@swift-nav.com](mailto:fergus@swift-nav.com)

SPDX-License-Identifier: LGPL-3.0-only

Definition in file [bits.h](#).

## 11.461 bits.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file bits.h
00003  * \brief Utilities for bit manipulation of the libswiftnav library
00004  * \author Fergus Noble <fergus@swift-nav.com>

```

```

00005  *
00006  * -----
00007  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00008  * This file is part of GNSS-SDR.
00009  *
00010  * This file was originally borrowed from libswiftnav
00011  * <https://github.com/swift-nav/libswiftnav>,
00012  * a portable C library implementing GNSS related functions and algorithms,
00013  * and then modified by J. Arribas and C. Fernandez
00014  *
00015  * Copyright (C) 2013, 2016 Swift Navigation Inc.
00016  * Contact: Fergus Noble <fergus@swift-nav.com>
00017  *
00018  * SPDX-License-Identifier: LGPL-3.0-only
00019  *
00020  */
00021
00022 #ifndef GNSS_SDR_BITS_H
00023 #define GNSS_SDR_BITS_H
00024
00025 #include "swift_common.h"
00026
00027 /** \addtogroup Telemetry_Decoder
00028  * \{ */
00029 /** \addtogroup Telemetry_Decoder_libswiftcnv telemetry_decoder_libswiftcnv
00030  * Utilities for CNAV message decoding by Swift Navigation Inc.
00031  * \{ */
00032
00033
00034 uint8_t parity(uint32_t x);
00035 uint32_t getbitu(const uint8_t *buff, uint32_t pos, uint8_t len);
00036 int32_t getbits(const uint8_t *buff, uint32_t pos, uint8_t len);
00037 void setbitu(uint8_t *buff, uint32_t pos, uint32_t len, uint32_t data);
00038 void setbits(uint8_t *buff, uint32_t pos, uint32_t len, int32_t data);
00039 void bitcopy(void *dst, uint32_t dst_index,
00040             const void *src, uint32_t src_index, uint32_t count);
00041 void bitshl(void *buf, uint32_t size, uint32_t shift);
00042 uint8_t count_bits_u64(uint64_t v, uint8_t bv);
00043 uint8_t count_bits_u32(uint32_t v, uint8_t bv);
00044 uint8_t count_bits_u16(uint16_t v, uint8_t bv);
00045 uint8_t count_bits_u8(uint8_t v, uint8_t bv);
00046
00047
00048 /** \} */
00049 /** \} */
00050 #endif /* GNSS_SDR_BITS_H */

```

## 11.462 cnav\_msg.h File Reference

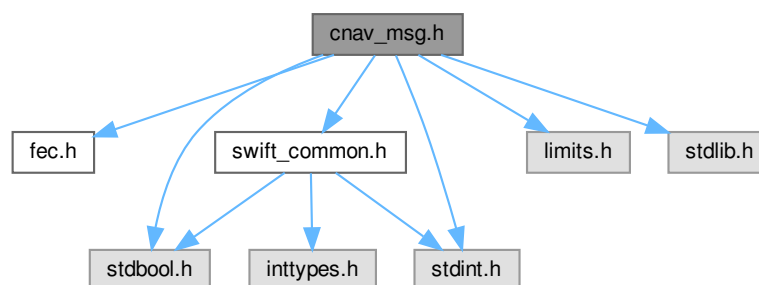
Utilities for CNAV message manipulation of the libswiftnav library.

```

#include "fec.h"
#include "swift_common.h"
#include <limits.h>
#include <stdbool.h>
#include <stdint.h>
#include <stdlib.h>

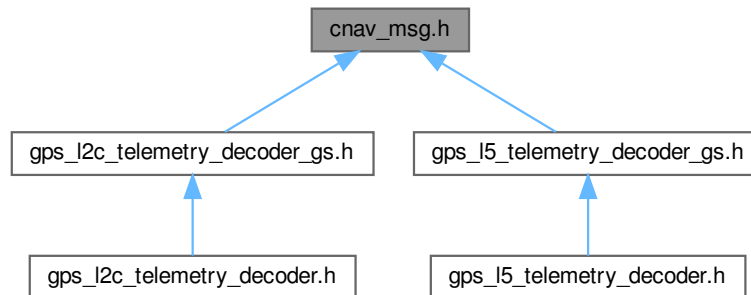
```

Include dependency graph for cnav\_msg.h:





This graph shows which files directly or indirectly include this file:



## Classes

- struct [cnav\\_msg\\_t](#)
- struct [cnav\\_v27\\_part\\_t](#)
- struct [cnav\\_msg\\_decoder\\_t](#)

## Macros

- #define [GPS\\_L2\\_V27\\_HISTORY\\_LENGTH\\_BITS](#) 64
- #define [GPS\\_L2C\\_V27\\_INIT\\_BITS](#) (32)
- #define [GPS\\_L2C\\_V27\\_DECODE\\_BITS](#) (32)
- #define [GPS\\_L2C\\_V27\\_DELAY\\_BITS](#) (32)

## Functions

- const [v27\\_poly\\_t](#) \* [cnav\\_msg\\_decoder\\_get\\_poly](#) (void)
- void [cnav\\_msg\\_decoder\\_init](#) ([cnav\\_msg\\_decoder\\_t](#) \*dec)
- bool [cnav\\_msg\\_decoder\\_add\\_symbol](#) ([cnav\\_msg\\_decoder\\_t](#) \*dec, unsigned char symbol, [cnav\\_msg\\_t](#) \*msg, [uint32\\_t](#) \*delay)

### 11.462.1 Detailed Description

Utilities for CNAV message manipulation of the libswiftnav library.

#### Author

Valeri Atamaniouk [valeri@swift-nav.com](mailto:valeri@swift-nav.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

This file was originally borrowed from libswiftnav <https://github.com/swift-nav/libswiftnav>, a portable C library implementing GNSS related functions and algorithms, and then modified by J. Arribas and C. Fernandez

Copyright (C) 2016 Swift Navigation Inc. Contact: Valeri Atamaniouk [valeri@swift-nav.com](mailto:valeri@swift-nav.com)

SPDX-License-Identifier: LGPL-3.0-only

Definition in file [cnav\\_msg.h](#).

## 11.463 cnav\_msg.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file cnav_msg.h
  
```

```

00003  * \brief Utilities for CNAV message manipulation of the libswiftnav library
00004  * \author Valeri Atamaniouk <valeri@swift-nav.com>
00005  *
00006  * -----
00007  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00008  * This file is part of GNSS-SDR.
00009  *
00010  * This file was originally borrowed from libswiftnav
00011  * <https://github.com/swift-nav/libswiftnav>,
00012  * a portable C library implementing GNSS related functions and algorithms,
00013  * and then modified by J. Arribas and C. Fernandez
00014  *
00015  * Copyright (C) 2016 Swift Navigation Inc.
00016  * Contact: Valeri Atamaniouk <valeri@swift-nav.com>
00017  *
00018  * SPDX-License-Identifier: LGPL-3.0-only
00019  *
00020  */
00021
00022
00023 #ifndef GNSS_SDR_CNAV_MSG_H
00024 #define GNSS_SDR_CNAV_MSG_H
00025
00026 #include "fec.h"
00027 #include "swift_common.h"
00028 #include <limits.h>
00029 #include <stdbool.h>
00030 #include <stdint.h>
00031 #include <stdlib.h>
00032
00033 /** \addtogroup Telemetry_Decoder
00034  * \{ */
00035 /** \addtogroup Telemetry_Decoder_libswiftnav
00036  * \{ */
00037
00038
00039 /** Size of the Viterbi decoder history. */
00040 #define GPS_L2_V27_HISTORY_LENGTH_BITS 64
00041 /** Bits to accumulate before decoding starts. */
00042 #define GPS_L2C_V27_INIT_BITS (32)
00043 /** Bits to decode at a time. */
00044 #define GPS_L2C_V27_DECODE_BITS (32)
00045 /** Bits in decoder tail. We ignore them. */
00046 #define GPS_L2C_V27_DELAY_BITS (32)
00047 /**
00048  * GPS CNAV message container.
00049  *
00050  * @sa cnav_msg_decoder_add_symbol
00051  */
00052 typedef struct
00053 {
00054     uint8_t prn; /**< SV PRN. 0..31 */
00055     uint8_t msg_id; /**< Message id. 0..31 */
00056     uint32_t tow; /**< GPS ToW in 6-second units.
00057     Multiply to 6 to get seconds. */
00058     bool alert; /**< CNAV message alert flag */
00059     uint8_t raw_msg[GPS_L2C_V27_DECODE_BITS + GPS_L2C_V27_DELAY_BITS]; /**< RAW MSG for GNSS-SDR */
00060 } cnav_msg_t;
00061 /**
00062  * GPS CNAV decoder component.
00063  * This component controls symbol decoding string.
00064  *
00065  * @sa cnav_msg_decoder_t
00066  */
00067 typedef struct
00068 {
00069     v27_t dec; /**< Viterbi block decoder object */
00070     v27_decision_t decisions[GPS_L2_V27_HISTORY_LENGTH_BITS];
00071     /**< Decision graph */
00072     unsigned char symbols[(GPS_L2C_V27_INIT_BITS + GPS_L2C_V27_DECODE_BITS) * 2];
00073     /**< Symbol buffer */
00074     size_t n_symbols; /**< Count of symbols in the symbol buffer */
00075     unsigned char decoded[GPS_L2C_V27_DECODE_BITS + GPS_L2C_V27_DELAY_BITS];
00076     /**< Decode buffer */
00077     size_t n_decoded; /**< Number of bits in the decode buffer */
00078     bool preamble_seen; /**< When true, the decode buffer is aligned on
00079     * preamble. */
00080     bool invert; /**< When true, indicates the bits are inverted */
00081     bool message_lock; /**< When true, indicates the message boundary
00082     * is found. */
00083     bool crc_ok; /**< Flag that the last message had good CRC */
00084     size_t n_crc_fail; /**< Counter for CRC failures */
00085     bool init; /**< Initial state flag. When true, initial bits
00086     * do not produce output. */
00087 } cnav_v27_part_t;
00088

```

```

00089 /**
00090  * GPS CNAV message lock and decoder object.
00091  *
00092  * Decoder uses two Viterbi decoder objects to ensure the lock is acquired when
00093  * the input symbol phase is not known.
00094  */
00095 typedef struct
00096 {
00097     cnav_v27_part_t part1; /**< Decoder for odd symbol pairs */
00098     cnav_v27_part_t part2; /**< Decoder for even symbol pairs */
00099 } cnav_msg_decoder_t;
00100
00101 const v27_poly_t *cnav_msg_decoder_get_poly(void);
00102 void cnav_msg_decoder_init(cnav_msg_decoder_t *dec);
00103 bool cnav_msg_decoder_add_symbol(cnav_msg_decoder_t *dec,
00104     unsigned char symbol,
00105     cnav_msg_t *msg,
00106     uint32_t *delay);
00107
00108 /** \} */
00109 /** \} */
00110 #endif /* GNSS_SDR_CNAV_MSG_H_ */

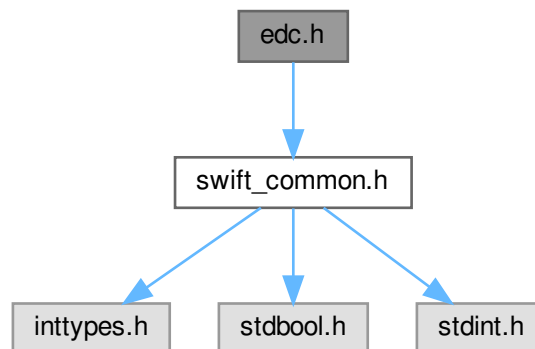
```

## 11.464 edc.h File Reference

Utilities for CRC computation of the libswiftnav library.

```
#include "swift_common.h"
```

Include dependency graph for edc.h:



### Functions

- uint32\_t **crc24q** (const uint8\_t \*buf, uint32\_t len, uint32\_t crc)
- uint32\_t **crc24q\_bits** (uint32\_t crc, const uint8\_t \*buf, uint32\_t n\_bits, bool invert)

### 11.464.1 Detailed Description

Utilities for CRC computation of the libswiftnav library.

#### Author

Fergus Noble [fergus@swift-nav.com](mailto:fergus@swift-nav.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

This file was originally borrowed from libswiftnav <https://github.com/swift-nav/libswiftnav>, a portable C library implementing GNSS related functions and algorithms, and then modified by J. Arribas and C. Fernandez

Copyright (C) 2010 Swift Navigation Inc. Contact: Fergus Noble [fergus@swift-nav.com](mailto:fergus@swift-nav.com)

SPDX-License-Identifier: LGPL-3.0-only  
 Definition in file [edc.h](#).

## 11.465 edc.h

[Go to the documentation of this file.](#)

```

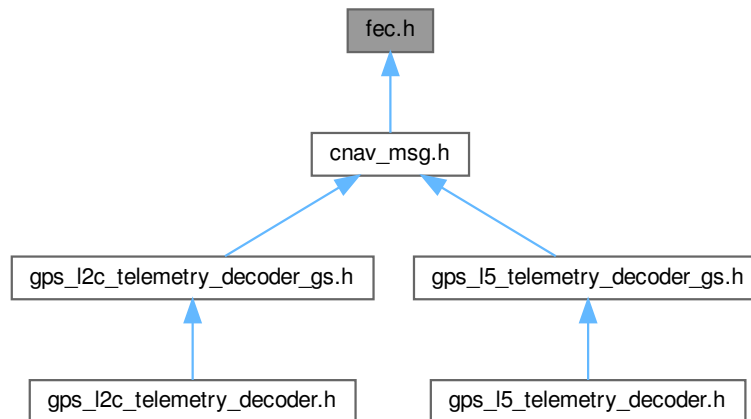
00001 /*!
00002  * \file edc.h
00003  * \brief Utilities for CRC computation of the libswiftnav library
00004  * \author Fergus Noble <fergus@swift-nav.com>
00005  *
00006  * -----
00007  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00008  * This file is part of GNSS-SDR.
00009  *
00010  * This file was originally borrowed from libswiftnav
00011  * <https://github.com/swift-nav/libswiftnav>,
00012  * a portable C library implementing GNSS related functions and algorithms,
00013  * and then modified by J. Arribas and C. Fernandez
00014  *
00015  * Copyright (C) 2010 Swift Navigation Inc.
00016  * Contact: Fergus Noble <fergus@swift-nav.com>
00017  *
00018  * SPDX-License-Identifier: LGPL-3.0-only
00019  *
00020  */
00021
00022
00023 #ifndef GNSS_SDR_EDC_H
00024 #define GNSS_SDR_EDC_H
00025
00026 #include "swift_common.h"
00027
00028 /** \addtogroup Telemetry_Decoder
00029  * \{ */
00030 /** \addtogroup Telemetry_Decoder_libswiftcnv
00031  * \{ */
00032
00033
00034 uint32_t crc24q(const uint8_t *buf, uint32_t len, uint32_t crc);
00035 uint32_t crc24q_bits(uint32_t crc, const uint8_t *buf, uint32_t n_bits, bool invert);
00036
00037
00038 /** \} */
00039 /** \} */
00040 #endif /* GNSS_SDR_EDC_H */

```

## 11.466 fec.h File Reference

Utilities for the convolutional encoder of the libswiftnav library.

This graph shows which files directly or indirectly include this file:



## Classes

- struct [v27\\_poly\\_t](#)
- struct [v27\\_decision\\_t](#)
- struct [v27\\_t](#)

## Macros

- #define [V27POLYA](#) 0x4f
- #define [V27POLYB](#) 0x6d

## Functions

- void **v27\_poly\_init** ([v27\\_poly\\_t](#) \*poly, const signed char polynomial[2])
- void **v27\_init** ([v27\\_t](#) \*v, [v27\\_decision\\_t](#) \*decisions, unsigned int decisions\_count, const [v27\\_poly\\_t](#) \*poly, unsigned char initial\_state)
- void **v27\_update** ([v27\\_t](#) \*v, const unsigned char \*syms, int nbits)
- void **v27\_chainback\_fixed** ([v27\\_t](#) \*v, unsigned char \*data, unsigned int nbits, unsigned char final\_state)
- void **v27\_chainback\_likely** ([v27\\_t](#) \*v, unsigned char \*data, unsigned int nbits)

## 11.466.1 Detailed Description

Utilities for the convolutional encoder of the libswiftnav library.

### Author

Phil Karn, KA9Q

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

This file was originally borrowed from libswiftnav <https://github.com/swift-nav/libswiftnav>, a portable C library implementing GNSS related functions and algorithms, and then modified by J. Arribas and C. Fernandez

Copyright (C) 2004, Phil Karn, KA9Q

SPDX-License-Identifier: LGPL-3.0-only

Definition in file [fec.h](#).

## 11.466.2 Macro Definition Documentation

### 11.466.2.1 V27POLYA

#define V27POLYA 0x4f

Definition at line 29 of file [fec.h](#).

### 11.466.2.2 V27POLYB

#define V27POLYB 0x6d

Definition at line 30 of file [fec.h](#).

## 11.467 fec.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file fec.h
00003  * \brief Utilities for the convolutional encoder of the libswiftnav library
00004  * \author Phil Karn, KA9Q
00005  *
00006  * -----
00007  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00008  * This file is part of GNSS-SDR.
00009  *
00010  * This file was originally borrowed from libswiftnav
00011  * <https://github.com/swift-nav/libswiftnav>,
00012  * a portable C library implementing GNSS related functions and algorithms,
00013  * and then modified by J. Arribas and C. Fernandez
00014  *
00015  * Copyright (C) 2004, Phil Karn, KA9Q
00016  *
00017  * SPDX-License-Identifier: LGPL-3.0-only
00018  *
00019  */
00020
00021
00022 #ifndef GNSS_SDR_FEC_H
00023 #define GNSS_SDR_FEC_H
00024
00025 /* r=1/2 k=7 convolutional encoder polynomials
00026  * The NASA-DSN convention is to use V27POLYA inverted, then V27POLYB
00027  * The CCSDS/NASA-GSFC convention is to use V27POLYB, then V27POLYA inverted
00028  */
00029 #define V27POLYA 0x4f
00030 #define V27POLYB 0x6d
00031
00032 typedef struct
00033 {
00034     unsigned char c0[32];
00035     unsigned char c1[32];
00036 } v27_poly_t;
00037
00038 typedef struct
00039 {
00040     unsigned int w[2];
00041 } v27_decision_t;
00042
00043 /* State info for instance of r=1/2 k=7 Viterbi decoder
00044  */
00045 typedef struct
00046 {
00047     unsigned int metrics1[64]; /* Path metric buffer 1 */
00048     unsigned int metrics2[64]; /* Path metric buffer 2 */
00049     /* Pointers to path metrics, swapped on every bit */
00050     unsigned int *old_metrics, *new_metrics;
00051     const v27_poly_t *poly; /* Polynomial to use */
00052     v27_decision_t *decisions; /* Beginning of decisions for block */
00053     unsigned int decisions_index; /* Index of current decision */
00054     unsigned int decisions_count; /* Number of decisions in history */
00055 } v27_t;
00056
00057 void v27_poly_init(v27_poly_t *poly, const signed char polynomial[2]);
00058
00059 void v27_init(v27_t *v, v27_decision_t *decisions, unsigned int decisions_count,
00060             const v27_poly_t *poly, unsigned char initial_state);
00061 void v27_update(v27_t *v, const unsigned char *syms, int nbits);
00062 void v27_chainback_fixed(v27_t *v, unsigned char *data, unsigned int nbits,
00063                        unsigned char final_state);
00064 void v27_chainback_likely(v27_t *v, unsigned char *data, unsigned int nbits);
00065

```

```
00066 #endif
```

## 11.468 swift\_common.h File Reference

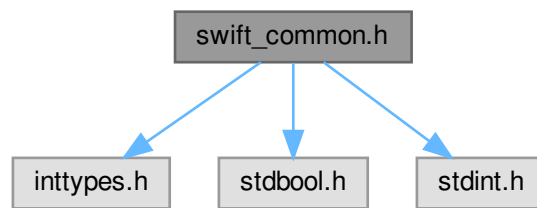
Common definitions used throughout the libswiftnav library.

```
#include <inttypes.h>
```

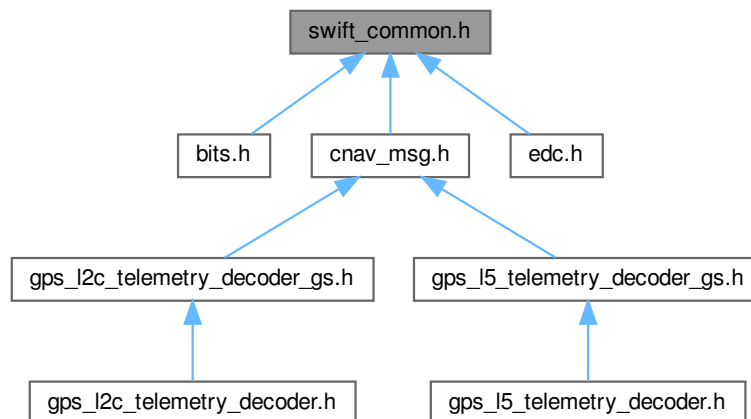
```
#include <stdbool.h>
```

```
#include <stdint.h>
```

Include dependency graph for swift\_common.h:



This graph shows which files directly or indirectly include this file:



### Macros

- #define [ABS](#)(x)
- #define [MIN](#)(x, y)
- #define [MAX](#)(x, y)
- #define [CLAMP\\_DIFF](#)(a, b)

### 11.468.1 Detailed Description

Common definitions used throughout the libswiftnav library.

**Author**

Henry Hallam [henry@swift-nav.com](mailto:henry@swift-nav.com) Fergus Noble [fergus@swift-nav.com](mailto:fergus@swift-nav.com)

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

This file was originally borrowed from libswiftnav <https://github.com/swift-nav/libswiftnav>, a portable C library implementing GNSS related functions and algorithms, and then modified by J. Arribas and C. Fernandez

Copyright (C) 2012 Swift Navigation Inc. Contact: Henry Hallam [henry@swift-nav.com](mailto:henry@swift-nav.com) Fergus Noble [fergus@swift-nav.com](mailto:fergus@swift-nav.com)

SPDX-License-Identifier: LGPL-3.0-only

Definition in file [swift\\_common.h](#).

**11.469 swift\_common.h**

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file swift_common.h
00003  * \brief Common definitions used throughout the libswiftnav library
00004  * \author Henry Hallam <henry@swift-nav.com>
00005  *         Fergus Noble <fergus@swift-nav.com>
00006  *
00007  * -----
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * This file was originally borrowed from libswiftnav
00012  * <https://github.com/swift-nav/libswiftnav>,
00013  * a portable C library implementing GNSS related functions and algorithms,
00014  * and then modified by J. Arribas and C. Fernandez
00015  *
00016  * Copyright (C) 2012 Swift Navigation Inc.
00017  * Contact: Henry Hallam <henry@swift-nav.com>
00018  *         Fergus Noble <fergus@swift-nav.com>
00019  *
00020  * SPDX-License-Identifier: LGPL-3.0-only
00021  *
00022  */
00023
00024
00025 #ifndef GNSS_SDR_SWIFT_COMMON_H
00026 #define GNSS_SDR_SWIFT_COMMON_H
00027
00028 /** \addtogroup Telemetry_Decoder
00029  * \{ */
00030 /** \addtogroup Telemetry_Decoder_libswiftcnv
00031  * \{ */
00032
00033
00034 #define ABS(x) ((x) < 0 ? -(x) : (x))
00035 #define MIN(x, y) (((x) < (y)) ? (x) : (y))
00036 #define MAX(x, y) (((x) > (y)) ? (x) : (y))
00037 #define CLAMP_DIFF(a, b) (MAX((a), (b)) - (b))
00038
00039 #include <inttypes.h>
00040 #include <stdbool.h>
00041 #include <stdint.h>
00042
00043
00044 /** \} */
00045 /** \} */
00046 #endif /* GNSS_SDR_SWIFT_COMMON_H */

```

**11.470 tlm\_conf.h File Reference**

Class that contains all the configuration parameters for generic telemetry decoder block.

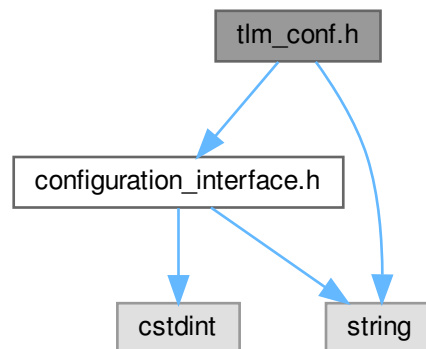
```

#include "configuration_interface.h"
#include <string>

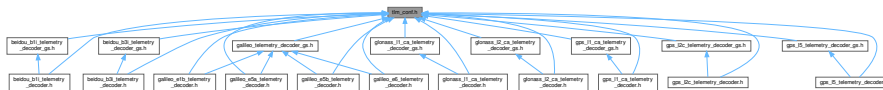
```



Include dependency graph for tlm\_conf.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Tlm\\_Conf](#)

## 11.470.1 Detailed Description

Class that contains all the configuration parameters for generic telemetry decoder block.

### Author

Carles Fernandez, 2020. cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [tlm\\_conf.h](#).

---

## 11.471 tlm\_conf.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file tlm_conf.h
00003  * \brief Class that contains all the configuration parameters for generic
00004  * telemetry decoder block.
00005  * \author Carles Fernandez, 2020. cfernandez(at)cttc.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
  
```

```

00017
00018 #ifndef GNSS_SDR_TLM_CONF_H
00019 #define GNSS_SDR_TLM_CONF_H
00020
00021 #include "configuration_interface.h"
00022 #include <string>
00023
00024 /** \addtogroup Telemetry_Decoder
00025  * \{ */
00026 /** \addtogroup Telemetry_Decoder_libs
00027  * \{ */
00028
00029
00030 class Tlm_Conf
00031 {
00032 public:
00033     Tlm_Conf() = default;
00034
00035     void SetFromConfiguration(const ConfigurationInterface *configuration, const std::string &role);
00036
00037     std::string dump_filename;
00038     std::string dump_crc_stats_filename;
00039     bool dump{false};
00040     bool dump_mat{false};
00041     bool remove_dat{false};
00042     bool enable_reed_solomon{false}; // for INAV message in Galileo E1B
00043     bool dump_crc_stats{false}; // telemetry CRC statistics
00044     bool enable_navdata_monitor{false};
00045     bool there_are_e1_channels{false};
00046     bool there_are_e6_channels{false};
00047     bool use_ced{false};
00048 };
00049
00050
00051 /** \} */
00052 /** \} */
00053 #endif // GNSS_SDR_TLM_CONF_H

```

## 11.472 tlm\_crc\_stats.h File Reference

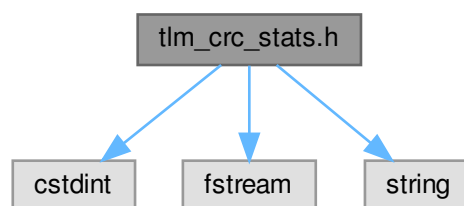
Class that computes the telemetry CRC statistics.

```

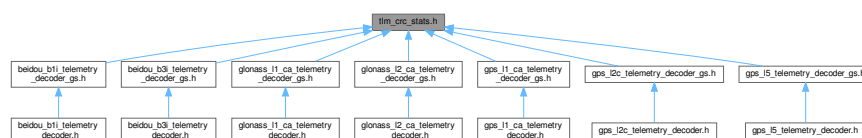
#include <cstdint>
#include <fstream>
#include <string>

```

Include dependency graph for tlm\_crc\_stats.h:



This graph shows which files directly or indirectly include this file:



**Classes**

- class [Tlm\\_CRC\\_Stats](#)

*Class that computes the telemetry CRC statistics.*

**11.472.1 Detailed Description**

Class that computes the telemetry CRC statistics.

**Author**

Marc Majoral, 2021. mmajoral(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [tlm\\_crc\\_stats.h](#).

---

**11.473 tlm\_crc\_stats.h**

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file tlm_crc_stats.h
00003  * \brief Class that computes the telemetry CRC statistics
00004  * \author Marc Majoral, 2021. mmajoral(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_CRC_STATS_H
00018 #define GNSS_SDR_CRC_STATS_H
00019
00020 #include <stdint>
00021 #include <fstream> // for std::ofstream
00022 #include <string> // for std::string
00023
00024 /** \addtogroup Telemetry_Decoder
00025  * \{ */
00026 /** \addtogroup Telemetry_Decoder_libs telemetry_decoder_libs
00027  * \{ */
00028
00029 /*!
00030 * \brief Class that computes the telemetry CRC statistics
00031 */
00032 class Tlm_CRC_Stats
00033 {
00034 public:
00035     Tlm_CRC_Stats() = default;
00036
00037     ~Tlm_CRC_Stats();
00038
00039     /*!
00040     * \brief Initialize the telemetry CRC statistics
00041     */
00042     void initialize(std::string dump_crc_stats_filename);
00043
00044     /*!
00045     * \brief Initialize the channel number and output file
00046     */
00047     bool set_channel(int32_t channel);
00048
00049     /*!
00050     * \brief Update the CRC statistics
00051     */
00052     void update_CRC_stats(bool CRC);
00053
00054 private:
00055     std::ofstream d_dump_file;
00056     std::string d_dump_crc_stats_filename;
00057     uint32_t d_num_crc_ok{0};
00058     uint32_t d_num_crc_not_ok{0};
00059     int32_t d_channel{0};

```

```

00060     bool d_enable_crc_stats{false};
00061 };
00062
00063
00064 /** \} */
00065 /** \} */
00066 #endif // GNSS_SDR_CRC_STATS_H

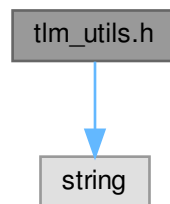
```

## 11.474 tlm\_utils.h File Reference

Utilities for the telemetry decoder blocks.

```
#include <string>
```

Include dependency graph for tlm\_utils.h:



### Functions

- int **save\_tlm\_matfile** (const std::string &dumpfile)
- bool **tlm\_remove\_file** (const std::string &file\_to\_remove)

### 11.474.1 Detailed Description

Utilities for the telemetry decoder blocks.

#### Author

Carles Fernandez, 2020. cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [tlm\\_utils.h](#).

---

## 11.475 tlm\_utils.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file tlm_utils.h
00003  * \brief Utilities for the telemetry decoder blocks.
00004  * \author Carles Fernandez, 2020. cfernandez(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_TLM_UTILS_H

```

```

00018 #define GNSS_SDR_TLM_UTILS_H
00019
00020 #include <string>
00021
00022 /** \addtogroup Telemetry_Decoder
00023  * \{ */
00024 /** \addtogroup Telemetry_Decoder_libs
00025  * \{ */
00026
00027 int save_tlm_matfile(const std::string &dumpfile);
00028
00029 bool tlm_remove_file(const std::string &file_to_remove);
00030
00031 /** \} */
00032 /** \} */
00033 #endif // GNSS_SDR_TLM_UTILS_H

```

## 11.476 viterbi\_decoder.h File Reference

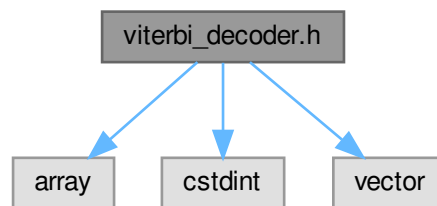
Class that implements a Viterbi decoder.

```
#include <array>
```

```
#include <cstdint>
```

```
#include <vector>
```

Include dependency graph for viterbi\_decoder.h:



### Classes

- class [Viterbi\\_Decoder](#)

*Class that implements a Viterbi decoder.*

### 11.476.1 Detailed Description

Class that implements a Viterbi decoder.

#### Author

Carles Fernandez, 2021. cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [viterbi\\_decoder.h](#).

---

## 11.477 viterbi\_decoder.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file viterbi_decoder.h
00003  * \brief Class that implements a Viterbi decoder
00004  * \author Carles Fernandez, 2021. cfernandez(at)cttc.es
00005  *

```

```

00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_VITERBI_DECODER_H
00018 #define GNSS_SDR_VITERBI_DECODER_H
00019
00020 #include <array>
00021 #include <stdint>
00022 #include <vector>
00023
00024 /** \addtogroup Telemetry_Decoder
00025  * \{ */
00026 /** \addtogroup Telemetry_Decoder_libs
00027  * Utilities for the decoding of GNSS navigation messages.
00028  * \{ */
00029
00030
00031 /*!
00032  * \brief Class that implements a Viterbi decoder
00033  */
00034 class Viterbi_Decoder
00035 {
00036 public:
00037     /*!
00038      * \brief Constructor of a Viterbi decoder
00039      * \param[in] KK Constraint length
00040      * \param[in] nn Coding rate 1/n
00041      * \param[in] LL Data length
00042      * \param[in] g Polynomial G1 and G2
00043      */
00044     Viterbi_Decoder(int32_t KK, int32_t nn, int32_t LL, const std::array<int32_t, 2>& g);
00045
00046     /*!
00047      * \brief Uses the Viterbi algorithm to perform hard-decision decoding of a convolutional code.
00048      * \param[out] output_u_int Hard decisions on the data bits
00049      * \param[in] input_c The received signal in LLR-form. For BPSK, must be in form  $r = 2ay/(\sigma^2)$ .
00050      */
00051     /*
00052      void decode(std::vector<int32_t>& output_u_int, const std::vector<float>& input_c);
00053
00054     /*!
00055      * \brief Reset internal status
00056      */
00057     void reset();
00058
00059 private:
00060     /*
00061      * Function that creates the transit and output vectors
00062      */
00063     void nsc_transit(std::vector<int32_t>& output_p,
00064                     std::vector<int32_t>& trans_p,
00065                     int32_t input) const;
00066
00067     /*
00068      * Computes the branch metric used for decoding.
00069      * \return (returned float) The metric between the hypothetical symbol and the received vector
00070      * \param[in] symbol The hypothetical symbol
00071      */
00072     float Gamma(int32_t symbol) const;
00073
00074
00075     /*
00076      * Determines if a symbol has odd (1) or even (0) parity
00077      * Output parameters:
00078      * \return (returned int): The symbol's parity = 1 for odd and 0 for even
00079      *
00080      * \param[in] symbol The integer-valued symbol
00081      * \param[in] length The highest bit position in the symbol
00082      *
00083      * This function is used by nsc_enc_bit()
00084      */
00085     int32_t parity_counter(int32_t symbol, int32_t length) const;
00086
00087     /*
00088      * Convolutionally encodes a single bit using a rate 1/n encoder.
00089      * Takes in one input bit at a time, and produces a n-bit output.
00090      *
00091      * \return (returned int): Computed output

```

```

00092      *
00093      * \param[in] input      The input data bit (i.e. a 0 or 1).
00094      * \param[in] state_in  The starting state of the encoder (an int from 0 to 2^m-1).
00095      * \param[out] state_out_p[] An integer containing the final state of the encoder
00096      *                          (i.e. the state after encoding this bit)
00097      *
00098      * This function is used by nsc_transit()
00099      */
00100      int32_t nsc_enc_bit(int32_t* state_out_p,
00101                          int32_t input,
00102                          int32_t state_in) const;
00103
00104      std::vector<float> d_prev_section{};
00105      std::vector<float> d_next_section{};
00106
00107      std::vector<float> d_rec_array{};
00108      std::vector<float> d_metric_c{};
00109      std::vector<int32_t> d_prev_bit{};
00110      std::vector<int32_t> d_prev_state{};
00111      std::array<int32_t, 2> d_g{};
00112
00113      std::vector<int32_t> d_out0;
00114      std::vector<int32_t> d_out1;
00115      std::vector<int32_t> d_state0;
00116      std::vector<int32_t> d_state1;
00117
00118      float d_MAXLOG = 1e7; // Define infinity
00119      int32_t d_KK{};
00120      int32_t d_nn{};
00121      int32_t d_LL{};
00122
00123      int32_t d_mm{};
00124      int32_t d_states{};
00125      int32_t d_number_symbols{};
00126 };
00127
00128 /** \} */
00129 /** \} */
00130 #endif // GNSS_SDR_VITERBI_DECODER_H

```

## 11.478 viterbi\_decoder\_sbass.h File Reference

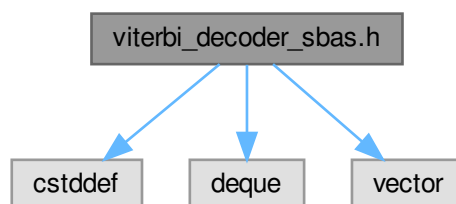
Interface of a Viterbi decoder class based on the Iterative Solutions Coded Modulation Library by Matthew C. Valenti.

```
#include <cstdint>
```

```
#include <deque>
```

```
#include <vector>
```

Include dependency graph for viterbi\_decoder\_sbass.h:



### Classes

- class [Viterbi\\_Decoder\\_Sbas](#)

*Class that implements a Viterbi decoder.*

### 11.478.1 Detailed Description

Interface of a Viterbi decoder class based on the Iterative Solutions Coded Modulation Library by Matthew C. Valenti.

## Author

Daniel Fehr 2013. daniel.co(at)bluewin.ch

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [viterbi\\_decoder\\_sbass.h](#).

---

## 11.479 viterbi\_decoder\_sbass.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file viterbi_decoder_sbass.h
00003  * \brief Interface of a Viterbi decoder class based on the Iterative Solutions
00004  * Coded Modulation Library by Matthew C. Valenti
00005  * \author Daniel Fehr 2013. daniel.co(at)bluewin.ch
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_VITERBI_DECODER_SBASS_H
00019 #define GNSS_SDR_VITERBI_DECODER_SBASS_H
00020
00021 #include <cstdlib> // for size_t
00022 #include <deque>
00023 #include <vector>
00024
00025 /** \addtogroup Telemetry_Decoder
00026  * \{ */
00027 /** \addtogroup Telemetry_Decoder_libs telemetry_decoder_libs
00028  * \{ */
00029
00030
00031 /*!
00032  * \brief Class that implements a Viterbi decoder
00033  */
00034 class Viterbi_Decoder_Sbas
00035 {
00036 public:
00037     Viterbi_Decoder_Sbas(const int g_encoder[], int KK, int nn);
00038
00039     void reset();
00040
00041     /*!
00042     * \brief Uses the Viterbi algorithm to perform hard-decision decoding of a convolutional code.
00043     *
00044     * \param[in] input_c[] The received signal in LLR-form. For BPSK, must be in form r =
00045     2*a*y/(sigma^2).
00046     * \param[in] LL The number of data bits to be decoded (does not include the mm
00047     zero-tail-bits)
00048     *
00049     * \return output_u_int[] Hard decisions on the data bits (without the mm zero-tail-bits)
00050     */
00049     float decode_block(const double input_c[], int* output_u_int, int LL);
00050
00051     float decode_continuous(const double sym[], int traceback_depth, int bits[],
00052         int nbits_requested, int& nbits_decoded);
00053
00054 private:
00055     class Prev
00056     {
00057     public:
00058         int num_states;
00059         Prev(int states, int tt);
00060         Prev(const Prev& prev);
00061         Prev& operator=(const Prev& other);
00062         ~Prev();
00063
00064         int get_ancestor_state_of_current_state(int current_state) const;
00065         int get_bit_of_current_state(int current_state) const;
00066         float get_metric_of_current_state(int current_state) const;
00067         int get_t() const;
00068         void set_current_state_as_ancestor_of_next_state(int next_state, int current_state);
00069         void set_decoded_bit_for_next_state(int next_state, int bit);
00070         void set_survivor_branch_metric_of_next_state(int next_state, float metric);

```



```

00071
00072     private:
00073         std::vector<float> v_metric;
00074         std::vector<int> state;
00075         std::vector<int> v_bit;
00076         int t;
00077         int refcount;
00078     };
00079
00080     // operations on the trellis (change decoder state)
00081     void init_trellis_state();
00082     int do_acs(const double sym[], int nbits);
00083     int do_traceback(std::size_t traceback_length);
00084     int do_tb_and_decode(int traceback_length, int requested_decoding_length, int state, int
output_u_int[], float& indicator_metric);
00085
00086     // branch metric function
00087     float gamma(const float rec_array[], int symbol, int nn);
00088
00089     // trellis generation
00090     void nsc_transit(int output_p[], int trans_p[], int input, const int g[], int KK, int nn);
00091     int nsc_enc_bit(int state_out_p[], int input, int state_in, const int g[], int KK, int nn);
00092     int parity_counter(int symbol, int length);
00093
00094     // trellis state
00095     std::deque<Prev> d_trellis_paths;
00096     std::vector<float> d_pm_t;
00097     std::vector<float> d_metric_c; /* Set of all possible branch metrics */
00098     std::vector<float> d_rec_array; /* Received values for one trellis section */
00099
00100     // trellis definition
00101     std::vector<int> d_out0;
00102     std::vector<int> d_state0;
00103     std::vector<int> d_out1;
00104     std::vector<int> d_statel;
00105
00106     // measures
00107     float d_indicator_metric;
00108
00109     // code properties
00110     int d_KK;
00111     int d_nn;
00112
00113     // derived code properties
00114     int d_mm;
00115     int d_states;
00116     int d_number_symbols;
00117     bool d_trellis_state_is_initialised;
00118 };
00119
00120
00121 /** \} */
00122 /** \} */
00123 #endif // GNSS_SDR_VITERBI_DECODER_SBAS_H

```

## 11.480 beidou b1i dll pll tracking.h File Reference

Interface of an adapter of a DLL+PLL tracking loop block for Beidou B1I to a [TrackingInterface](#).

```
#include "dll_pll_veml_tracking.h"
```

```
#include "tracking_interface.h"
```

```
#include <string>
```

Include dependency graph for beidou b1i dll pll tracking.h:



## Classes

- class BeidouB1iDIIPITracking

*This class implements a code DLL + carrier PLL tracking loop.*

## 11.480.1 Detailed Description

Interface of an adapter of a DLL+PLL tracking loop block for Beidou B1I to a [TrackingInterface](#).

### Author

Sergi Segura, 2018. [sergi.segura.munoz\(at\)gmail.com](mailto:sergi.segura.munoz(at)gmail.com)

Code DLL + carrier PLL according to the algorithms described in: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen, A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach, Birkhauser, 2007  
GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [beidou\\_b1i\\_dll\\_pll\\_tracking.h](#).

## 11.481 beidou\_b1i\_dll\_pll\_tracking.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file beidou_b1i_dll_pll_tracking.h
00003  * \brief Interface of an adapter of a DLL+PLL tracking loop block
00004  * for Beidou B1I to a TrackingInterface
00005  * \author Sergi Segura, 2018. sergi.segura.munoz\(at\)gmail.com
00006  *
00007  * Code DLL + carrier PLL according to the algorithms described in:
00008  * K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen,
00009  * A Software-Defined GPS and Galileo Receiver. A Single-Frequency
00010  * Approach, Birkhauser, 2007
00011  *
00012  * -----
00013  *
00014  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00015  * This file is part of GNSS-SDR.
00016  *
00017  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00018  * SPDX-License-Identifier: GPL-3.0-or-later
00019  *
00020  * -----
00021  */
00022
00023 #ifndef GNSS_SDR_BEIDOU_B1I_DLL_PLL_TRACKING_H
00024 #define GNSS_SDR_BEIDOU_B1I_DLL_PLL_TRACKING_H
00025
00026 #include "dll_pll_veml_tracking.h"
00027 #include "tracking_interface.h"
00028 #include <string>
00029
00030 /** \addtogroup Tracking
00031  * \{ */
00032 /** \addtogroup Tracking_adapters
00033  * \{ */
00034
00035
00036 class ConfigurationInterface;
00037
00038 /*!
00039 * \brief This class implements a code DLL + carrier PLL tracking loop
00040 */
00041 class BeidouB1iDllPllTracking : public TrackingInterface
00042 {
00043 public:
00044     BeidouB1iDllPllTracking(
00045         const ConfigurationInterface* configuration,
00046         const std::string& role,
00047         unsigned int in_streams,
00048         unsigned int out_streams);
00049
00050     ~BeidouB1iDllPllTracking() = default;
00051
00052     inline std::string role() override
00053     {
00054         return role_;
00055     }
00056
00057     inline std::string implementation() override
00058     {
00059         return "BEIDOU_B1I_DLL_PLL_Tracking";
00060     }
00061
00062     inline size_t item_size() override
00063     {
00064         return item_size_;
00065     }

```

```

00065     }
00066
00067     void connect(gr::top_block_sptr top_block) override;
00068     void disconnect(gr::top_block_sptr top_block) override;
00069     gr::basic_block_sptr get_left_block() override;
00070     gr::basic_block_sptr get_right_block() override;
00071
00072     /*!
00073      * \brief Set tracking channel unique ID
00074      */
00075     void set_channel(unsigned int channel) override;
00076
00077     /*!
00078      * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00079      * to efficiently exchange synchronization data between acquisition and tracking blocks
00080      */
00081     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00082
00083     void start_tracking() override;
00084
00085     /*!
00086      * \brief Stop running tracking
00087      */
00088     void stop_tracking() override;
00089
00090 private:
00091     dll_pll_veml_tracking_sptr tracking_sptr_;
00092     std::string role_;
00093     size_t item_size_;
00094     unsigned int channel_;
00095     unsigned int in_streams_;
00096     unsigned int out_streams_;
00097 };
00098
00099
00100 /** \} */
00101 /** \} */
00102 #endif // GNSS_SDR_BEIDOU_B1I_DLL_PLL_TRACKING_H

```

## 11.482 beidou\_b3i\_dll\_pll\_tracking.h File Reference

Interface of an adapter of a DLL+PLL tracking loop block for Beidou B3I to a [TrackingInterface](#).

```
#include "dll_pll_veml_tracking.h"
```

```
#include "tracking interface.h"
```

```
#include <string>
```

Include dependency graph for beidou b3i dll pll tracking.h:



## Classes

- class BeidouB3iDIIPIITracking

*This class implements a code DLL + carrier PLL tracking loop.*

### 11.482.1 Detailed Description

Interface of an adapter of a DLL+PLL tracking loop block for Beidou B3I to a [TrackingInterface](#).

**Author**

Damian Miralles, 2019. dmiralles2009(at)gmail.com

Code DLL + carrier PLL according to the algorithms described in: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen. A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach. Birkhauser, 2007

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [beidou\\_b3i\\_dll\\_pll\\_tracking.h](#).

## 11.483 beidou\_b3i\_dll\_pll\_tracking.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file beidou_b3i_dll_pll_tracking.h
00003  * \brief Interface of an adapter of a DLL+PLL tracking loop block
00004  * for Beidou B3I to a TrackingInterface
00005  * \author Damian Miralles, 2019. dmiralles2009(at)gmail.com
00006  *
00007  * Code DLL + carrier PLL according to the algorithms described in:
00008  * K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen,
00009  * A Software-Defined GPS and Galileo Receiver. A Single-Frequency
00010  * Approach, Birkhauser, 2007
00011  *
00012  * -----
00013  *
00014  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00015  * This file is part of GNSS-SDR.
00016  *
00017  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00018  * SPDX-License-Identifier: GPL-3.0-or-later
00019  *
00020  * -----
00021  */
00022
00023 #ifndef GNSS_SDR_BEIDOU_B3I_DLL_PLL_TRACKING_H
00024 #define GNSS_SDR_BEIDOU_B3I_DLL_PLL_TRACKING_H
00025
00026 #include "dll_pll_veml_tracking.h"
00027 #include "tracking_interface.h"
00028 #include <string>
00029
00030 /** \addtogroup Tracking
00031  * \{ */
00032 /** \addtogroup Tracking_adapters
00033  * \{ */
00034
00035
00036 class ConfigurationInterface;
00037
00038 /*!
00039 * \brief This class implements a code DLL + carrier PLL tracking loop
00040 */
00041 class BeidouB3iDllPllTracking : public TrackingInterface
00042 {
00043 public:
00044     BeidouB3iDllPllTracking(
00045         const ConfigurationInterface* configuration,
00046         const std::string& role,
00047         unsigned int in_streams,
00048         unsigned int out_streams);
00049
00050     ~BeidouB3iDllPllTracking() = default;
00051
00052     inline std::string role() override
00053     {
00054         return role_;
00055     }
00056
00057     inline std::string implementation() override
00058     {
00059         return "BEIDOU_B3I_DLL_PLL_Tracking";
00060     }
00061
00062     inline size_t item_size() override
00063     {
00064         return item_size_;
00065     }
00066
00067     void connect(gr::top_block_sptr top_block) override;
00068     void disconnect(gr::top_block_sptr top_block) override;
00069     gr::basic_block_sptr get_left_block() override;
00070     gr::basic_block_sptr get_right_block() override;
00071
00072     /*!
00073     * \brief Set tracking channel unique ID
00074     */
00075     void set_channel(unsigned int channel) override;
00076
00077     /*!
00078     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00079     * to efficiently exchange synchronization data between acquisition and tracking blocks
00080     */
00081     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00082
00083     void start_tracking() override;

```

```
00084
00085      /*!
00086       * \brief Stop running tracking
00087       */
00088      void stop_tracking() override;
00089
00090  private:
00091      dll_pll_veml_tracking_sp_ptr tracking_sp_ptr_;
00092      std::string role_;
00093      size_t item_size_;
00094      unsigned int channel_;
00095      unsigned int in_streams_;
00096      unsigned int out_streams_;
00097  };
00098
00099
00100  /** \} */
00101  /** \} */
00102  #endif // GNSS_SDR_BEIDOU_B3I_DLL_PLL_TRACKING_H
```

## 11.484 galileo\_e1\_dll\_pll\_veml\_tracking.h File Reference

Adapts a DLL+PLL VEML (Very Early Minus Late) tracking loop block to a [TrackingInterface](#) for Galileo E1 signals.

```
#include "dll_pll_veml_tracking.h"
```

```
#include "tracking_interface.h"
```

```
#include <string>
```

Include dependency graph for galileo\_e1\_dll\_pll\_veml\_tracking.h:



## Classes

- class `GalileoE1DIIPIIVemlTracking`

This class Adapts a DLL+PLL VEML (Very Early Minus Late) tracking loop block to a [TrackingInterface](#) for Galileo E1 signals.

### 11.484.1 Detailed Description

Adapts a DLL+PLL VEML (Very Early Minus Late) tracking loop block to a [TrackingInterface](#) for Galileo E1 signals.

**Author**

Luis Esteve, 2012. luis(at)epsilon-formacion.com

Code DLL + carrier PLL according to the algorithms described in: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen, A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach, Birkhauser, 2007  
GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [galileo\\_e1\\_dll\\_pll\\_veml\\_tracking.h](#).

## 11.485 galileo e1\_dll\_pll\_veml\_tracking.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file galileo_e1_dll_pll_veml_tracking.h
00003  * \brief Adapts a DLL+PLL VEML (Very Early Minus Late) tracking loop block
00004  *       to a TrackingInterface for Galileo E1 signals
00005  * \author Luis Esteve, 2012. luis(at)epsilon-formacion.com
00006  *
00007  * Code DLL + carrier PLL according to the algorithms described in:
00008  * K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen,
00009  * A Software-Defined GPS and Galileo Receiver. A Single-Frequency
```

```

00010  * Approach, Birkhauser, 2007
00011  *
00012  * -----
00013  *
00014  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00015  * This file is part of GNSS-SDR.
00016  *
00017  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00018  * SPDX-License-Identifier: GPL-3.0-or-later
00019  *
00020  * -----
00021  */
00022
00023 #ifndef GNSS_SDR_GALILEO_E1_DLL_PLL_VEML_TRACKING_H
00024 #define GNSS_SDR_GALILEO_E1_DLL_PLL_VEML_TRACKING_H
00025
00026 #include "dll_pll_veml_tracking.h"
00027 #include "tracking_interface.h"
00028 #include <string>
00029
00030 /** \addtogroup Tracking
00031  * \{ */
00032 /** \addtogroup Tracking_adapters
00033  * \{ */
00034
00035
00036 class ConfigurationInterface;
00037
00038 /*!
00039  * \brief This class Adapts a DLL+PLL VEML (Very Early Minus Late) tracking
00040  * loop block to a TrackingInterface for Galileo E1 signals
00041  */
00042 class GalileoE1DllPllVemlTracking : public TrackingInterface
00043 {
00044 public:
00045     GalileoE1DllPllVemlTracking(
00046         const ConfigurationInterface* configuration,
00047         const std::string& role,
00048         unsigned int in_streams,
00049         unsigned int out_streams);
00050
00051     ~GalileoE1DllPllVemlTracking() = default;
00052
00053     inline std::string role() override
00054     {
00055         return role_;
00056     }
00057
00058     /*! Returns "Galileo_E1_DLL_PLL_VEML_Tracking"
00059     inline std::string implementation() override
00060     {
00061         return "Galileo_E1_DLL_PLL_VEML_Tracking";
00062     }
00063
00064     inline size_t item_size() override
00065     {
00066         return item_size_;
00067     }
00068
00069     void connect(gr::top_block_sptr top_block) override;
00070     void disconnect(gr::top_block_sptr top_block) override;
00071     gr::basic_block_sptr get_left_block() override;
00072     gr::basic_block_sptr get_right_block() override;
00073
00074     /*!
00075     * \brief Set tracking channel unique ID
00076     */
00077     void set_channel(unsigned int channel) override;
00078
00079     /*!
00080     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00081     * to efficiently exchange synchronization data between acquisition and
00082     * tracking blocks
00083     */
00084     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00085
00086     void start_tracking() override;
00087
00088     /*!
00089     * \brief Stop running tracking
00090     */
00091     void stop_tracking() override;
00092
00093 private:
00094     dll_pll_veml_tracking_sptr tracking_sptr_;
00095     std::string role_;
00096     size_t item_size_;

```

```
00097     unsigned int channel_;
00098     unsigned int in_streams_;
00099     unsigned int out_streams_;
00100 };
00101
00102
00103 /** \} */
00104 /** \} */
00105 #endif // GNSS_SDR_GALILEO_E1_DLL_PLL_VEMU_TRACKING_H
```

## 11.486 galileo\_e1\_dll\_pll\_veml\_tracking\_fpga.h File Reference

Adapts a DLL+PLL VEML (Very Early Minus Late) tracking loop block to a [TrackingInterface](#) for Galileo E1 signals for the FPGA.

```
#include "dll_pll_veml_tracking_fpga.h"
#include "tracking_interface.h"
#include <string>
```

Include dependency graph for galileo\_e1\_dll\_pll\_veml\_tracking\_fpga.h:



## Classes

- class `GalileoE1DIIPIVemlTrackingFpga`

This class Adapts a DLL+PLL VEML (Very Early Minus Late) tracking loop block to a [TrackingInterface](#) for Galileo E1 signals.

### 11.486.1 Detailed Description

Adapts a DLL+PLL VEML (Very Early Minus Late) tracking loop block to a [TrackingInterface](#) for Galileo E1 signals for the FPGA.

**Author**

Marc Majoral, 2019. mmajoral(at)cttc.cat

Code DLL + carrier PLL according to the algorithms described in: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen, A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach, Birkhauser, 2007  
GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [galileo\\_e1\\_dll\\_pll\\_veml\\_tracking\\_fpga.h](#).

## 11.487 galileo\_e1\_dll\_pll\_veml\_tracking\_fpga.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file galileo_e1_dll_pll_veml_tracking_fpga.h
00003  * \brief Adapts a DLL+PLL VEML (Very Early Minus Late) tracking loop block
00004  * to a TrackingInterface for Galileo E1 signals for the FPGA
00005  * \author Marc Majoral, 2019. mmajoral(at)cttc.cat
00006  *
00007  * Code DLL + carrier PLL according to the algorithms described in:
00008  * K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen,
00009  * A Software-Defined GPS and Galileo Receiver. A Single-Frequency
00010  * Approach, Birkhauser, 2007
00011  *
00012  * -----
00013  *
00014  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00015  * This file is part of GNSS-SDR.
00016  *
00017  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
```

```

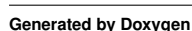
00018  * SPDX-License-Identifier: GPL-3.0-or-later
00019  *
00020  * -----
00021  */
00022
00023 #ifndef GNSS_SDR_GALILEO_E1_DLL_PLL_VEML_TRACKING_FPGA_H
00024 #define GNSS_SDR_GALILEO_E1_DLL_PLL_VEML_TRACKING_FPGA_H
00025
00026 #include "dll_pll_veml_tracking_fpga.h"
00027 #include "tracking_interface.h"
00028 #include <string>
00029
00030 /** \addtogroup Tracking
00031  * \{ */
00032 /** \addtogroup Tracking_adapters
00033  * \{ */
00034
00035
00036 class ConfigurationInterface;
00037
00038 /*!
00039  * \brief This class Adapts a DLL+PLL VEML (Very Early Minus Late) tracking
00040  * loop block to a TrackingInterface for Galileo E1 signals
00041  */
00042 class GalileoE1DllPllVemlTrackingFpga : public TrackingInterface
00043 {
00044 public:
00045     /*!
00046      * \brief Constructor
00047      */
00048     GalileoE1DllPllVemlTrackingFpga(
00049         const ConfigurationInterface* configuration,
00050         const std::string& role,
00051         unsigned int in_streams,
00052         unsigned int out_streams);
00053
00054     /*!
00055      * \brief Destructor
00056      */
00057     virtual ~GalileoE1DllPllVemlTrackingFpga();
00058
00059     /*!
00060      * \brief Role
00061      */
00062     inline std::string role() override
00063     {
00064         return role_;
00065     }
00066
00067     /*!
00068      * \brief Returns "Galileo_E1_DLL_PLL_VEML_Tracking_FPGA"
00069      */
00070     inline std::string implementation() override
00071     {
00072         return "Galileo_E1_DLL_PLL_VEML_Tracking_FPGA";
00073     }
00074
00075     /*!
00076      * \brief Returns size of lv_l6sc_t
00077      */
00078     size_t item_size() override
00079     {
00080         return sizeof(int16_t);
00081     }
00082
00083     /*!
00084      * \brief Connect
00085      */
00086     void connect(gr::top_block_sptr top_block) override;
00087
00088     /*!
00089      * \brief Disconnect
00090      */
00091     void disconnect(gr::top_block_sptr top_block) override;
00092
00093     /*!
00094      * \brief Get left block
00095      */
00096     gr::basic_block_sptr get_left_block() override;
00097
00098     /*!
00099      * \brief Get right block
00100      */
00101     gr::basic_block_sptr get_right_block() override;
00102
00103     /*!
00104      * \brief Set tracking channel unique ID

```



## 11.488 galileo\_e1\_tcp\_connector\_tracking.h File Reference

Include dependency graph for galileo\_e1\_tcp\_connector\_tracking.h:



### 11.488.1 Detailed Description

Interface of an adapter of a TCP connector block based on code DLL + carrier PLL for Galileo E1 to a [TrackingInterface](#).

#### Author

David Pubill, 2012. dpubill(at)cttc.es Luis Esteve, 2012. luis(at)epsilon-formacion.com Javier Arribas, 2011. jarribas(at)cttc.es

Code DLL + carrier PLL according to the algorithms described in: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen, A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach, Birkhauser, 2007  
GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2012-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [galileo\\_e1\\_tcp\\_connector\\_tracking.h](#).

## 11.489 galileo\_e1\_tcp\_connector\_tracking.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file galileo_e1_tcp_connector_tracking.h
00003  * \brief Interface of an adapter of a TCP connector block based on code DLL + carrier PLL
00004  * for Galileo E1 to a TrackingInterface
00005  * \author David Pubill, 2012. dpubill(at)cttc.es
00006  *         Luis Esteve, 2012. luis(at)epsilon-formacion.com
00007  *         Javier Arribas, 2011. jarribas(at)cttc.es
00008  *
00009  * Code DLL + carrier PLL according to the algorithms described in:
00010  * K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen,
00011  * A Software-Defined GPS and Galileo Receiver. A Single-Frequency
00012  * Approach, Birkhauser, 2007
00013  *
00014  * -----
00015  *
00016  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00017  * This file is part of GNSS-SDR.
00018  *
00019  * Copyright (C) 2012-2020 (see AUTHORS file for a list of contributors)
00020  * SPDX-License-Identifier: GPL-3.0-or-later
00021  *
00022  * -----
00023  */
00024
00025 #ifndef GNSS_SDR_GALILEO_E1_TCP_CONNECTOR_TRACKING_H
00026 #define GNSS_SDR_GALILEO_E1_TCP_CONNECTOR_TRACKING_H
00027
00028 #include "galileo_e1_tcp_connector_tracking_cc.h"
00029 #include "tracking_interface.h"
00030 #include <string>
00031
00032 /** \addtogroup Tracking
00033  * \{ */
00034 /** \addtogroup Tracking_adapters
00035  * \{ */
00036
00037
00038 class ConfigurationInterface;
00039
00040 /*!
00041  * \brief This class implements a code DLL + carrier PLL tracking loop
00042  */
00043 class GalileoE1TcpConnectorTracking : public TrackingInterface
00044 {
00045 public:
00046     GalileoE1TcpConnectorTracking(
00047         const ConfigurationInterface* configuration,
00048         const std::string& role,
00049         unsigned int in_streams,
00050         unsigned int out_streams);
00051
00052     ~GalileoE1TcpConnectorTracking() = default;
00053
00054     inline std::string role() override
00055     {
00056         return role_;
00057     }
00058
00059     ///! Returns "Galileo_E1_TCP_CONNECTOR_Tracking"
00060     inline std::string implementation() override
00061     {
```

```

00062         return "Galileo_E1_TCP_CONNECTOR_Tracking";
00063     }
00064
00065     inline size_t item_size() override
00066     {
00067         return item_size_;
00068     }
00069
00070     void connect(gr::top_block_sptr top_block) override;
00071     void disconnect(gr::top_block_sptr top_block) override;
00072     gr::basic_block_sptr get_left_block() override;
00073     gr::basic_block_sptr get_right_block() override;
00074
00075     /*!
00076      * \brief Set tracking channel unique ID
00077      */
00078     void set_channel(unsigned int channel) override;
00079
00080     /*!
00081      * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00082      * to efficiently exchange synchronization data between acquisition and
00083      * tracking blocks
00084      */
00085     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00086
00087     void start_tracking() override;
00088     /*!
00089      * \brief Stop running tracking
00090      */
00091     void stop_tracking() override;
00092
00093 private:
00094     galileo_e1_tcp_connector_tracking_cc_sptr tracking_sptr_;
00095     std::string role_;
00096     size_t item_size_;
00097     unsigned int channel_;
00098     unsigned int in_streams_;
00099     unsigned int out_streams_;
00100 };
00101
00102
00103 /** \} */
00104 /** \} */
00105 #endif // GNSS_SDR_GALILEO_E1_TCP_CONNECTOR_TRACKING_H

```

## 11.490 galileo\_e5a\_dll\_pll\_tracking.h File Reference

Adapts a code DLL + carrier PLL tracking block to a [TrackingInterface](#) for Galileo E5a signals.

```
#include "dll_pll_veml_tracking.h"
```

```
#include "tracking_interface.h"
```

```
#include <string>
```

Include dependency graph for galileo e5a dll pll tracking.h:



## Classes

- class `GalileoE5aDIIPIITracking`

*This class implements a code DLL + carrier PLL tracking loop.*

### 11.490.1 Detailed Description

Adapts a code DLL + carrier PLL tracking block to a [TrackingInterface](#) for Galileo E5a signals.

## Author

Marc Sales, 2014. marcsales92(at)gmail.com \based on work from:

- Javier Arribas, 2011. jarribas(at)cttc.es
- Luis Esteve, 2012. luis(at)epsilon-formacion.com

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [galileo\\_e5a\\_dll\\_pll\\_tracking.h](#).

## 11.491 galileo\_e5a\_dll\_pll\_tracking.h

[Go to the documentation of this file.](#)

```
00001  /*!
00002  * \file galileo_e5a_dll_pll_tracking.h
00003  * \brief Adapts a code DLL + carrier PLL
00004  * tracking block to a TrackingInterface for Galileo E5a signals
00005  * \author Marc Sales, 2014. marcsales92(at)gmail.com
00006  * \based on work from:
00007  *      <ul>
00008  *          <li> Javier Arribas, 2011. jarribas(at)cttc.es
00009  *          <li> Luis Esteve, 2012. luis(at)epsilon-formacion.com
00010  *      </ul>
00011  *
00012  * -----
00013  *
00014  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00015  * This file is part of GNSS-SDR.
00016  *
00017  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00018  * SPDX-License-Identifier: GPL-3.0-or-later
00019  *
00020  * -----
00021  */
00022
00023 #ifndef GNSS_SDR_GALILEO_E5A_DLL_PLL_TRACKING_H
00024 #define GNSS_SDR_GALILEO_E5A_DLL_PLL_TRACKING_H
00025
00026 #include "dll_pll_veml_tracking.h"
00027 #include "tracking_interface.h"
00028 #include <string>
00029
00030 /** \addtogroup Tracking
00031  * \{ */
00032 /** \addtogroup Tracking_adapters
00033  * \{ */
00034
00035
00036 class ConfigurationInterface;
00037
00038 /*!
00039 * \brief This class implements a code DLL + carrier PLL tracking loop
00040 */
00041 class GalileoE5aDllPllTracking : public TrackingInterface
00042 {
00043 public:
00044     GalileoE5aDllPllTracking(
00045         const ConfigurationInterface* configuration,
00046         const std::string& role,
00047         unsigned int in_streams,
00048         unsigned int out_streams);
00049
00050     ~GalileoE5aDllPllTracking() = default;
00051
00052     inline std::string role() override
00053     {
00054         return role_;
00055     }
00056
00057     /*! Returns "Galileo_E5a_DLL_PLL_Tracking"
00058     inline std::string implementation() override
00059     {
00060         return "Galileo_E5a_DLL_PLL_Tracking";
00061     }
00062
00063     inline size_t item_size() override
00064     {
00065         return item_size_;
00066     }
00067
00068     void connect(gr::top_block_sptr top_block) override;
```

```

00069 void disconnect(gr::top_block_sptr top_block) override;
00070 gr::basic_block_sptr get_left_block() override;
00071 gr::basic_block_sptr get_right_block() override;
00072
00073 /*!
00074  * \brief Set tracking channel unique ID
00075  */
00076 void set_channel(unsigned int channel) override;
00077
00078 /*!
00079  * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00080  * to efficiently exchange synchronization data between acquisition and tracking blocks
00081  */
00082 void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00083
00084 void start_tracking() override;
00085
00086 /*!
00087  * \brief Stop running tracking
00088  */
00089 void stop_tracking() override;
00090
00091 private:
00092     dll_pll_veml_tracking_sptr tracking_sptr_;
00093     std::string role_;
00094     size_t item_size_;
00095     unsigned int channel_;
00096     unsigned int in_streams_;
00097     unsigned int out_streams_;
00098 };
00099
00100
00101 /** \} */
00102 /** \} */
00103 #endif // GNSS_SDR_GALILEO_E5A_DLL_PLL_TRACKING_H

```

## 11.492 galileo\_e5a\_dll\_pll\_tracking\_fpga.h File Reference

Adapts a code DLL + carrier PLL tracking block to a [TrackingInterface](#) for Galileo E5a signals for the FPGA.

```
#include "dll_pll_veml_tracking_fpga.h"
```

```
#include "tracking_interface.h"
```

```
#include <string>
```

Include dependency graph for galileo\_e5a\_dll\_pll\_tracking\_fpga.h:



## Classes

- class `GalileoE5aDIIPIITrackingFpga`

*This class implements a code DLL + carrier PLL tracking loop.*

### 11.492.1 Detailed Description

Adapts a code DLL + carrier PLL tracking block to a [TrackingInterface](#) for Galileo E5a signals for the FPGA.

## Author

Marc Majoral, 2019. mmajoral(at)cttc.cat

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [galileo\\_e5a\\_dll\\_pll\\_tracking\\_fpga.h](#).

## 11.493 galileo e5a dll pll tracking fpga.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file galileo_e5a_dll_pll_tracking_fpga.h
00003  * \brief Adapts a code DLL + carrier PLL
00004  * tracking block to a TrackingInterface for Galileo E5a signals for the FPGA
00005  * \author Marc Majoral, 2019. mmajoral(at)cttc.cat
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_GALILEO_E5A_DLL_PLL_TRACKING_FPGA_H
00019 #define GNSS_SDR_GALILEO_E5A_DLL_PLL_TRACKING_FPGA_H
00020
00021 #include "dll_pll_veml_tracking_fpga.h"
00022 #include "tracking_interface.h"
00023 #include <string>
00024
00025 /** \addtogroup Tracking
00026  * \{ */
00027 /** \addtogroup Tracking_adapters
00028  * \{ */
00029
00030
00031 class ConfigurationInterface;
00032
00033 /*!
00034 * \brief This class implements a code DLL + carrier PLL tracking loop
00035 */
00036 class GalileoE5aDllPllTrackingFpga : public TrackingInterface
00037 {
00038 public:
00039     /*!
00040     * \brief Constructor
00041     */
00042     GalileoE5aDllPllTrackingFpga(
00043         const ConfigurationInterface* configuration,
00044         const std::string& role,
00045         unsigned int in_streams,
00046         unsigned int out_streams);
00047
00048     /*!
00049     * \brief Destructor
00050     */
00051     virtual ~GalileoE5aDllPllTrackingFpga();
00052
00053     /*!
00054     * \brief Role
00055     */
00056     inline std::string role() override
00057     {
00058         return role_;
00059     }
00060
00061     /*!
00062     * \brief Returns "Galileo_E5a_DLL_PLL_Tracking_FPGA"
00063     */
00064     inline std::string implementation() override
00065     {
00066         return "Galileo_E5a_DLL_PLL_Tracking_FPGA";
00067     }
00068
00069     /*!
00070     * \brief Returns size of lv_16sc_t
00071     */
00072     size_t item_size() override
00073     {
00074         return sizeof(int16_t);
00075     }
00076
00077     /*!
00078     * \brief Connect
00079     */
00080     void connect(gr::top_block_sptr top_block) override;
00081
00082     /*!
00083     * \brief Disconnect
00084     */
00085     void disconnect(gr::top_block_sptr top_block) override;
00086
00087     /*!

```

## 11.494 galileo\_e5b\_dll\_pll\_tracking.h File Reference

```
#include "dll_pll_veml_tracking.h"
#include "tracking_interface.h"
#include <string>
```

[illegible]

**Classes**

- class [GalileoE5bDllPllTracking](#)

*This class implements a code DLL + carrier PLL tracking loop.*

**11.494.1 Detailed Description**

Adapts a code DLL + carrier PLL tracking block to a [TrackingInterface](#) for Galileo E5b signals.

**Author**

Piyush Gupta, 2020. [piyush04111999@gmail.com](mailto:piyush04111999@gmail.com) \based on work from:

- Javier Arribas, 2011. [jarribas@cttc.es](mailto:jarribas@cttc.es)
- Luis Esteve, 2012. [luis@epsilon-formacion.com](mailto:luis@epsilon-formacion.com)
- Marc Sales, 2014. [marcsales92@gmail.com](mailto:marcsales92@gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [galileo\\_e5b\\_dll\\_pll\\_tracking.h](#).

---

**11.495 galileo\_e5b\_dll\_pll\_tracking.h**

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file galileo_e5b_dll_pll_tracking.h
00003  * \brief Adapts a code DLL + carrier PLL
00004  * \brief tracking block to a TrackingInterface for Galileo E5b signals
00005  * \author Piyush Gupta, 2020. piyush04111999@gmail.com
00006  * \based on work from:
00007  *      <ul>
00008  *          <li> Javier Arribas, 2011. jarribas@cttc.es
00009  *          <li> Luis Esteve, 2012. luis@epsilon-formacion.com
00010  *          <li> Marc Sales, 2014. marcsales92@gmail.com
00011  *      </ul>
00012  *
00013  * -----
00014  *
00015  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00016  * This file is part of GNSS-SDR.
00017  *
00018  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00019  * SPDX-License-Identifier: GPL-3.0-or-later
00020  *
00021  * -----
00022  */
00023
00024 #ifndef GNSS_SDR_GALILEO_E5B_DLL_PLL_TRACKING_H
00025 #define GNSS_SDR_GALILEO_E5B_DLL_PLL_TRACKING_H
00026
00027 #include "dll_pll_veml_tracking.h"
00028 #include "tracking_interface.h"
00029 #include <string>
00030
00031 /** \addtogroup Tracking
00032  * \{ */
00033 /** \addtogroup Tracking_adapters
00034  * \{ */
00035
00036
00037 class ConfigurationInterface;
00038
00039 /*!
00040  * \brief This class implements a code DLL + carrier PLL tracking loop
00041  */
00042 class GalileoE5bDllPllTracking : public TrackingInterface
00043 {
00044 public:
00045     GalileoE5bDllPllTracking(
00046         const ConfigurationInterface* configuration,
00047         const std::string& role,
00048         unsigned int in_streams,
00049         unsigned int out_streams);
00050
00051     ~GalileoE5bDllPllTracking() = default;
00052

```



```

00053     inline std::string role() override
00054     {
00055         return role_;
00056     }
00057
00058     /// Returns "Galileo_E5b_DLL_PLL_Tracking"
00059     inline std::string implementation() override
00060     {
00061         return "Galileo_E5b_DLL_PLL_Tracking";
00062     }
00063
00064     inline size_t item_size() override
00065     {
00066         return item_size_;
00067     }
00068
00069     /*!
00070     * \brief Connect
00071     */
00072     void connect(gr::top_block_sptr top_block) override;
00073
00074     /*!
00075     * \brief Disconnect
00076     */
00077     void disconnect(gr::top_block_sptr top_block) override;
00078
00079     /*!
00080     * \brief Get left block
00081     */
00082     gr::basic_block_sptr get_left_block() override;
00083
00084     /*!
00085     * \brief Get right block
00086     */
00087     gr::basic_block_sptr get_right_block() override;
00088
00089     /*!
00090     * \brief Set tracking channel unique ID
00091     */
00092     void set_channel(unsigned int channel) override;
00093
00094     /*!
00095     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00096     * to efficiently exchange synchronization data between acquisition and tracking blocks
00097     */
00098     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00099
00100     void start_tracking() override;
00101
00102     /*!
00103     * \brief Stop running tracking
00104     */
00105     void stop_tracking() override;
00106
00107 private:
00108     dll_pll_veml_tracking_sptr tracking_sptr_;
00109     std::string role_;
00110     size_t item_size_;
00111     unsigned int channel_;
00112     unsigned int in_streams_;
00113     unsigned int out_streams_;
00114 };
00115
00116 /** \} */
00117 /** \} */
00118 /** \} */
00119 #endif // GNSS_SDR_GALILEO_E5B_DLL_PLL_TRACKING_H

```

## 11.496 galileo\_e6\_dll\_pll\_tracking.h File Reference

Adapts a code DLL + carrier PLL tracking block to a [TrackingInterface](#) for Galileo E6 signals.

```

#include "dll_pll_veml_tracking.h"
#include "tracking_interface.h"
#include <string>

```



```

00046 ~GalileoE6DllPllTracking() = default;
00047
00048 inline std::string role() override
00049 {
00050     return role_;
00051 }
00052
00053 //! Returns "Galileo_E6_DLL_PLL_Tracking"
00054 inline std::string implementation() override
00055 {
00056     return "Galileo_E6_DLL_PLL_Tracking";
00057 }
00058
00059 inline size_t item_size() override
00060 {
00061     return item_size_;
00062 }
00063
00064 /*!
00065  * \brief Connect
00066  */
00067 void connect(gr::top_block_sptr top_block) override;
00068
00069 /*!
00070  * \brief Disconnect
00071  */
00072 void disconnect(gr::top_block_sptr top_block) override;
00073
00074 /*!
00075  * \brief Get left block
00076  */
00077 gr::basic_block_sptr get_left_block() override;
00078
00079 /*!
00080  * \brief Get right block
00081  */
00082 gr::basic_block_sptr get_right_block() override;
00083
00084 /*!
00085  * \brief Set tracking channel unique ID
00086  */
00087 void set_channel(unsigned int channel) override;
00088
00089 /*!
00090  * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00091  * to efficiently exchange synchronization data between acquisition and tracking blocks
00092  */
00093 void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00094
00095 void start_tracking() override;
00096
00097 /*!
00098  * \brief Stop running tracking
00099  */
00100 void stop_tracking() override;
00101
00102 private:
00103     dll_pll_veml_tracking_sptr tracking_sptr_;
00104     std::string role_;
00105     size_t item_size_;
00106     unsigned int channel_;
00107     unsigned int in_streams_;
00108     unsigned int out_streams_;
00109 };
00110
00111
00112 /** \} */
00113 /** \} */
00114 #endif // GNSS_SDR_GALILEO_E6_DLL_PLL_TRACKING_H

```

## 11.498 glonass\_l1\_ca\_dll\_pll\_c\_aid\_tracking.h File Reference

Interface of an adapter of a DLL+PLL tracking loop block for Glonass L1 C/A to a [TrackingInterface](#).

```

#include "glonass_l1_ca_dll_pll_c_aid_tracking_cc.h"
#include "glonass_l1_ca_dll_pll_c_aid_tracking_sc.h"
#include "tracking_interface.h"
#include <string>

```



```

00039
00040 class ConfigurationInterface;
00041
00042 /*!
00043  * \brief This class implements a code DLL + carrier PLL tracking loop
00044  */
00045 class GlonassL1CaDllPllCAidTracking : public TrackingInterface
00046 {
00047 public:
00048     GlonassL1CaDllPllCAidTracking(
00049         const ConfigurationInterface* configuration,
00050         const std::string& role,
00051         unsigned int in_streams,
00052         unsigned int out_streams);
00053
00054     ~GlonassL1CaDllPllCAidTracking() = default;
00055
00056     inline std::string role() override
00057     {
00058         return role_;
00059     }
00060
00061     /*! Returns "GLONASS_L1_CA_DLL_PLL_C_Aid_Tracking"
00062     inline std::string implementation() override
00063     {
00064         return "GLONASS_L1_CA_DLL_PLL_C_Aid_Tracking";
00065     }
00066
00067     inline size_t item_size() override
00068     {
00069         return item_size_;
00070     }
00071
00072     void connect(gr::top_block_sptr top_block) override;
00073     void disconnect(gr::top_block_sptr top_block) override;
00074     gr::basic_block_sptr get_left_block() override;
00075     gr::basic_block_sptr get_right_block() override;
00076
00077     /*!
00078     * \brief Set tracking channel unique ID
00079     */
00080     void set_channel(unsigned int channel) override;
00081
00082     /*!
00083     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00084     * to efficiently exchange synchronization data between acquisition and tracking blocks
00085     */
00086     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00087
00088     void start_tracking() override;
00089     /*!
00090     * \brief Stop running tracking
00091     */
00092     void stop_tracking() override;
00093
00094 private:
00095     glonass_l1_ca_dll_pll_c_aid_tracking_cc_sptr tracking_cc_sptr_;
00096     glonass_l1_ca_dll_pll_c_aid_tracking_sc_sptr tracking_sc_sptr_;
00097     std::string role_;
00098     std::string item_type_;
00099     size_t item_size_;
00100     unsigned int channel_;
00101     unsigned int in_streams_;
00102     unsigned int out_streams_;
00103 };
00104
00105
00106 /** \} */
00107 /** \} */
00108 #endif // GNSS_SDR_GLOMSS_L1_CA_DLL_PLL_C_AID_TRACKING_H

```

## 11.500 glonass\_l1\_ca\_dll\_pll\_tracking.h File Reference

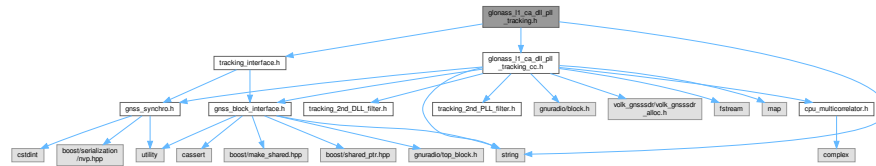
Interface of an adapter of a DLL+PLL tracking loop block for Glonass L1 C/A to a [TrackingInterface](#).

```

#include "glonass_l1_ca_dll_pll_tracking_cc.h"
#include "tracking_interface.h"
#include <string>

```

Include dependency graph for `glonass_l1_ca_dll_pll_tracking.h`:



## Classes

- class [GlonassL1CaDllPllTracking](#)

*This class implements a code DLL + carrier PLL tracking loop.*

## 11.500.1 Detailed Description

Interface of an adapter of a DLL+PLL tracking loop block for Glonass L1 C/A to a [TrackingInterface](#).

### Author

Gabriel Araujo, 2017. [gabriel.araujo.5000\(at\)gmail.com](mailto:gabriel.araujo.5000(at)gmail.com)

Luis Esteve, 2017. [luis\(at\)epsilon-formacion.com](mailto:luis(at)epsilon-formacion.com)

Code DLL + carrier PLL according to the algorithms described in: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen, A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach, Birkha user, 2007  
GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [glonass\\_l1\\_ca\\_dll\\_pll\\_tracking.h](#).

## 11.501 glonass\_l1\_ca\_dll\_pll\_tracking.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file glonass_l1_ca_dll_pll_tracking.h
00003  * \brief Interface of an adapter of a DLL+PLL tracking loop block
00004  * for Glonass L1 C/A to a TrackingInterface
00005  * \author Gabriel Araujo, 2017. gabriel.araujo.5000(at)gmail.com
00006  * \author Luis Esteve, 2017. luis(at)epsilon-formacion.com
00007  *
00008  *
00009  * Code DLL + carrier PLL according to the algorithms described in:
00010  * K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen,
00011  * A Software-Defined GPS and Galileo Receiver. A Single-Frequency
00012  * Approach, Birkha user, 2007
00013  *
00014  * -----
00015  *
00016  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00017  * This file is part of GNSS-SDR.
00018  *
00019  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00020  * SPDX-License-Identifier: GPL-3.0-or-later
00021  *
00022  * -----
00023  */
00024
00025 #ifndef GNSS_SDR_GLOPASS_L1_CA_DLL_PLL_TRACKING_H
00026 #define GNSS_SDR_GLOPASS_L1_CA_DLL_PLL_TRACKING_H
00027
00028 #include "glonass_l1_ca_dll_pll_tracking_cc.h"
00029 #include "tracking_interface.h"
00030 #include <string>
00031
00032 /** \addtogroup Tracking
00033  * \{ */
00034 /** \addtogroup Tracking_adapters
00035  * \{ */
00036
00037
```

```

00038 class ConfigurationInterface;
00039
00040 /*!
00041  * \brief This class implements a code DLL + carrier PLL tracking loop
00042  */
00043 class GlonassL1CaDllPllTracking : public TrackingInterface
00044 {
00045 public:
00046     GlonassL1CaDllPllTracking(
00047         const ConfigurationInterface* configuration,
00048         const std::string& role,
00049         unsigned int in_streams,
00050         unsigned int out_streams);
00051
00052     ~GlonassL1CaDllPllTracking() = default;
00053
00054     inline std::string role() override
00055     {
00056         return role_;
00057     }
00058
00059     /*! Returns "GLONASS_L1_CA_DLL_PLL_Tracking"
00060     inline std::string implementation() override
00061     {
00062         return "GLONASS_L1_CA_DLL_PLL_Tracking";
00063     }
00064
00065     inline size_t item_size() override
00066     {
00067         return item_size_;
00068     }
00069
00070     void connect(gr::top_block_sptr top_block) override;
00071     void disconnect(gr::top_block_sptr top_block) override;
00072     gr::basic_block_sptr get_left_block() override;
00073     gr::basic_block_sptr get_right_block() override;
00074
00075     /*!
00076     * \brief Set tracking channel unique ID
00077     */
00078     void set_channel(unsigned int channel) override;
00079
00080     /*!
00081     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00082     * to efficiently exchange synchronization data between acquisition and tracking blocks
00083     */
00084     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00085
00086     void start_tracking() override;
00087
00088     /*!
00089     * \brief Stop running tracking
00090     */
00091     void stop_tracking() override;
00092
00093 private:
00094     glonass_l1_ca_dll_pll_tracking_cc_sptr tracking_sptr_;
00095     std::string role_;
00096     size_t item_size_;
00097     unsigned int channel_;
00098     unsigned int in_streams_;
00099     unsigned int out_streams_;
00100 };
00101
00102
00103 /** \} */
00104 /** \} */
00105 #endif // GNSS_SDR_GLOASS_L1_CA_DLL_PLL_TRACKING_H

```

## 11.502 glonass\_l2\_ca\_dll\_pll\_c\_aid\_tracking.h File Reference

Interface of an adapter of a DLL+PLL tracking loop block for Glonass L2 C/A to a [TrackingInterface](#).

```

#include "glonass_l2_ca_dll_pll_c_aid_tracking_cc.h"
#include "glonass_l2_ca_dll_pll_c_aid_tracking_sc.h"
#include "tracking_interface.h"
#include <string>

```





```

00043 class GlonassL2CaDllPllCAidTracking : public TrackingInterface
00044 {
00045 public:
00046     GlonassL2CaDllPllCAidTracking(
00047         const ConfigurationInterface* configuration,
00048         const std::string& role,
00049         unsigned int in_streams,
00050         unsigned int out_streams);
00051
00052     ~GlonassL2CaDllPllCAidTracking() = default;
00053
00054     inline std::string role() override
00055     {
00056         return role_;
00057     }
00058
00059     /*! Returns "GLONASS_L2_CA_DLL_PLL_C_Aid_Tracking"
00060     inline std::string implementation() override
00061     {
00062         return "GLONASS_L2_CA_DLL_PLL_C_Aid_Tracking";
00063     }
00064
00065     inline size_t item_size() override
00066     {
00067         return item_size_;
00068     }
00069
00070     void connect(gr::top_block_sptr top_block) override;
00071     void disconnect(gr::top_block_sptr top_block) override;
00072     gr::basic_block_sptr get_left_block() override;
00073     gr::basic_block_sptr get_right_block() override;
00074
00075     /*!
00076      * \brief Set tracking channel unique ID
00077      */
00078     void set_channel(unsigned int channel) override;
00079
00080     /*!
00081      * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00082      * to efficiently exchange synchronization data between acquisition and tracking blocks
00083      */
00084     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00085
00086     void start_tracking() override;
00087
00088     /*!
00089      * \brief Stop running tracking
00090      */
00091     void stop_tracking() override;
00092
00093 private:
00094     glonass_l2_ca_dll_pll_c_aid_tracking_cc_sptr tracking_cc_sptr_;
00095     glonass_l2_ca_dll_pll_c_aid_tracking_sc_sptr tracking_sc_sptr_;
00096     std::string role_;
00097     std::string item_type_;
00098     size_t item_size_;
00099     unsigned int channel_;
00100     unsigned int in_streams_;
00101     unsigned int out_streams_;
00102 };
00103
00104
00105 /** \} */
00106 /** \} */
00107 #endif // GNSS_SDR_GLOASS_L2_CA_DLL_PLL_C_AID_TRACKING_H

```

## 11.504 glonass\_l2\_ca\_dll\_pll\_tracking.h File Reference

Interface of an adapter of a DLL+PLL tracking loop block for Glonass L2 C/A to a [TrackingInterface](#).

```

#include "glonass_l2_ca_dll_pll_tracking_cc.h"
#include "tracking_interface.h"
#include <string>

```



```

00040  * \brief This class implements a code DLL + carrier PLL tracking loop
00041  */
00042  class GlonassL2CaDllPllTracking : public TrackingInterface
00043  {
00044  public:
00045      GlonassL2CaDllPllTracking(
00046          const ConfigurationInterface* configuration,
00047          const std::string& role,
00048          unsigned int in_streams,
00049          unsigned int out_streams);
00050
00051      ~GlonassL2CaDllPllTracking() = default;
00052
00053      inline std::string role() override
00054      {
00055          return role_;
00056      }
00057
00058      //! Returns "GLONASS_L1_CA_DLL_PLL_Tracking"
00059      inline std::string implementation() override
00060      {
00061          return "GLONASS_L2_CA_DLL_PLL_Tracking";
00062      }
00063
00064      inline size_t item_size() override
00065      {
00066          return item_size_;
00067      }
00068
00069      void connect(gr::top_block_sptr top_block) override;
00070      void disconnect(gr::top_block_sptr top_block) override;
00071      gr::basic_block_sptr get_left_block() override;
00072      gr::basic_block_sptr get_right_block() override;
00073
00074      /*!
00075       * \brief Set tracking channel unique ID
00076       */
00077      void set_channel(unsigned int channel) override;
00078
00079      /*!
00080       * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00081       * to efficiently exchange synchronization data between acquisition and tracking blocks
00082       */
00083      void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00084
00085      void start_tracking() override;
00086
00087      /*!
00088       * \brief Stop running tracking
00089       */
00090      void stop_tracking() override;
00091
00092  private:
00093      glonass_l2_ca_dll_pll_tracking_cc_sptr tracking_sptr_;
00094      std::string role_;
00095      size_t item_size_;
00096      unsigned int channel_;
00097      unsigned int in_streams_;
00098      unsigned int out_streams_;
00099  };
00100
00101
00102  /** \} */
00103  /** \} */
00104  #endif // GNSS_SDR_GLOASS_L2_CA_DLL_PLL_TRACKING_H

```

## 11.506 gps\_l1\_ca\_dll\_pll\_tracking.h File Reference

Interface of an adapter of a DLL+PLL tracking loop block for GPS L1 C/A to a [TrackingInterface](#).

```

#include "dll_pll_veml_tracking.h"
#include "tracking_interface.h"
#include <string>

```



## 11.508 gps\_l1\_ca\_dll\_pll\_tracking\_fpga.h File Reference

Include dependency graph for `gps_l1_ca_dll_pll_tracking_fpga.h`:



## Classes

- class [GpsL1CaDllPllTrackingFpga](#)

*This class implements a code DLL + carrier PLL tracking loop.*

### 11.508.1 Detailed Description

Interface of an adapter of a DLL+PLL tracking loop block for GPS L1 C/A to a [TrackingInterface](#) for the FPGA.

#### Author

Marc Majoral, 2019, mmajoral(at)cttc.es

Code DLL + carrier PLL according to the algorithms described in: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen, A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach, Birkhauser, 2007  
 GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gps\\_l1\\_ca\\_dll\\_pll\\_tracking\\_fpga.h](#).

## 11.509 gps\_l1\_ca\_dll\_pll\_tracking\_fpga.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file gps_l1_ca_dll_pll_tracking_fpga.h
00003  * \brief Interface of an adapter of a DLL+PLL tracking loop block
00004  * for GPS L1 C/A to a TrackingInterface for the FPGA
00005  * \author Marc Majoral, 2019, mmajoral(at)cttc.es
00006  *
00007  * Code DLL + carrier PLL according to the algorithms described in:
00008  * K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen,
00009  * A Software-Defined GPS and Galileo Receiver. A Single-Frequency
00010  * Approach, Birkhauser, 2007
00011  *
00012  * -----
00013  *
00014  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00015  * This file is part of GNSS-SDR.
00016  *
00017  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00018  * SPDX-License-Identifier: GPL-3.0-or-later
00019  *
00020  * -----
00021  */
00022
00023 #ifndef GNSS_SDR_GPS_L1_CA_DLL_PLL_TRACKING_FPGA_H
00024 #define GNSS_SDR_GPS_L1_CA_DLL_PLL_TRACKING_FPGA_H
00025
00026 #include "dll_pll_veml_tracking_fpga.h"
00027 #include "tracking_interface.h"
00028 #include <string>
00029
00030 /** \addtogroup Tracking
00031  * \{ */
00032 /** \addtogroup Tracking_adapters
00033  * \{ */
00034
00035
00036 class ConfigurationInterface;
00037
00038 /*!
00039  * \brief This class implements a code DLL + carrier PLL tracking loop
00040  */
00041 class GpsL1CaDllPllTrackingFpga : public TrackingInterface
00042 {
00043 public:
00044     /*!
00045      * \brief Constructor
00046      */
00047     GpsL1CaDllPllTrackingFpga (
00048         const ConfigurationInterface* configuration,
00049         const std::string& role,
00050         unsigned int in_streams,
00051         unsigned int out_streams);
00052
00053     /*!
00054      * \brief Destructor
00055      */
00056     virtual ~GpsL1CaDllPllTrackingFpga();
```

```

00057
00058     /*!
00059     * \brief Role
00060     */
00061     inline std::string role() override
00062     {
00063         return role_;
00064     }
00065
00066     /*!
00067     * \brief Returns "GPS_L1_CA_DLL_PLL_Tracking_FPGA"
00068     */
00069     inline std::string implementation() override
00070     {
00071         return "GPS_L1_CA_DLL_PLL_Tracking_FPGA";
00072     }
00073
00074     /*!
00075     * \brief Returns size of lv_l6sc_t
00076     */
00077     size_t item_size() override
00078     {
00079         return sizeof(intl6_t);
00080     }
00081
00082     /*!
00083     * \brief Connect
00084     */
00085     void connect(gr::top_block_sptr top_block) override;
00086
00087     /*!
00088     * \brief Disconnect
00089     */
00090     void disconnect(gr::top_block_sptr top_block) override;
00091
00092     /*!
00093     * \brief Get left block
00094     */
00095     gr::basic_block_sptr get_left_block() override;
00096
00097     /*!
00098     * \brief Get right block
00099     */
00100     gr::basic_block_sptr get_right_block() override;
00101
00102     /*!
00103     * \brief Set tracking channel unique ID
00104     */
00105     void set_channel(unsigned int channel) override;
00106
00107     /*!
00108     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00109     * to efficiently exchange synchronization data between acquisition and tracking blocks
00110     */
00111     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00112
00113     /*!
00114     * \brief Start the tracking process in the FPGA
00115     */
00116     void start_tracking() override;
00117
00118     /*!
00119     * \brief Stop the tracking process in the FPGA
00120     */
00121     void stop_tracking() override;
00122
00123 private:
00124     const std::string default_device_name_GPS_L1 = "multicorrelator_resampler_S00_AXI"; // UIO
00125     const std::string default_device_name_Galileo_E1 = "multicorrelator_resampler_5_1_AXI"; // UIO
00126     alternative device name
00127
00127     static const uint32_t NUM_PRNs = 32; // total number of PRNs
00128     static const int32_t GPS_CA_BIT_DURATION_MS = 20;
00129     // the following flag is FPGA-specific and they are using arrange the values of the local code in
00130     the way the FPGA
00130     // expects. This arrangement is done in the initialisation to avoid consuming unnecessary clock
00131     cycles during tracking.
00131     static const int32_t LOCAL_CODE_FPGA_ENABLE_WRITE_MEMORY = 0x0C000000; // flag that enables WE
00132     (Write Enable) of the local code FPGA
00132
00133     dll_pll_veml_tracking_fpga_sptr tracking_fpga_sc_sptr_;
00134     std::string role_;
00135     std::string device_name_;
00136     int32_t* ca_codes_ptr_;
00137     uint32_t num_prev_assigned_ch_;
00138     uint32_t channel_;

```

```

00139     uint32_t in_streams_;
00140     uint32_t out_streams_;
00141 };
00142
00143
00144 /** \} */
00145 /** \} */
00146 #endif // GNSS_SDR_GPS_L1_CA_DLL_PLL_TRACKING_FPGA_H

```

## 11.510 gps\_l1\_ca\_dll\_pll\_tracking\_gpu.h File Reference

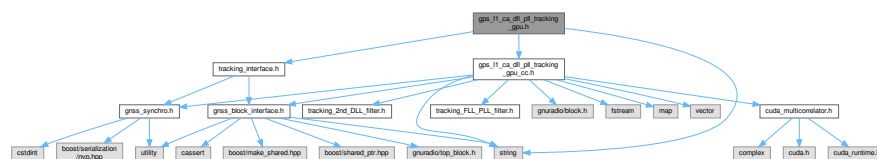
Implementation of an adapter of a DLL+PLL tracking loop block using GPU accelerated functions for GPS L1 C/A to a [TrackingInterface](#).

```
#include "gps_l1_ca_dll_pll_tracking_gpu_cc.h"
```

```
#include "tracking_interface.h"
```

```
#include <string>
```

Include dependency graph for `gps_l1_ca_dll_pll_tracking_gpu.h`:



### Classes

- class [GpsL1CaDllPllTrackingGPU](#)

*This class implements a code DLL + carrier PLL tracking loop using GPU accelerated functions.*

### 11.510.1 Detailed Description

Implementation of an adapter of a DLL+PLL tracking loop block using GPU accelerated functions for GPS L1 C/A to a [TrackingInterface](#).

#### Author

Javier Arribas, 2015. jarribas(at)cttc.es

Code DLL + carrier PLL according to the algorithms described in: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen, A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach, Birkhauser, 2007

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [gps\\_l1\\_ca\\_dll\\_pll\\_tracking\\_gpu.h](#).

## 11.511 gps\_l1\_ca\_dll\_pll\_tracking\_gpu.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002   * \file gps_l1_ca_dll_pll_tracking_gpu.h
00003   * \brief Implementation of an adapter of a DLL+PLL tracking loop block using GPU accelerated
00004   *        functions
00005   * for GPS L1 C/A to a TrackingInterface
00006   * \author Javier Arribas, 2015. jarribas(at)cttc.es
00007   *
00008   * Code DLL + carrier PLL according to the algorithms described in:
00009   * K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen,
00010   * A Software-Defined GPS and Galileo Receiver. A Single-Frequency
00011   * Approach, Birkhauser, 2007
00012   *
00013   * -----
00014   * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00015   * This file is part of GNSS-SDR.

```



```

00016  *
00017  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00018  * SPDX-License-Identifier: GPL-3.0-or-later
00019  *
00020  * -----
00021  */
00022
00023 #ifndef GNSS_SDR_GPS_L1_CA_DLL_PLL_TRACKING_GPU_H
00024 #define GNSS_SDR_GPS_L1_CA_DLL_PLL_TRACKING_GPU_H
00025
00026 #include "gps_l1_ca_dll_pll_tracking_gpu_cc.h"
00027 #include "tracking_interface.h"
00028 #include <string>
00029
00030 /** \addtogroup Tracking
00031  * \{ */
00032 /** \addtogroup Tracking_adapters
00033  * \{ */
00034
00035
00036 class ConfigurationInterface;
00037
00038 /*!
00039  * \brief This class implements a code DLL + carrier PLL tracking loop using GPU accelerated functions
00040  */
00041 class GpsL1CaDllPllTrackingGPU : public TrackingInterface
00042 {
00043 public:
00044     GpsL1CaDllPllTrackingGPU(
00045         const ConfigurationInterface* configuration,
00046         std::string role,
00047         unsigned int in_streams,
00048         unsigned int out_streams);
00049
00050     virtual ~GpsL1CaDllPllTrackingGPU();
00051
00052     inline std::string role() override
00053     {
00054         return role_;
00055     }
00056
00057     /*! Returns "GPS_L1_CA_DLL_PLL_Tracking_GPU"
00058     inline std::string implementation() override
00059     {
00060         return "GPS_L1_CA_DLL_PLL_Tracking_GPU";
00061     }
00062
00063     inline size_t item_size() override
00064     {
00065         return item_size_;
00066     }
00067
00068     void connect(gr::top_block_sptr top_block) override;
00069     void disconnect(gr::top_block_sptr top_block) override;
00070     gr::basic_block_sptr get_left_block() override;
00071     gr::basic_block_sptr get_right_block() override;
00072
00073     /*!
00074     * \brief Set tracking channel unique ID
00075     */
00076     void set_channel(unsigned int channel) override;
00077
00078     /*!
00079     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00080     * to efficiently exchange synchronization data between acquisition and tracking blocks
00081     */
00082     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00083
00084     void start_tracking() override;
00085
00086     /*!
00087     * \brief Stop running tracking
00088     */
00089     void stop_tracking() override;
00090
00091 private:
00092     gps_l1_ca_dll_pll_tracking_gpu_cc_sptr tracking_;
00093     size_t item_size_;
00094     unsigned int channel_;
00095     std::string role_;
00096     unsigned int in_streams_;
00097     unsigned int out_streams_;
00098 };
00099
00100
00101 /** \} */
00102 /** \} */

```



```

00024  */
00025
00026 #ifndef GNSS_SDR_GPS_L1_CA_GAUSSIAN_TRACKING_H
00027 #define GNSS_SDR_GPS_L1_CA_GAUSSIAN_TRACKING_H
00028
00029 #include "gps_l1_ca_gaussian_tracking_cc.h"
00030 #include "tracking_interface.h"
00031 #include <string>
00032
00033 /** \addtogroup Tracking
00034  * \{ */
00035 /** \addtogroup Tracking_adapters
00036  * \{ */
00037
00038
00039 class ConfigurationInterface;
00040
00041 /*!
00042  * \brief This class implements a code DLL + carrier PLL tracking loop
00043  */
00044 class GpsL1CaGaussianTracking : public TrackingInterface
00045 {
00046 public:
00047     GpsL1CaGaussianTracking(
00048         const ConfigurationInterface* configuration,
00049         const std::string& role,
00050         unsigned int in_streams,
00051         unsigned int out_streams);
00052
00053     ~GpsL1CaGaussianTracking() = default;
00054
00055     inline std::string role() override
00056     {
00057         return role_;
00058     }
00059
00060     //! Returns "GPS_L1_CA_Gaussian_Tracking"
00061     inline std::string implementation() override
00062     {
00063         return "GPS_L1_CA_Gaussian_Tracking";
00064     }
00065
00066     inline size_t item_size() override
00067     {
00068         return item_size_;
00069     }
00070
00071     void connect(gr::top_block_sptr top_block) override;
00072     void disconnect(gr::top_block_sptr top_block) override;
00073     gr::basic_block_sptr get_left_block() override;
00074     gr::basic_block_sptr get_right_block() override;
00075
00076     /*!
00077     * \brief Set tracking channel unique ID
00078     */
00079     void set_channel(unsigned int channel) override;
00080
00081     /*!
00082     * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00083     * to efficiently exchange synchronization data between acquisition and tracking blocks
00084     */
00085     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00086
00087     void start_tracking() override;
00088
00089     /*!
00090     * \brief Stop running tracking
00091     */
00092     void stop_tracking() override;
00093
00094 private:
00095     gps_l1_ca_gaussian_tracking_cc_sptr tracking_sptr_;
00096     std::string role_;
00097     size_t item_size_;
00098     unsigned int channel_;
00099     unsigned int in_streams_;
00100     unsigned int out_streams_;
00101 };
00102
00103
00104 /** \} */
00105 /** \} */
00106 #endif // GNSS_SDR_GPS_L1_CA_GAUSSIAN_TRACKING_H

```

## 11.514 gps\_l1\_ca\_kf\_tracking.h File Reference

Interface of an adapter of a code + carrier Kalman Filter tracking loop with VTL capabilities block for GPS L1 C/A to a [TrackingInterface](#).

```
#include "kf_tracking.h"
#include "tracking_interface.h"
#include <string>
```

Include dependency graph for `gps_l1_ca_kf_tracking.h`:



### Classes

- class [GpsL1CaKfTracking](#)

*This class implements a code + carrier Kalman Filter tracking loop with VTL capabilities.*

### 11.514.1 Detailed Description

Interface of an adapter of a code + carrier Kalman Filter tracking loop with VTL capabilities block for GPS L1 C/A to a [TrackingInterface](#).

#### Author

Javier Arribas, 2020. jarribas(at)cttc.es

---

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)

GNSS-SDR is a software defined Global Navigation Satellite Systems receiver

This file is part of GNSS-SDR.

SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [gps\\_l1\\_ca\\_kf\\_tracking.h](#).

---

## 11.515 gps\_l1\_ca\_kf\_tracking.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file gps_l1_ca_kf_tracking.h
00003  * \brief Interface of an adapter of a code + carrier Kalman Filter tracking
00004  * loop with VTL capabilities block
00005  * for GPS L1 C/A to a TrackingInterface
00006  * \author Javier Arribas, 2020. jarribas(at)cttc.es
00007  *
00008  *
00009  * -----
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  *
00013  * GNSS-SDR is a software defined Global Navigation
00014  * Satellite Systems receiver
00015  *
00016  * This file is part of GNSS-SDR.
00017  *
00018  * SPDX-License-Identifier: GPL-3.0-or-later
00019  *
00020  * -----
00021  */
00022
00023 #ifndef GNSS_SDR_GPS_L1_CA_KF_TRACKING_H
00024 #define GNSS_SDR_GPS_L1_CA_KF_TRACKING_H
00025
00026 #include "kf_tracking.h"
00027 #include "tracking_interface.h"
00028 #include <string>
00029
00030 class ConfigurationInterface;
00031
00032 /*!
```

```

00033  * \brief This class implements a code + carrier Kalman Filter tracking loop
00034  * with VTL capabilities
00035  */
00036  class GpsL1CaKfTracking : public TrackingInterface
00037  {
00038  public:
00039      GpsL1CaKfTracking(
00040          const ConfigurationInterface* configuration,
00041          const std::string& role,
00042          unsigned int in_streams,
00043          unsigned int out_streams);
00044
00045      ~GpsL1CaKfTracking() = default;
00046
00047      inline std::string role() override
00048      {
00049          return role_;
00050      }
00051
00052      //! Returns "GPS_L1_CA_KF_Tracking"
00053      inline std::string implementation() override
00054      {
00055          return "GPS_L1_CA_KF_Tracking";
00056      }
00057
00058      inline size_t item_size() override
00059      {
00060          return item_size_;
00061      }
00062
00063      void connect(gr::top_block_sptr top_block) override;
00064      void disconnect(gr::top_block_sptr top_block) override;
00065      gr::basic_block_sptr get_left_block() override;
00066      gr::basic_block_sptr get_right_block() override;
00067
00068      /*!
00069       * \brief Set tracking channel unique ID
00070       */
00071      void set_channel(unsigned int channel) override;
00072
00073      /*!
00074       * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00075       * to efficiently exchange synchronization data between acquisition
00076       * and tracking blocks
00077       */
00078      void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00079
00080      void start_tracking() override;
00081
00082      /*!
00083       * \brief Stop running tracking
00084       */
00085      void stop_tracking() override;
00086
00087  private:
00088      kf_tracking_sptr tracking_sptr_;
00089      std::string role_;
00090      size_t item_size_;
00091      unsigned int channel_;
00092      unsigned int in_streams_;
00093      unsigned int out_streams_;
00094  };
00095
00096  #endif // GNSS_SDR_GPS_L1_CA_KF_TRACKING_H

```

## 11.516 gps\_l1\_ca\_tcp\_connector\_tracking.h File Reference

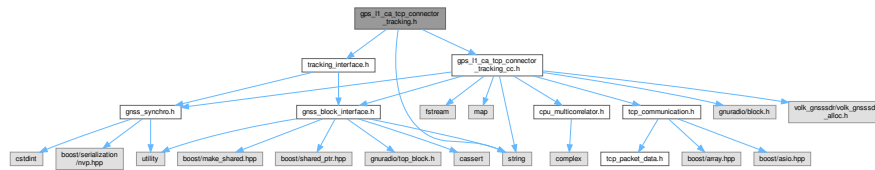
Interface of an adapter of a TCP connector block based on code DLL + carrier PLL for GPS L1 C/A to a [TrackingInterface](#).

```

#include "gps_l1_ca_tcp_connector_tracking_cc.h"
#include "tracking_interface.h"
#include <string>

```

Include dependency graph for `gps_l1_ca_tcp_connector_tracking.h`:



## Classes

- class [GpsL1CaTcpConnectorTracking](#)  
*This class implements a code DLL + carrier PLL tracking loop.*

### 11.516.1 Detailed Description

Interface of an adapter of a TCP connector block based on code DLL + carrier PLL for GPS L1 C/A to a [TrackingInterface](#).

#### Author

David Pubill, 2012. [dpubill\(at\)cttc.es](mailto:dpubill(at)cttc.es) Javier Arribas, 2011. [jarribas\(at\)cttc.es](mailto:jarribas(at)cttc.es)

Code DLL + carrier PLL according to the algorithms described in: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen, A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach, Birkhauser, 2007  
GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [gps\\_l1\\_ca\\_tcp\\_connector\\_tracking.h](#).

### 11.517 `gps_l1_ca_tcp_connector_tracking.h`

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file gps_l1_ca_tcp_connector_tracking.h
00003  * \brief Interface of an adapter of a TCP connector block based on code DLL + carrier PLL
00004  * for GPS L1 C/A to a TrackingInterface
00005  * \author David Pubill, 2012. dpubill\(at\)cttc.es
00006  *          Javier Arribas, 2011. jarribas\(at\)cttc.es
00007  *
00008  * Code DLL + carrier PLL according to the algorithms described in:
00009  * K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen,
00010  * A Software-Defined GPS and Galileo Receiver. A Single-Frequency
00011  * Approach, Birkhauser, 2007
00012  *
00013  * -----
00014  *
00015  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00016  * This file is part of GNSS-SDR.
00017  *
00018  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00019  * SPDX-License-Identifier: GPL-3.0-or-later
00020  *
00021  * -----
00022  */
00023
00024 #ifndef GNSS_SDR_GPS_L1_CA_TCP_CONNECTOR_TRACKING_H
00025 #define GNSS_SDR_GPS_L1_CA_TCP_CONNECTOR_TRACKING_H
00026
00027 #include "gps_l1_ca_tcp_connector_tracking_cc.h"
00028 #include "tracking_interface.h"
00029 #include <string>
00030
00031 /** \addtogroup Tracking
00032  * \{ */
00033 /** \addtogroup Tracking_adapters
00034  * \{ */
00035
00036
00037 class ConfigurationInterface;
00038
```

```

00039  /*!
00040  * \brief This class implements a code DLL + carrier PLL tracking loop
00041  */
00042  class GpsLlCaTcpConnectorTracking : public TrackingInterface
00043  {
00044  public:
00045      GpsLlCaTcpConnectorTracking(
00046          const ConfigurationInterface* configuration,
00047          const std::string& role,
00048          unsigned int in_streams,
00049          unsigned int out_streams);
00050
00051      ~GpsLlCaTcpConnectorTracking() = default;
00052
00053      inline std::string role() override
00054      {
00055          return role_;
00056      }
00057
00058      /*! Returns "GPS_L1_CA_TCP_CONNECTOR_Tracking"
00059      inline std::string implementation() override
00060      {
00061          return "GPS_L1_CA_TCP_CONNECTOR_Tracking";
00062      }
00063
00064      inline size_t item_size() override
00065      {
00066          return item_size_;
00067      }
00068
00069      void connect(gr::top_block_sptr top_block) override;
00070      void disconnect(gr::top_block_sptr top_block) override;
00071      gr::basic_block_sptr get_left_block() override;
00072      gr::basic_block_sptr get_right_block() override;
00073
00074      /*!
00075      * \brief Set tracking channel unique ID
00076      */
00077      void set_channel(unsigned int channel) override;
00078
00079      /*!
00080      * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00081      * to efficiently exchange synchronization data between acquisition and tracking blocks
00082      */
00083      void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00084
00085      void start_tracking() override;
00086
00087      /*!
00088      * \brief Stop running tracking
00089      */
00090      void stop_tracking() override;
00091
00092  private:
00093      gps_ll_ca_tcp_connector_tracking_cc_sptr tracking_sptr_;
00094      std::string role_;
00095      size_t item_size_;
00096      unsigned int channel_;
00097      unsigned int in_streams_;
00098      unsigned int out_streams_;
00099  };
00100
00101
00102  /** \} */
00103  /** \} */
00104  #endif // GNSS_SDR_GPS_L1_CA_TCP_CONNECTOR_TRACKING_H

```

## 11.518 gps\_l2\_m\_dll\_pll\_tracking.h File Reference

Interface of an adapter of a DLL+PLL tracking loop block for GPS L1 C/A to a [TrackingInterface](#).

```

#include "dll_pll_veml_tracking.h"
#include "tracking_interface.h"
#include <string>

```





```

00043 {
00044     public:
00045         GpsL2MD11P11Tracking(
00046             const ConfigurationInterface* configuration,
00047             const std::string& role,
00048             unsigned int in_streams,
00049             unsigned int out_streams);
00050
00051         ~GpsL2MD11P11Tracking() = default;
00052
00053         inline std::string role() override
00054         {
00055             return role_;
00056         }
00057
00058         //! Returns "GPS_L2_M_DLL_PLL_Tracking"
00059         inline std::string implementation() override
00060         {
00061             return "GPS_L2_M_DLL_PLL_Tracking";
00062         }
00063
00064         inline size_t item_size() override
00065         {
00066             return item_size_;
00067         }
00068
00069         void connect(gr::top_block_sptr top_block) override;
00070         void disconnect(gr::top_block_sptr top_block) override;
00071         gr::basic_block_sptr get_left_block() override;
00072         gr::basic_block_sptr get_right_block() override;
00073
00074         /*!
00075          * \brief Set tracking channel unique ID
00076          */
00077         void set_channel(unsigned int channel) override;
00078
00079         /*!
00080          * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00081          * to efficiently exchange synchronization data between acquisition and tracking blocks
00082          */
00083         void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00084
00085         void start_tracking() override;
00086
00087         /*!
00088          * \brief Stop running tracking
00089          */
00090         void stop_tracking() override;
00091
00092     private:
00093         dll_pll_veml_tracking_sptr tracking_sptr_;
00094         std::string role_;
00095         size_t item_size_;
00096         unsigned int channel_;
00097         unsigned int in_streams_;
00098         unsigned int out_streams_;
00099 };
00100
00101
00102 /** @} */
00103 /** @} */
00104 #endif // GNSS_SDR_GPS_L2_M_DLL_PLL_TRACKING_H

```

## 11.520 gps\_l2\_m\_dll\_pll\_tracking\_fpga.h File Reference

Interface of an adapter of a DLL+PLL tracking loop block for GPS L2C to a [TrackingInterface](#) for the FPGA.

```

#include "dll_pll_veml_tracking_fpga.h"
#include "tracking_interface.h"
#include <gnuradio/runtime_types.h>
#include <cstdint>
#include <string>

```



```

00043  */
00044  class GpsL2MD11P11TrackingFpga : public TrackingInterface
00045  {
00046  public:
00047      GpsL2MD11P11TrackingFpga(
00048          const ConfigurationInterface* configuration,
00049          const std::string& role,
00050          unsigned int in_streams,
00051          unsigned int out_streams);
00052
00053      virtual ~GpsL2MD11P11TrackingFpga();
00054
00055      inline std::string role() override
00056      {
00057          return role_;
00058      }
00059
00060      /*! Returns "GPS_L2_M_DLL_PLL_Tracking_FPGA"
00061      inline std::string implementation() override
00062      {
00063          return "GPS_L2_M_DLL_PLL_Tracking_FPGA";
00064      }
00065
00066      inline size_t item_size() override
00067      {
00068          return sizeof(int);
00069      }
00070
00071      void connect(gr::top_block_sptr top_block) override;
00072      void disconnect(gr::top_block_sptr top_block) override;
00073      gr::basic_block_sptr get_left_block() override;
00074      gr::basic_block_sptr get_right_block() override;
00075
00076      /*!
00077       * \brief Set tracking channel unique ID
00078       */
00079      void set_channel(unsigned int channel) override;
00080
00081      /*!
00082       * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00083       * to efficiently exchange synchronization data between acquisition and tracking blocks
00084       */
00085      void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00086
00087      void start_tracking() override;
00088
00089      /*!
00090       * \brief Stop running tracking
00091       */
00092      void stop_tracking() override;
00093
00094  private:
00095      const std::string default_device_name_GPS_L2 = "multicorrelator_resampler_S00_AXI"; // UIO device
00096      name
00097      static const uint32_t NUM_PRNs = 32;
00098
00099      dll_pll_veml_tracking_fpga_sptr tracking_fpga_sc_sptr_;
00100      std::string role_;
00101      std::string device_name_;
00102      int* prn_codes_ptr_;
00103      uint32_t num_prev_assigned_ch_;
00104      unsigned int channel_;
00105      unsigned int in_streams_;
00106      unsigned int out_streams_;
00107  };
00108
00109  /** @} */
00110  /** @} */
00111  #endif // GNSS_SDR_GPS_L2_M_DLL_PLL_TRACKING_FPGA_H

```

## 11.522 gps\_l5\_dll\_pll\_tracking.h File Reference

Interface of an adapter of a DLL+PLL tracking loop block for GPS L5 to a [TrackingInterface](#).

```

#include "dll_pll_veml_tracking.h"
#include "tracking_interface.h"
#include <string>

```



## 11.524 gps\_l5\_dll\_pll\_tracking\_fpga.h File Reference

Include dependency graph for gps l5 dll pll tracking fpga.h:



## Classes

- class [GpsL5DllPllTrackingFpga](#)

*This class implements a code DLL + carrier PLL tracking loop.*

### 11.524.1 Detailed Description

Interface of an adapter of a DLL+PLL tracking loop block for GPS L5 to a [TrackingInterface](#) for the FPGA.

#### Author

Marc Majoral, 2019. mmajoral(at)cttc.cat Javier Arribas, 2019. jarribas(at)cttc.es

Code DLL + carrier PLL according to the algorithms described in: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen, A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach, Birkhauser, 2007  
 GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gps\\_l5\\_dll\\_pll\\_tracking\\_fpga.h](#).

## 11.525 gps\_l5\_dll\_pll\_tracking\_fpga.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file gps_l5_dll_pll_tracking_fpga.h
00003  * \brief Interface of an adapter of a DLL+PLL tracking loop block
00004  * for GPS L5 to a TrackingInterface for the FPGA
00005  * \author Marc Majoral, 2019. mmajoral(at)cttc.cat
00006  *         Javier Arribas, 2019. jarribas(at)cttc.es
00007  *
00008  * Code DLL + carrier PLL according to the algorithms described in:
00009  * K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen,
00010  * A Software-Defined GPS and Galileo Receiver. A Single-Frequency
00011  * Approach, Birkhauser, 2007
00012  *
00013  * -----
00014  *
00015  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00016  * This file is part of GNSS-SDR.
00017  *
00018  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00019  * SPDX-License-Identifier: GPL-3.0-or-later
00020  *
00021  * -----
00022  */
00023
00024 #ifndef GNSS_SDR_GPS_L5_DLL_PLL_TRACKING_FPGA_H
00025 #define GNSS_SDR_GPS_L5_DLL_PLL_TRACKING_FPGA_H
00026
00027 #include "dll_pll_veml_tracking_fpga.h"
00028 #include "tracking_interface.h"
00029 #include <string>
00030
00031 /** \addtogroup Tracking
00032  * \{ */
00033 /** \addtogroup Tracking_adapters
00034  * \{ */
00035
00036
00037 class ConfigurationInterface;
00038
00039 /*!
00040  * \brief This class implements a code DLL + carrier PLL tracking loop
00041  */
00042 class GpsL5DllPllTrackingFpga : public TrackingInterface
00043 {
00044 public:
00045     /*!
00046      * \brief Constructor
00047      */
00048     GpsL5DllPllTrackingFpga(
00049         const ConfigurationInterface* configuration,
00050         const std::string& role,
00051         unsigned int in_streams,
00052         unsigned int out_streams);
00053
00054     /*!
00055      * \brief Destructor
00056      */
00057 }
```

```

00057     virtual ~GpsL5DllPllTrackingFpga();
00058
00059     /*!
00060      * \brief Role
00061      */
00062     inline std::string role() override
00063     {
00064         return role_;
00065     }
00066
00067     /*!
00068      * \brief Returns "GPS_L5_DLL_PLL_Tracking_FPGA"
00069      */
00070     inline std::string implementation() override
00071     {
00072         return "GPS_L5_DLL_PLL_Tracking_FPGA";
00073     }
00074
00075     /*!
00076      * \brief Returns size of lv_l6sc_t
00077      */
00078     size_t item_size() override
00079     {
00080         return sizeof(int16_t);
00081     }
00082
00083     /*!
00084      * \brief Connect
00085      */
00086     void connect(gr::top_block_sptr top_block) override;
00087
00088     /*!
00089      * \brief Disconnect
00090      */
00091     void disconnect(gr::top_block_sptr top_block) override;
00092
00093     /*!
00094      * \brief Get left block
00095      */
00096     gr::basic_block_sptr get_left_block() override;
00097
00098     /*!
00099      * \brief Get right block
00100      */
00101     gr::basic_block_sptr get_right_block() override;
00102
00103     /*!
00104      * \brief Set tracking channel unique ID
00105      */
00106     void set_channel(unsigned int channel) override;
00107
00108     /*!
00109      * \brief Set acquisition/tracking common Gnss_Synchro object pointer
00110      * to efficiently exchange synchronization data between acquisition and tracking blocks
00111      */
00112     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro) override;
00113
00114     /*!
00115      * \brief Start the tracking process in the FPGA
00116      */
00117     void start_tracking() override;
00118
00119     /*!
00120      * \brief Stop the tracking process in the FPGA
00121      */
00122     void stop_tracking() override;
00123
00124 private:
00125     const std::string default_device_name_GPS_L5_ = "multicorrelator_resampler_3_1_AXI"; // UIO
00126     device name
00127
00128     static const uint32_t NUM_PRNs = 32; // total number of PRNs
00129
00130     // the following flags are FPGA-specific and they are using arrange the values of the local code
00131     // in the way the FPGA
00132     // expects. This arrangement is done in the initialisation to avoid consuming unnecessary clock
00133     // cycles during tracking.
00134     static const int32_t LOCAL_CODE_FPGA_ENABLE_WRITE_MEMORY = 0x0C000000; // flag that enables
00135     WE (Write Enable) of the local code FPGA
00136     static const int32_t LOCAL_CODE_FPGA_CORRELATOR_SELECT_COUNT = 0x20000000; // flag that selects
00137     the writing of the pilot code in the FPGA (as opposed to the data code)
00138
00139     dll_pll_veml_tracking_fpga_sptr tracking_fpga_sc_sptr_;
00140     std::string role_;
00141     std::string device_name_;
00142     int32_t* prn_codes_ptr_;
00143     int32_t* data_codes_ptr_;

```

```

00139     uint32_t channel_;
00140     uint32_t num_prev_assigned_ch_;
00141     uint32_t in_streams_;
00142     uint32_t out_streams_;
00143     bool track_pilot_;
00144 };
00145
00146
00147 /** \} */
00148 /** \} */
00149 #endif // GNSS_SDR_GPS_L5_DLL_PLL_TRACKING_FPGA_H

```

## 11.526 dll\_pll\_veml\_tracking.h File Reference

Implementation of a code DLL + carrier PLL tracking block.

```

#include "cpu_multicorrelator_real_codes.h"
#include "dll_pll_conf.h"
#include "exponential_smoother.h"
#include "gnss_block_interface.h"
#include "gnss_time.h"
#include "tracking_FLL_PLL_filter.h"
#include "tracking_loop_filter.h"
#include <boost/circular_buffer.hpp>
#include <gnuradio/block.h>
#include <gnuradio/gr_complex.h>
#include <gnuradio/types.h>
#include <pmt/pmt.h>
#include <volk_gnssssdr/volk_gnssssdr_alloc.h>
#include <cstring>
#include <stdint>
#include <fstream>
#include <string>
#include <typeinfo>
#include <utility>

```

Include dependency graph for `dll_pll_veml_tracking.h`:



This graph shows which files directly or indirectly include this file:



### Classes

- class [dll\\_pll\\_veml\\_tracking](#)  
This class implements a code DLL + carrier PLL tracking block.

### Typedefs

- using [dll\\_pll\\_veml\\_tracking\\_sptr](#) = `gnss_shared_ptr<dll_pll_veml_tracking>`

### Functions

- `dll_pll_veml_tracking_sptr` [dll\\_pll\\_veml\\_make\\_tracking](#) (const [Dll\\_Pll\\_Conf](#) &conf\_)



## 11.526.1 Detailed Description

Implementation of a code DLL + carrier PLL tracking block.

### Author

Javier Arribas, 2018. jarribas(at)cttc.es

Antonio Ramos, 2018 antonio.ramosdet(at)gmail.com

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [dll\\_pll\\_veml\\_tracking.h](#).

---

## 11.527 dll\_pll\_veml\_tracking.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file dll_pll_veml_tracking.h
00003  * \brief Implementation of a code DLL + carrier PLL tracking block.
00004  * \author Javier Arribas, 2018. jarribas(at)cttc.es
00005  * \author Antonio Ramos, 2018 antonio.ramosdet(at)gmail.com
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_DLL_PLL_VEML_TRACKING_H
00019 #define GNSS_SDR_DLL_PLL_VEML_TRACKING_H
00020
00021 #include "cpu_multicorrelator_real_codes.h"
00022 #include "dll_pll_conf.h"
00023 #include "exponential_smoother.h"
00024 #include "gnss_block_interface.h"
00025 #include "gnss_time.h"           // for timetags produced by File_Timestamp_Signal_Source
00026 #include "tracking_FLL_PLL_filter.h" // for PLL/FLL filter
00027 #include "tracking_loop_filter.h" // for DLL filter
00028 #include <boost/circular_buffer.hpp>
00029 #include <gnuradio/block.h>           // for block
00030 #include <gnuradio/gr_complex.h>     // for gr_complex
00031 #include <gnuradio/types.h>          // for gr_vector_int, gr_vector...
00032 #include <pmt/pmt.h>                  // for pmt_t
00033 #include <volk_gnssdr/volk_gnssdr_alloc.h> // for volk_gnssdr::vector
00034 #include <cstdint>                    // for size_t
00035 #include <cstdint>                    // for int32_t
00036 #include <fstream>                    // for ofstream
00037 #include <string>                     // for string
00038 #include <typeinfo>                   // for typeid
00039 #include <utility>                    // for pair
00040
00041 /** \addtogroup Tracking
00042  * \{ */
00043 /** \addtogroup Tracking_gnuradio_blocks tracking_gr_blocks
00044  * GNU Radio blocks for GNSS signal tracking.
00045  * \{ */
00046
00047
00048 class Gnss_Synchro;
00049 class dll_pll_veml_tracking;
00050
00051 using dll_pll_veml_tracking_sptr = gnss_shared_ptr<dll_pll_veml_tracking>;
00052
00053 dll_pll_veml_tracking_sptr dll_pll_veml_make_tracking(const Dll_Pll_Conf &conf_);
00054
00055 /*!
00056 * \brief This class implements a code DLL + carrier PLL tracking block.
00057 */
00058 class dll_pll_veml_tracking : public gr::block
00059 {
00060 public:
00061     ~dll_pll_veml_tracking() override;
00062
00063     void set_channel(uint32_t channel);
00064     void set_gnss_synchro(Gnss_Synchro *p_gnss_synchro);
00065     void start_tracking();

```

```

00066     void stop_tracking();
00067
00068     int general_work(int noutput_items, gr_vector_int &ninput_items,
00069                     gr_vector_const_void_star &input_items, gr_vector_void_star &output_items) override;
00070
00071     void forecast(int noutput_items, gr_vector_int &ninput_items_required) override;
00072
00073 private:
00074     friend dll_pll_veml_tracking_sptr dll_pll_veml_make_tracking(const Dll_Pll_Conf &conf_);
00075     explicit dll_pll_veml_tracking(const Dll_Pll_Conf &conf_);
00076
00077     void msg_handler_telemetry_to_trk(const pmt::pmt_t &msg);
00078     void do_correlation_step(const gr_complex *input_samples);
00079     void run_dll_pll();
00080     void check_carrier_phase_coherent_initialization();
00081     void update_tracking_vars();
00082     void clear_tracking_vars();
00083     void save_correlation_results();
00084     void log_data();
00085     bool cn0_and_tracking_lock_status(double coh_integration_time_s);
00086     bool acquire_secondary();
00087     int64_t uint64diff(uint64_t first, uint64_t second);
00088     int32_t save_matfile() const;
00089
00090     Cpu_Multicorrelator_Real_Codes d_multicorrelator_cpu;
00091     Cpu_Multicorrelator_Real_Codes d_correlator_data_cpu; // for data channel
00092
00093     Dll_Pll_Conf d_trk_parameters;
00094
00095     Exponential_Smoother d_cn0_smoother;
00096     Exponential_Smoother d_carrier_lock_test_smoother;
00097
00098     Tracking_loop_filter d_code_loop_filter;
00099     Tracking_FLL_PLL_filter d_carrier_loop_filter;
00100
00101     Gnss_Synchro *d_acquisition_gnss_synchro;
00102
00103     volk_gnssssdr::vector<float> d_tracking_code;
00104     volk_gnssssdr::vector<float> d_data_code;
00105     volk_gnssssdr::vector<float> d_local_code_shift_chips;
00106     volk_gnssssdr::vector<gr_complex> d_correlator_outs;
00107     volk_gnssssdr::vector<gr_complex> d_Prompt_Data;
00108     volk_gnssssdr::vector<gr_complex> d_Prompt_buffer;
00109
00110     boost::circular_buffer<float> d_dll_filt_history;
00111     boost::circular_buffer<std::pair<double, double>> d_code_ph_history;
00112     boost::circular_buffer<std::pair<double, double>> d_carr_ph_history;
00113     boost::circular_buffer<gr_complex> d_Prompt_circular_buffer;
00114
00115     const size_t int_type_hash_code = typeid(int).hash_code();
00116
00117     double d_signal_carrier_freq;
00118     double d_code_period;
00119     double d_code_chip_rate;
00120     double d_acq_code_phase_samples;
00121     double d_acq_carrier_doppler_hz;
00122     double d_current_correlation_time_s;
00123     double d_carr_phase_error_hz;
00124     double d_carr_freq_error_hz;
00125     double d_carr_error_filt_hz;
00126     double d_code_error_chips;
00127     double d_code_error_filt_chips;
00128     double d_code_freq_chips;
00129     double d_carrier_doppler_hz;
00130     double d_acc_carrier_phase_rad;
00131     double d_rem_code_phase_chips;
00132     double d_T_chip_seconds;
00133     double d_T_prn_seconds;
00134     double d_T_prn_samples;
00135     double d_K_blk_samples;
00136     double d_carrier_lock_test;
00137     double d_CN0_SNV_dB_Hz;
00138     double d_carrier_lock_threshold;
00139     double d_carrier_phase_step_rad;
00140     double d_carrier_phase_rate_step_rad;
00141     double d_code_phase_step_chips;
00142     double d_code_phase_rate_step_chips;
00143     double d_rem_code_phase_samples;
00144
00145     gr_complex *d_Very_Early;
00146     gr_complex *d_Early;
00147     gr_complex *d_Prompt;
00148     gr_complex *d_Late;
00149     gr_complex *d_Very_Late;
00150
00151     gr_complex d_VE_accu;
00152     gr_complex d_E_accu;

```

```

00153     gr_complex d_P_accu;
00154     gr_complex d_P_accu_old;
00155     gr_complex d_L_accu;
00156     gr_complex d_VL_accu;
00157     gr_complex d_P_data_accu;
00158
00159     std::string d_secondary_code_string;
00160     std::string d_data_secondary_code_string;
00161     std::string d_systemName;
00162     std::string d_signal_type;
00163     std::string d_signal_pretty_name;
00164     std::string d_dump_filename;
00165
00166     std::ofstream d_dump_file;
00167
00168     // uint64_t d_sample_counter;
00169     uint64_t d_acq_sample_stamp;
00170     GnssTime d_last_timetag{};
00171     uint64_t d_last_timetag_samplecounter;
00172     bool d_timetag_waiting;
00173
00174     float *d_prompt_data_shift;
00175     float d_rem_carr_phase_rad;
00176
00177     int32_t d_symbols_per_bit;
00178     int32_t d_state;
00179     int32_t d_correlation_length_ms;
00180     int32_t d_n_correlator_taps;
00181     int32_t d_current_prn_length_samples;
00182     int32_t d_extend_correlation_symbols_count;
00183     int32_t d_extend_correlation_symbols;
00184     int32_t d_current_symbol;
00185     int32_t d_current_data_symbol;
00186     int32_t d_cn0_estimation_counter;
00187     int32_t d_carrier_lock_fail_counter;
00188     int32_t d_code_lock_fail_counter;
00189     int32_t d_code_samples_per_chip; // All signals have 1 sample per chip code except Gal. E1 which
has 2 (CBOC disabled) or 12 (CBOC enabled)
00190     int32_t d_code_length_chips;
00191
00192     uint32_t d_channel;
00193     uint32_t d_secondary_code_length;
00194     uint32_t d_data_secondary_code_length;
00195
00196     bool d_pull_in_transitory;
00197     bool d_corrected_doppler;
00198     bool d_interchange_iq;
00199     bool d_veml;
00200     bool d_cloop;
00201     bool d_secondary;
00202     bool d_dump;
00203     bool d_dump_mat;
00204     bool d_acc_carrier_phase_initialized;
00205     bool d_enable_extended_integration;
00206     bool d_Flag_PLL_180_deg_phase_locked;
00207 };
00208
00209
00210 /** \} */
00211 /** \} */
00212 #endif // GNSS_SDR_DLL_PLL_VEML_TRACKING_H

```

## 11.528 dll\_pll\_veml\_tracking\_fpga.h File Reference

Implementation of a code DLL + carrier PLL tracking block using an FPGA.

```

#include "dll_pll_conf_fpga.h"
#include "exponential_smoother.h"
#include "gnss_block_interface.h"
#include "tracking_FLL_PLL_filter.h"
#include "tracking_loop_filter.h"
#include <boost/circular_buffer.hpp>
#include <gnuradio/block.h>
#include <gnuradio/gr_complex.h>
#include <gnuradio/types.h>
#include <pmt/pmt.h>
#include <volk_gnssssdr/volk_gnssssdr_alloc.h>
#include <cstdint>
#include <cstdint>

```



```

00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_DLL_PLL_VEML_TRACKING_FPGA_H
00019 #define GNSS_SDR_DLL_PLL_VEML_TRACKING_FPGA_H
00020
00021 #include "dll_pll_conf_fpga.h"
00022 #include "exponential_smoother.h"
00023 #include "gnss_block_interface.h"
00024 #include "tracking_FLL_PLL_filter.h" // for PLL/FLL filter
00025 #include "tracking_loop_filter.h" // for DLL filter
00026 #include <boost/circular_buffer.hpp>
00027 #include <gnuradio/block.h> // for block
00028 #include <gnuradio/gr_complex.h> // for gr_complex
00029 #include <gnuradio/types.h> // for gr_vector_int, gr_vector...
00030 #include <pmt/pmt.h> // for pmt_t
00031 #include <volk_gnssssdr/volk_gnssssdr_alloc.h> // for volk_gnssssdr::vector
00032 #include <stddef> // for size_t
00033 #include <cstdint> // for int32_t
00034 #include <fstream> // for string, ofstream
00035 #include <memory> // for std::shared_ptr
00036 #include <string> // for string
00037 #include <typeinfo> // for typeid
00038 #include <utility> // for pair
00039
00040 /** \addtogroup Tracking
00041  * \{ */
00042 /** \addtogroup Tracking_gnuradio_blocks
00043  * \{ */
00044
00045
00046 class Fpga_Multicorrelator_8sc;
00047 class Gnss_Synchro;
00048 class dll_pll_veml_tracking_fpga;
00049
00050 using dll_pll_veml_tracking_fpga_sptr = gnss_shared_ptr<dll_pll_veml_tracking_fpga>;
00051
00052 dll_pll_veml_tracking_fpga_sptr dll_pll_veml_make_tracking_fpga(const Dll_Pll_Conf_Fpga &conf_);
00053
00054
00055 /*!
00056  * \brief This class implements a code DLL + carrier PLL tracking block.
00057  */
00058 class dll_pll_veml_tracking_fpga : public gr::block
00059 {
00060 public:
00061     /*!
00062      * \brief Destructor
00063      */
00064     ~dll_pll_veml_tracking_fpga();
00065
00066     /*!
00067      * \brief Set the channel number and configure some multicorrelator parameters
00068      */
00069     void set_channel(uint32_t channel, const std::string &device_io_name);
00070
00071     /*!
00072      * \brief This function is used with two purposes:
00073      * 1 -> To set the gnss_synchro
00074      * 2 -> A set_gnss_synchro command with a valid PRN is received when the system is going to run
00075      * acquisition with that PRN. We can use this command to pre-initialize tracking parameters and
00076      * variables before the actual acquisition process takes place. In this way we minimize the
00077      * latency between acquisition and tracking once the acquisition has been made.
00078      */
00079     void set_gnss_synchro(Gnss_Synchro *p_gnss_synchro);
00080
00081     /*!
00082      * \brief This function starts the tracking process
00083      */
00084     void start_tracking();
00085
00086     /*!
00087      * \brief This function sets a flag that makes general_work to stop in order to finish the
00088      tracking process.
00089      */
00089     void stop_tracking();
00090
00091     /*!
00092      * \brief General Work
00093      */
00094     int general_work(int noutput_items, gr_vector_int &ninput_items,
00095         gr_vector_const_void_star &input_items, gr_vector_void_star &output_items);

```

```

00096
00097     /*!
00098     * \brief This function disables the HW multicorrelator in the FPGA in order to stop the tracking
00099     process
00100     */
00101     void reset();
00102 private:
00103     friend dll_pll_veml_tracking_fpga_sptra dll_pll_veml_make_tracking_fpga(const Dll_Pll_Conf_Fpga
&conf_);
00104     explicit dll_pll_veml_tracking_fpga(const Dll_Pll_Conf_Fpga &conf_);
00105
00106     void msg_handler_telemetry_to_trk(const pmt::pmt_t &msg);
00107     bool cn0_and_tracking_lock_status(double coh_integration_time_s);
00108     bool acquire_secondary();
00109     void do_correlation_step();
00110     void run_dll_pll();
00111     void check_carrier_phase_coherent_initialization();
00112     void update_tracking_vars();
00113     void clear_tracking_vars();
00114     void save_correlation_results();
00115     void log_data();
00116     int32_t save_matfile() const;
00117
00118     Dll_Pll_Conf_Fpga d_trk_parameters;
00119
00120     Exponential_Smoother d_cn0_smoother;
00121     Exponential_Smoother d_carrier_lock_test_smoother;
00122
00123     Gnss_Synchro *d_acquisition_gnss_synchro;
00124
00125     Tracking_loop_filter d_code_loop_filter;
00126
00127     Tracking_FLL_PLL_filter d_carrier_loop_filter;
00128
00129     volk_gnssssdr::vector<float> d_local_code_shift_chips;
00130     volk_gnssssdr::vector<gr_complex> d_correlator_outs;
00131     volk_gnssssdr::vector<gr_complex> d_Prompt_Data;
00132     volk_gnssssdr::vector<gr_complex> d_Prompt_buffer;
00133
00134     boost::circular_buffer<float> d_dll_filt_history;
00135     boost::circular_buffer<std::pair<double, double>> d_code_ph_history;
00136     boost::circular_buffer<std::pair<double, double>> d_carr_ph_history;
00137     boost::circular_buffer<gr_complex> d_Prompt_circular_buffer;
00138
00139     std::string d_systemName;
00140     std::string d_signal_type;
00141     std::string d_secondary_code_string;
00142     std::string d_data_secondary_code_string;
00143     std::string d_signal_pretty_name;
00144     std::string d_dump_filename;
00145
00146     std::ofstream d_dump_file;
00147
00148     std::shared_ptr<Fpga_Multicorrelator_8sc> d_multicorrelator_fpga;
00149
00150     boost::condition_variable d_m_condition;
00151
00152     boost::mutex d_mutex;
00153
00154     const size_t int_type_hash_code = typeid(int).hash_code();
00155
00156     double d_signal_carrier_freq;
00157     double d_code_period;
00158     double d_code_chip_rate;
00159     double d_code_phase_step_chips;
00160     double d_code_phase_rate_step_chips;
00161     double d_carrier_phase_step_rad;
00162     double d_carrier_phase_rate_step_rad;
00163     double d_acq_code_phase_samples;
00164     double d_acq_carrier_doppler_hz;
00165     double d_rem_code_phase_samples;
00166     double d_rem_code_phase_samples_prev;
00167     double d_current_correlation_time_s;
00168     double d_carr_phase_error_hz;
00169     double d_carr_freq_error_hz;
00170     double d_carr_error_filt_hz;
00171     double d_code_error_chips;
00172     double d_code_error_filt_chips;
00173     double d_code_freq_chips;
00174     double d_carrier_doppler_hz;
00175     double d_acc_carrier_phase_rad;
00176     double d_rem_code_phase_chips;
00177     double d_T_chip_seconds;
00178     double d_T_prn_seconds;
00179     double d_T_prn_samples;
00180     double d_K_blk_samples;

```

```

00181     double d_carrier_lock_test;
00182     double d_CNO_SNV_dB_Hz;
00183     double d_carrier_lock_threshold;
00184
00185     gr_complex *d_Very_Early;
00186     gr_complex *d_Early;
00187     gr_complex *d_Prompt;
00188     gr_complex *d_Late;
00189     gr_complex *d_Very_Late;
00190
00191     gr_complex d_VE_accu;
00192     gr_complex d_E_accu;
00193     gr_complex d_P_accu;
00194     gr_complex d_P_accu_old;
00195     gr_complex d_L_accu;
00196     gr_complex d_VL_accu;
00197     gr_complex d_P_data_accu;
00198
00199     uint64_t d_sample_counter;
00200     uint64_t d_acq_sample_stamp;
00201     uint64_t d_sample_counter_next;
00202
00203     float *d_prompt_data_shift;
00204     float d_rem_carr_phase_rad;
00205
00206     int32_t d_symbols_per_bit;
00207     int32_t d_state;
00208     int32_t d_extend_correlation_symbols_count;
00209     int32_t d_current_symbol;
00210     int32_t d_current_data_symbol;
00211     int32_t d_current_integration_length_samples;
00212     int32_t d_cn0_estimation_counter;
00213     int32_t d_carrier_lock_fail_counter;
00214     int32_t d_code_lock_fail_counter;
00215     int32_t d_correlation_length_ms;
00216     int32_t d_n_correlator_taps;
00217     int32_t d_next_integration_length_samples;
00218     int32_t d_extend_fpga_integration_periods;
00219
00220     uint32_t d_channel;
00221     uint32_t d_secondary_code_length;
00222     uint32_t d_data_secondary_code_length;
00223     uint32_t d_code_length_chips;
00224     uint32_t d_code_samples_per_chip; // All signals have 1 sample per chip code except Gal. E1 which
has 2 (CBOC disabled) or 12 (CBOC enabled)
00225     uint32_t d_fpga_integration_period;
00226     uint32_t d_current_fpga_integration_period;
00227
00228     bool d_veml;
00229     bool d_cloop;
00230     bool d_secondary;
00231     bool d_enable_extended_integration;
00232     bool d_dump;
00233     bool d_dump_mat;
00234     bool d_pull_in_transitory;
00235     bool d_corrected_doppler;
00236     bool d_interchange_iq;
00237     bool d_acc_carrier_phase_initialized;
00238     bool d_worker_is_done;
00239     bool d_extended_correlation_in_fpga;
00240     bool d_current_extended_correlation_in_fpga;
00241     bool d_stop_tracking;
00242     bool d_sc_demodulate_enabled;
00243     bool d_Flag_PLL_180_deg_phase_locked;
00244 };
00245
00246
00247 /** \} */
00248 /** \} */
00249 #endif // GNSS_SDR_DLL_PLL_VEML_TRACKING_FPGA_H

```

## 11.530 galileo\_e1\_tcp\_connector\_tracking\_cc.h File Reference

Interface of a TCP connector block based on code DLL + carrier PLL VEML (Very Early Minus Late) tracking block for Galileo E1 signals.

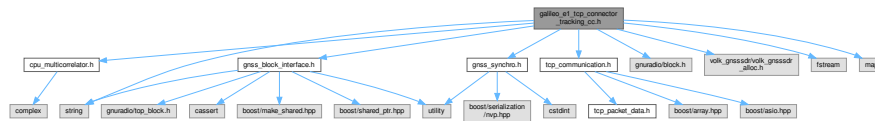
```

#include "cpu_multicorrelator.h"
#include "gnss_block_interface.h"
#include "gnss_synchro.h"
#include "tcp_communication.h"
#include <gnuradio/block.h>
#include <volk_gnssssdr/volk_gnssssdr_alloc.h>

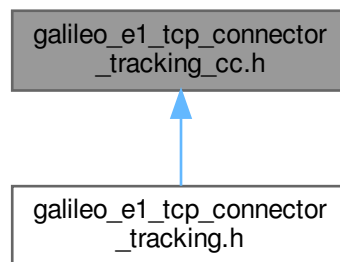
```

```
#include <fstream>
#include <map>
#include <string>
```

Include dependency graph for `galileo_e1_tcp_connector_tracking_cc.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Galileo\\_E1\\_Tcp\\_Connector\\_Tracking\\_cc](#)

*This class implements a code DLL + carrier PLL VEML (Very Early Minus Late) tracking block for Galileo E1 signals.*

## Typedefs

- using [galileo\\_e1\\_tcp\\_connector\\_tracking\\_cc\\_sptr](#) = `gnss_shared_ptr<Galileo_E1_Tcp_Connector_Tracking_cc>`

## Functions

- `galileo_e1_tcp_connector_tracking_cc_sptr` **galileo\_e1\_tcp\_connector\_make\_tracking\_cc** (`int64_t fs_in`, `uint32_t vector_length`, `bool dump`, `const std::string &dump_filename`, `float pll_bw_hz`, `float dll_bw_hz`, `float early_late_space_chips`, `float very_early_late_space_chips`, `size_t port_ch0`)

### 11.530.1 Detailed Description

Interface of a TCP connector block based on code DLL + carrier PLL VEML (Very Early Minus Late) tracking block for Galileo E1 signals.

#### Author

David Pubill, 2012. [dpubill\(at\)cttc.es](mailto:dpubill@cttc.es) Luis Esteve, 2012. [luis\(at\)epsilon-formacion.com](mailto:luis(at)epsilon-formacion.com) Javier Arribas, 2011. [jarribas\(at\)cttc.es](mailto:jarribas(at)cttc.es)

Code DLL + carrier PLL according to the algorithms described in: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen, A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach, Birkhauser, 2007  
GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [galileo\\_e1\\_tcp\\_connector\\_tracking\\_cc.h](#).



## 11.531 galileo\_e1\_tcp\_connector\_tracking\_cc.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file galileo_e1_tcp_connector_tracking_cc.h
00003  * \brief Interface of a TCP connector block based on code DLL + carrier PLL VEML (Very Early
00004  * Minus Late) tracking block for Galileo E1 signals
00005  * \author David Pubill, 2012. dpubill(at)cttc.es
00006  *         Luis Esteve, 2012. luis(at)epsilon-formacion.com
00007  *         Javier Arribas, 2011. jarribas(at)cttc.es
00008  *
00009  * Code DLL + carrier PLL according to the algorithms described in:
00010  * K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen,
00011  * A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach,
00012  * Birkhauser, 2007
00013  *
00014  * -----
00015  *
00016  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00017  * This file is part of GNSS-SDR.
00018  *
00019  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00020  * SPDX-License-Identifier: GPL-3.0-or-later
00021  *
00022  * -----
00023  */
00024
00025 #ifndef GNSS_SDR_GALILEO_E1_TCP_CONNECTOR_TRACKING_CC_H
00026 #define GNSS_SDR_GALILEO_E1_TCP_CONNECTOR_TRACKING_CC_H
00027
00028 #include "cpu_multicorrelator.h"
00029 #include "gnss_block_interface.h"
00030 #include "gnss_synchro.h"
00031 #include "tcp_communication.h"
00032 #include <gnuradio/block.h>
00033 #include <volk_gnssssdr/volk_gnssssdr_alloc.h> // for volk_gnssssdr::vector
00034 #include <fstream>
00035 #include <map>
00036 #include <string>
00037
00038 /** \addtogroup Tracking
00039  * \{ */
00040 /** \addtogroup Tracking_gnuradio_blocks
00041  * \{ */
00042
00043
00044 class Galileo_E1_Tcp_Connector_Tracking_cc;
00045
00046 using galileo_e1_tcp_connector_tracking_cc_sptr =
00047     gnss_shared_ptr<Galileo_E1_Tcp_Connector_Tracking_cc>;
00048
00049 galileo_e1_tcp_connector_tracking_cc_sptr
00050 galileo_e1_tcp_connector_make_tracking_cc(
00051     int64_t fs_in, uint32_t vector_length,
00052     bool dump,
00053     const std::string &dump_filename,
00054     float pll_bw_hz,
00055     float dll_bw_hz,
00056     float early_late_space_chips,
00057     float very_early_late_space_chips,
00058     size_t port_ch0);
00059
00060 /*!
00061  * \brief This class implements a code DLL + carrier PLL VEML (Very Early
00062  * Minus Late) tracking block for Galileo E1 signals
00063  */
00064 class Galileo_E1_Tcp_Connector_Tracking_cc : public gr::block
00065 {
00066 public:
00067     ~Galileo_E1_Tcp_Connector_Tracking_cc();
00068
00069     void set_channel(uint32_t channel);
00070     void set_gnss_synchro(Gnss_Synchro *p_gnss_synchro);
00071     void start_tracking();
00072
00073     int general_work(int noutput_items, gr_vector_int &ninput_items,
00074         gr_vector_const_void_star &input_items, gr_vector_void_star &output_items);
00075
00076     void forecast(int noutput_items, gr_vector_int &ninput_items_required);
00077 private:
00078     friend galileo_e1_tcp_connector_tracking_cc_sptr
00079     galileo_e1_tcp_connector_make_tracking_cc(
00080         int64_t fs_in, uint32_t vector_length,
00081         bool dump,
00082         const std::string &dump_filename,

```

```

00083         float pll_bw_hz,
00084         float dll_bw_hz,
00085         float early_late_space_chips,
00086         float very_early_late_space_chips,
00087         size_t port_ch0);
00088
00089     Galileo_E1_Tcp_Connector_Tracking_cc(
00090         int64_t fs_in, uint32_t vector_length,
00091         bool dump,
00092         const std::string &dump_filename,
00093         float pll_bw_hz,
00094         float dll_bw_hz,
00095         float early_late_space_chips,
00096         float very_early_late_space_chips,
00097         size_t port_ch0);
00098
00099     void update_local_code();
00100
00101     void update_local_carrier();
00102
00103     // tracking configuration vars
00104     uint32_t d_vector_length;
00105     bool d_dump;
00106
00107     Gnss_Synchro *d_acquisition_gnss_synchro;
00108     uint32_t d_channel;
00109
00110     int64_t d_fs_in;
00111
00112     int32_t d_correlation_length_samples;
00113     int32_t d_n_correlator_taps;
00114     float d_early_late_spc_chips;
00115     float d_very_early_late_spc_chips;
00116
00117     volk_gnssssdr::vector<gr_complex> d_ca_code;
00118
00119     gr_complex *d_Very_Early;
00120     gr_complex *d_Early;
00121     gr_complex *d_Prompt;
00122     gr_complex *d_Late;
00123     gr_complex *d_Very_Late;
00124
00125     // remaining code phase and carrier phase between tracking loops
00126     double d_rem_code_phase_samples;
00127     float d_next_rem_code_phase_samples;
00128     float d_rem_carr_phase_rad;
00129
00130     // acquisition
00131     float d_acq_code_phase_samples;
00132     float d_acq_carrier_doppler_hz;
00133
00134     // correlator
00135     volk_gnssssdr::vector<float> d_local_code_shift_chips;
00136     volk_gnssssdr::vector<gr_complex> d_correlator_outs;
00137     Cpu_Multicorrelator multicorrelator_cpu;
00138
00139     // tracking vars
00140     double d_code_freq_chips;
00141     float d_carrier_doppler_hz;
00142     float d_acc_carrier_phase_rad;
00143     float d_acc_code_phase_secs;
00144     float d_code_phase_samples;
00145     size_t d_port_ch0;
00146     size_t d_port;
00147     int32_t d_listen_connection;
00148     float d_control_id;
00149     Tcp_Communication d_tcp_com;
00150
00151     // PRN period in samples
00152     int32_t d_current_prn_length_samples;
00153     int32_t d_next_prn_length_samples;
00154
00155     // processing samples counters
00156     uint64_t d_sample_counter;
00157     uint64_t d_acq_sample_stamp;
00158
00159     // CN0 estimation and lock detector
00160     int32_t d_cn0_estimation_counter;
00161     volk_gnssssdr::vector<gr_complex> d_Prompt_buffer;
00162     float d_carrier_lock_test;
00163     float d_CN0_SNV_dB_Hz;
00164     float d_carrier_lock_threshold;
00165     int32_t d_carrier_lock_fail_counter;
00166
00167     // control vars
00168     bool d_enable_tracking;
00169     bool d_pull_in;

```

```

00170
00171 // file dump
00172 std::string d_dump_filename;
00173 std::ofstream d_dump_file;
00174
00175 std::map<std::string, std::string> systemName;
00176 std::string sys;
00177 };
00178
00179
00180 /** \} */
00181 /** \} */
00182 #endif // GNSS_SDR_GALILEO_E1_TCP_CONNECTOR_TRACKING_CC_H

```

## 11.532 glonass\_l1\_ca\_dll\_pll\_c\_aid\_tracking\_cc.h File Reference

Implementation of a code DLL + carrier PLL tracking block.

```

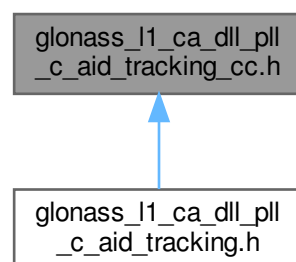
#include "gnss_synchro.h"
#include "tracking_2nd_DLL_filter.h"
#include "tracking_FLL_PLL_filter.h"
#include "cpu_multicorrelator.h"
#include "gnss_block_interface.h"
#include <gnuradio/block.h>
#include <pmt/pmt.h>
#include <volk_gnssssdr/volk_gnssssdr_alloc.h>
#include <deque>
#include <fstream>
#include <map>
#include <string>

```

Include dependency graph for glonass\_l1\_ca\_dll\_pll\_c\_aid\_tracking\_cc.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [glonass\\_l1\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_cc](#)

*This class implements a DLL + PLL tracking loop block.*

## Typedefs

- using `glonass_l1_ca_dll_pll_c_aid_tracking_cc_sptr` = `gnss_shared_ptr<glonass_l1_ca_dll_pll_c_aid_tracking_cc>`

## Functions

- `glonass_l1_ca_dll_pll_c_aid_tracking_cc_sptr glonass_l1_ca_dll_pll_c_aid_make_tracking_cc` (`int64_t fs_in`, `uint32_t vector_length`, `bool dump`, `const std::string &dump_filename`, `float pll_bw_hz`, `float dll_bw_hz`, `float pll_bw_narrow_hz`, `float dll_bw_narrow_hz`, `int32_t extend_correlation_ms`, `float early_late_space_chips`)

### 11.532.1 Detailed Description

Implementation of a code DLL + carrier PLL tracking block.

#### Author

Gabriel Araujo, 2017. gabriel.araujo.5000(at)gmail.com

Luis Esteve, 2017. luis(at)epsilon-formacion.com

Damian Miralles, 2017. dmiralles2009(at)gmail.com

Code DLL + carrier PLL according to the algorithms described in: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen, A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach, Birkha user, 2007  
GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [glonass\\_l1\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_cc.h](#).

## 11.533 glonass\_l1\_ca\_dll\_pll\_c\_aid\_tracking\_cc.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file glonass_l1_ca_dll_pll_c_aid_tracking_cc.h
00003  * \brief Implementation of a code DLL + carrier PLL tracking block
00004  * \author Gabriel Araujo, 2017. gabriel.araujo.5000(at)gmail.com
00005  * \author Luis Esteve, 2017. luis(at)epsilon-formacion.com
00006  * \author Damian Miralles, 2017. dmiralles2009(at)gmail.com
00007  *
00008  *
00009  * Code DLL + carrier PLL according to the algorithms described in:
00010  * K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen,
00011  * A Software-Defined GPS and Galileo Receiver. A Single-Frequency
00012  * Approach, Birkha user, 2007
00013  *
00014  * -----
00015  *
00016  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00017  * This file is part of GNSS-SDR.
00018  *
00019  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00020  * SPDX-License-Identifier: GPL-3.0-or-later
00021  *
00022  * -----
00023  */
00024
00025 #ifndef GNSS_SDR_GLONASS_L1_CA_DLL_PLL_C_AID_TRACKING_CC_H
00026 #define GNSS_SDR_GLONASS_L1_CA_DLL_PLL_C_AID_TRACKING_CC_H
00027
00028 #include "gnss_synchro.h"
00029 #include "tracking_2nd_DLL_filter.h"
00030 #include "tracking_FLL_PLL_filter.h"
00031 // #include "tracking_loop_filter.h"
00032 #include "cpu_multicorrelator.h"
00033 #include "gnss_block_interface.h"
00034 #include <gnuradio/block.h>
00035 #include <pmt/pmt.h>
00036 #include <volk_gnssssdr/volk_gnssssdr_alloc.h> // for volk_gnssssdr::vector
00037 #include <deque>
00038 #include <fstream>
00039 #include <map>
00040 #include <string>
00041
00042 /** \addtogroup Tracking
00043  * \{ */
```

```

00044 /** \addtogroup Tracking_gnuradio_blocks
00045 * \{ */
00046
00047
00048 class glonass_l1_ca_dll_pll_c_aid_tracking_cc;
00049
00050 using glonass_l1_ca_dll_pll_c_aid_tracking_cc_sptr =
    gnss_shared_ptr<glonass_l1_ca_dll_pll_c_aid_tracking_cc>;
00051
00052 glonass_l1_ca_dll_pll_c_aid_tracking_cc_sptr
00053 glonass_l1_ca_dll_pll_c_aid_make_tracking_cc(
00054     int64_t fs_in, uint32_t vector_length,
00055     bool dump,
00056     const std::string& dump_filename,
00057     float pll_bw_hz,
00058     float dll_bw_hz,
00059     float pll_bw_narrow_hz,
00060     float dll_bw_narrow_hz,
00061     int32_t extend_correlation_ms,
00062     float early_late_space_chips);
00063
00064
00065 /*!
00066 * \brief This class implements a DLL + PLL tracking loop block
00067 */
00068 class glonass_l1_ca_dll_pll_c_aid_tracking_cc : public gr::block
00069 {
00070 public:
00071     ~glonass_l1_ca_dll_pll_c_aid_tracking_cc();
00072
00073     void set_channel(uint32_t channel);
00074     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro);
00075     void start_tracking();
00076
00077     int general_work(int noutput_items, gr_vector_int& ninput_items,
00078         gr_vector_const_void_star& input_items, gr_vector_void_star& output_items);
00079
00080     void forecast(int noutput_items, gr_vector_int& ninput_items_required);
00081
00082 private:
00083     friend glonass_l1_ca_dll_pll_c_aid_tracking_cc_sptr
00084     glonass_l1_ca_dll_pll_c_aid_make_tracking_cc(
00085         int64_t fs_in, uint32_t vector_length,
00086         bool dump,
00087         const std::string& dump_filename,
00088         float pll_bw_hz,
00089         float dll_bw_hz,
00090         float pll_bw_narrow_hz,
00091         float dll_bw_narrow_hz,
00092         int32_t extend_correlation_ms,
00093         float early_late_space_chips);
00094
00095     glonass_l1_ca_dll_pll_c_aid_tracking_cc(
00096         int64_t fs_in, uint32_t vector_length,
00097         bool dump,
00098         const std::string& dump_filename,
00099         float pll_bw_hz,
00100         float dll_bw_hz,
00101         float pll_bw_narrow_hz,
00102         float dll_bw_narrow_hz,
00103         int32_t extend_correlation_ms,
00104         float early_late_space_chips);
00105
00106     void msg_handler_preamble_index(const pmt::pmt_t& msg);
00107
00108     void check_carrier_phase_coherent_initialization();
00109
00110     int32_t save_matfile() const;
00111
00112     volk_gnssssdr::vector<gr_complex> d_ca_code;
00113     volk_gnssssdr::vector<float> d_local_code_shift_chips;
00114     volk_gnssssdr::vector<gr_complex> d_correlator_outs;
00115     volk_gnssssdr::vector<gr_complex> d_Prompt_buffer;
00116
00117     Cpu_Multicorrelator multicorrelator_cpu;
00118
00119     // PLL and DLL filter library
00120     // Tracking_2nd_DLL_filter d_code_loop_filter;
00121     Tracking_2nd_DLL_filter d_code_loop_filter;
00122     Tracking_FLL_PLL_filter d_carrier_loop_filter;
00123
00124     // symbol history to detect bit transition
00125     std::deque<gr_complex> d_E_history;
00126     std::deque<gr_complex> d_P_history;
00127     std::deque<gr_complex> d_L_history;
00128
00129     // file dump

```

```

00130     std::string d_dump_filename;
00131     std::ofstream d_dump_file;
00132
00133     std::map<std::string, std::string> systemName;
00134     std::string sys;
00135
00136     // tracking configuration vars
00137     Gnss_Synchro* d_acquisition_gnss_synchro;
00138     int64_t d_fs_in;
00139     double d_glonass_freq_ch;
00140     double d_early_late_spc_chips;
00141     uint32_t d_vector_length;
00142     uint32_t d_channel;
00143     int32_t d_n_correlator_taps;
00144
00145     // remaining code phase and carrier phase between tracking loops
00146     double d_rem_code_phase_samples;
00147     double d_rem_code_phase_chips;
00148     double d_rem_carrier_phase_rad;
00149     int32_t d_rem_code_phase_integer_samples;
00150
00151     // acquisition
00152     double d_acq_code_phase_samples;
00153     double d_acq_carrier_doppler_hz;
00154
00155     // tracking vars
00156     float d_dll_bw_hz;
00157     float d_pll_bw_hz;
00158     float d_dll_bw_narrow_hz;
00159     float d_pll_bw_narrow_hz;
00160     double d_code_freq_chips;
00161     double d_code_phase_step_chips;
00162     double d_carrier_doppler_hz;
00163     double d_carrier_frequency_hz;
00164     double d_carrier_frequency_old_hz;
00165     double d_carrier_phase_step_rad;
00166     double d_acc_carrier_phase_cycles;
00167     double d_code_phase_samples;
00168     double d_pll_to_dll_assist_secs_Ti;
00169     double d_code_error_chips_Ti;
00170     double d_code_error_filt_chips_s;
00171     double d_code_error_filt_chips_Ti;
00172     double d_carr_phase_error_secs_Ti;
00173
00174     double d_preamble_timestamp_samples;
00175     int32_t d_extend_correlation_ms;
00176
00177     // Integration period in samples
00178     int32_t d_correlation_length_samples;
00179
00180     // processing samples counters
00181     uint64_t d_sample_counter;
00182     uint64_t d_acq_sample_stamp;
00183
00184     // CN0 estimation and lock detector
00185     int32_t d_cn0_estimation_counter;
00186     double d_carrier_lock_test;
00187     double d_CN0_SNV_dB_Hz;
00188     double d_carrier_lock_threshold;
00189     int32_t d_carrier_lock_fail_counter;
00190
00191     bool d_enable_extended_integration;
00192     bool d_preamble_synchronized;
00193
00194     // control vars
00195     bool d_enable_tracking;
00196     bool d_pull_in;
00197     bool d_acc_carrier_phase_initialized;
00198
00199     bool d_dump;
00200 };
00201
00202
00203 /** \} */
00204 /** \} */
00205 #endif // GNSS_SDR_GLO_NASS_L1_CA_DLL_PLL_C_AID_TRACKING_CC_H

```

## 11.534 glonass\_l1\_ca\_dll\_pll\_c\_aid\_tracking\_sc.h File Reference

Implementation of a code DLL + carrier PLL tracking block.

```

#include "cpu_multicorrelator_16sc.h"
#include "glonass_l1_signal_replica.h"
#include "gnss_block_interface.h"

```

```

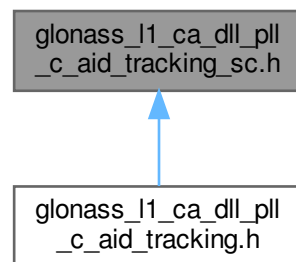
#include "gnss_synchro.h"
#include "tracking_2nd_DLL_filter.h"
#include "tracking_FLL_PLL_filter.h"
#include <gnuradio/block.h>
#include <volk_gnssdr/volk_gnssdr.h>
#include <volk_gnssdr/volk_gnssdr_alloc.h>
#include <deque>
#include <fstream>
#include <map>
#include <string>

```

Include dependency graph for glonass\_l1\_ca\_dll\_pll\_c\_aid\_tracking\_sc.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [glonass\\_l1\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_sc](#)  
*This class implements a DLL + PLL tracking loop block.*

## Typedefs

- using [glonass\\_l1\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_sc\\_sptr](#) = `gnss_shared_ptr<glonass_l1_ca_dll_pll_c_aid_tracking_sc>`

## Functions

- `glonass_l1_ca_dll_pll_c_aid_tracking_sc_sptr` **glonass\_l1\_ca\_dll\_pll\_c\_aid\_make\_tracking\_sc** (int64\_t fs\_in, uint32\_t vector\_length, bool dump, const std::string &dump\_filename, float pll\_bw\_hz, float dll\_bw\_hz, float pll\_bw\_narrow\_hz, float dll\_bw\_narrow\_hz, int32\_t extend\_correlation\_ms, float early\_late\_space\_chips)

### 11.534.1 Detailed Description

Implementation of a code DLL + carrier PLL tracking block.

**Author**

Gabriel Araujo, 2017. gabriel.araujo.5000(at)gmail.com

Luis Esteve, 2017. luis(at)epsilon-formacion.com

Damian Miralles, 2017. dmiralles2009(at)gmail.com

Code DLL + carrier PLL according to the algorithms described in: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen, A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach, Birkha user, 2007  
GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [glonass\\_l1\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_sc.h](#).

**11.535 glonass\_l1\_ca\_dll\_pll\_c\_aid\_tracking\_sc.h**

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file glonass_l1_ca_dll_pll_c_aid_tracking_sc.h
00003  * \brief Implementation of a code DLL + carrier PLL tracking block
00004  * \author Gabriel Araujo, 2017. gabriel.araujo.5000(at)gmail.com
00005  * \author Luis Esteve, 2017. luis(at)epsilon-formacion.com
00006  * \author Damian Miralles, 2017. dmiralles2009(at)gmail.com
00007  *
00008  *
00009  * Code DLL + carrier PLL according to the algorithms described in:
00010  * K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen,
00011  * A Software-Defined GPS and Galileo Receiver. A Single-Frequency
00012  * Approach, Birkha user, 2007
00013  *
00014  * -----
00015  *
00016  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00017  * This file is part of GNSS-SDR.
00018  *
00019  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00020  * SPDX-License-Identifier: GPL-3.0-or-later
00021  *
00022  * -----
00023  */
00024
00025 #ifndef GNSS_SDR_GLOPASS_L1_CA_DLL_PLL_C_AID_TRACKING_SC_H
00026 #define GNSS_SDR_GLOPASS_L1_CA_DLL_PLL_C_AID_TRACKING_SC_H
00027
00028 #include "cpu_multicorrelator_16sc.h"
00029 #include "glonass_l1_signal_replica.h"
00030 #include "gnss_block_interface.h"
00031 #include "gnss_synchro.h"
00032 #include "tracking_2nd_DLL_filter.h"
00033 #include "tracking_FLL_PLL_filter.h"
00034 #include <gnuradio/block.h>
00035 #include <volk_gnssssdr/volk_gnssssdr.h>
00036 #include <volk_gnssssdr/volk_gnssssdr_alloc.h> // for volk_gnssssdr::vector
00037 #include <deque>
00038 #include <fstream>
00039 #include <map>
00040 #include <string>
00041
00042
00043 /** \addtogroup Tracking
00044  * \{ */
00045 /** \addtogroup Tracking_gnuradio_blocks
00046  * \{ */
00047
00048
00049 class glonass_l1_ca_dll_pll_c_aid_tracking_sc;
00050
00051 using glonass_l1_ca_dll_pll_c_aid_tracking_sc_sptr =
00052     gnss_shared_ptr<glonass_l1_ca_dll_pll_c_aid_tracking_sc>;
00053
00054 glonass_l1_ca_dll_pll_c_aid_tracking_sc_sptr
00055 glonass_l1_ca_dll_pll_c_aid_tracking_sc(
00056     int64_t fs_in, uint32_t vector_length,
00057     bool dump,
00058     const std::string& dump_filename,
00059     float pll_bw_hz,
00060     float dll_bw_hz,
00061     float pll_bw_narrow_hz,
00062     float dll_bw_narrow_hz,
00063     int32_t extend_correlation_ms,
00064     float early_late_space_chips);

```



```

00065
00066 /*!
00067  * \brief This class implements a DLL + PLL tracking loop block
00068  */
00069 class glonass_l1_ca_dll_pll_c_aid_tracking_sc : public gr::block
00070 {
00071 public:
00072     ~glonass_l1_ca_dll_pll_c_aid_tracking_sc();
00073
00074     void set_channel(uint32_t channel);
00075     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro);
00076     void start_tracking();
00077
00078     int general_work(int noutput_items, gr_vector_int& ninput_items,
00079                     gr_vector_const_void_star& input_items, gr_vector_void_star& output_items);
00080
00081     void forecast(int noutput_items, gr_vector_int& ninput_items_required);
00082
00083 private:
00084     friend glonass_l1_ca_dll_pll_c_aid_tracking_sc_sptr
00085     glonass_l1_ca_dll_pll_c_aid_tracking_sc(
00086         int64_t fs_in, uint32_t vector_length,
00087         bool dump,
00088         const std::string& dump_filename,
00089         float pll_bw_hz,
00090         float dll_bw_hz,
00091         float pll_bw_narrow_hz,
00092         float dll_bw_narrow_hz,
00093         int32_t extend_correlation_ms,
00094         float early_late_space_chips);
00095
00096     glonass_l1_ca_dll_pll_c_aid_tracking_sc(
00097         int64_t fs_in, uint32_t vector_length,
00098         bool dump,
00099         const std::string& dump_filename,
00100         float pll_bw_hz,
00101         float dll_bw_hz,
00102         float pll_bw_narrow_hz,
00103         float dll_bw_narrow_hz,
00104         int32_t extend_correlation_ms,
00105         float early_late_space_chips);
00106
00107     void msg_handler_preamble_index(const pmt::pmt_t& msg);
00108
00109     void check_carrier_phase_coherent_initialization();
00110
00111     int32_t save_matfile() const;
00112
00113     volk_gnssssdr::vector<gr_complex> d_ca_code;
00114     volk_gnssssdr::vector<lv_16sc_t> d_ca_code_16sc;
00115     volk_gnssssdr::vector<float> d_local_code_shift_chips;
00116     volk_gnssssdr::vector<lv_16sc_t> d_correlator_outs_16sc;
00117     volk_gnssssdr::vector<gr_complex> d_Prompt_buffer;
00118
00119     Cpu_Multicorrelator_16sc multicorrelator_cpu_16sc;
00120
00121     // PLL and DLL filter library
00122     Tracking_2nd_DLL_filter d_code_loop_filter;
00123     Tracking_FLL_PLL_filter d_carrier_loop_filter;
00124
00125     // symbol history to detect bit transition
00126     std::deque<lv_16sc_t> d_E_history;
00127     std::deque<lv_16sc_t> d_P_history;
00128     std::deque<lv_16sc_t> d_L_history;
00129
00130     Gnss_Synchro* d_acquisition_gnss_synchro;
00131
00132     // file dump
00133     std::string d_dump_filename;
00134     std::ofstream d_dump_file;
00135
00136     std::map<std::string, std::string> systemName;
00137     std::string sys;
00138
00139     // tracking configuration vars
00140     int64_t d_fs_in;
00141     int64_t d_glonass_freq_ch;
00142     double d_early_late_spc_chips;
00143     uint32_t d_vector_length;
00144     uint32_t d_channel;
00145     int32_t d_n_correlator_taps;
00146
00147     // remaining code phase and carrier phase between tracking loops
00148     double d_rem_code_phase_samples;
00149     double d_rem_code_phase_chips;
00150     double d_rem_carrier_phase_rad;
00151     int32_t d_rem_code_phase_integer_samples;

```

```

00152
00153 // acquisition
00154 double d_acq_code_phase_samples;
00155 double d_acq_carrier_doppler_hz;
00156
00157 // tracking vars
00158 float d_dll_bw_hz;
00159 float d_pll_bw_hz;
00160 float d_dll_bw_narrow_hz;
00161 float d_pll_bw_narrow_hz;
00162 double d_code_freq_chips;
00163 double d_code_phase_step_chips;
00164 double d_carrier_doppler_hz;
00165 double d_carrier_frequency_hz;
00166 double d_carrier_frequency_old_hz;
00167 double d_carrier_phase_step_rad;
00168 double d_acc_carrier_phase_cycles;
00169 double d_code_phase_samples;
00170 double d_pll_to_dll_assist_secs_Ti;
00171 double d_carr_phase_error_secs_Ti;
00172 double d_code_error_chips_Ti;
00173 double d_preamble_timestamp_samples;
00174 double d_code_error_filt_chips_s;
00175 double d_code_error_filt_chips_Ti;
00176 int32_t d_extend_correlation_ms;
00177
00178 // Integration period in samples
00179 int32_t d_correlation_length_samples;
00180
00181 // processing samples counters
00182 uint64_t d_sample_counter;
00183 uint64_t d_acq_sample_stamp;
00184
00185 // CN0 estimation and lock detector
00186 int32_t d_cn0_estimation_counter;
00187 double d_carrier_lock_test;
00188 double d_CN0_SNV_dB_Hz;
00189 double d_carrier_lock_threshold;
00190 int32_t d_carrier_lock_fail_counter;
00191
00192 bool d_enable_extended_integration;
00193 bool d_preamble_synchronized;
00194
00195 // control vars
00196 bool d_enable_tracking;
00197 bool d_pull_in;
00198 bool d_acc_carrier_phase_initialized;
00199
00200 bool d_dump;
00201 };
00202
00203
00204 /** \} */
00205 /** \} */
00206 #endif // GNSS_SDR_GLO_NASS_L1_CA_DLL_PLL_C_AID_TRACKING_SC_H

```

## 11.536 glonass\_l1\_ca\_dll\_pll\_tracking\_cc.h File Reference

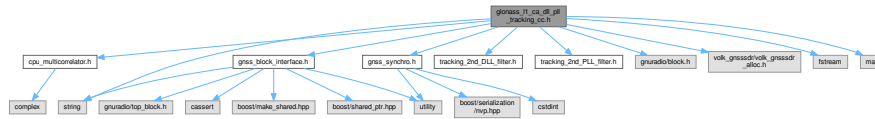
Implementation of a code DLL + carrier PLL tracking block.

```

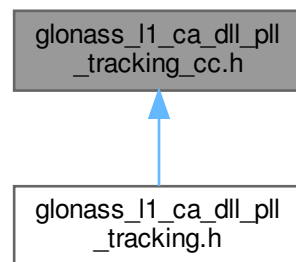
#include "cpu_multicorrelator.h"
#include "gnss_block_interface.h"
#include "gnss_synchro.h"
#include "tracking_2nd_DLL_filter.h"
#include "tracking_2nd_PLL_filter.h"
#include <gnuradio/block.h>
#include <volk_gnssssdr/volk_gnssssdr_alloc.h>
#include <fstream>
#include <map>
#include <string>

```

Include dependency graph for glonass\_l1\_ca\_dll\_pll\_tracking\_cc.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Glonass\\_L1\\_Ca\\_Dll\\_Pll\\_Tracking\\_cc](#)  
*This class implements a DLL + PLL tracking loop block.*

## Typedefs

- using [glonass\\_l1\\_ca\\_dll\\_pll\\_tracking\\_cc\\_sptr](#) = `gnss_shared_ptr<Glonass\_L1\_Ca\_Dll\_Pll\_Tracking\_cc>`

## Functions

- `glonass_l1_ca_dll_pll_tracking_cc_sptr` [glonass\\_l1\\_ca\\_dll\\_pll\\_make\\_tracking\\_cc](#) (int64\_t fs\_in, uint32\_t vector\_length, bool dump, const std::string &dump\_filename, float pll\_bw\_hz, float dll\_bw\_hz, float early\_↔late\_space\_chips)

## 11.536.1 Detailed Description

Implementation of a code DLL + carrier PLL tracking block.

### Author

Gabriel Araujo, 2017. [gabriel.araujo.5000@gmail.com](mailto:gabriel.araujo.5000@gmail.com)

Luis Esteve, 2017. [luis@epsilon-formacion.com](mailto:luis@epsilon-formacion.com)

Damian Miralles, 2017. [dmiralles2009@gmail.com](mailto:dmiralles2009@gmail.com)

Code DLL + carrier PLL according to the algorithms described in: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen, A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach, Birkha user, 2007  
GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [glonass\\_l1\\_ca\\_dll\\_pll\\_tracking\\_cc.h](#).

## 11.537 glonass\_l1\_ca\_dll\_pll\_tracking\_cc.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file glonass_l1_ca_dll_pll_tracking_cc.h
00003  * \brief Implementation of a code DLL + carrier PLL tracking block
00004  * \author Gabriel Araujo, 2017. gabriel.araujo.5000(at)gmail.com
00005  * \author Luis Esteve, 2017. luis(at)epsilon-formacion.com
00006  * \author Damian Miralles, 2017. dmiralles2009(at)gmail.com
00007  *
00008  *
00009  * Code DLL + carrier PLL according to the algorithms described in:
00010  * K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen,
00011  * A Software-Defined GPS and Galileo Receiver. A Single-Frequency
00012  * Approach, Birkha user, 2007
00013  *
00014  * -----
00015  *
00016  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00017  * This file is part of GNSS-SDR.
00018  *
00019  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00020  * SPDX-License-Identifier: GPL-3.0-or-later
00021  *
00022  * -----
00023  */
00024
00025 #ifndef GNSS_SDR_GLOPASS_L1_CA_DLL_PLL_TRACKING_CC_H
00026 #define GNSS_SDR_GLOPASS_L1_CA_DLL_PLL_TRACKING_CC_H
00027
00028 #include "cpu_multicorrelator.h"
00029 #include "gnss_block_interface.h"
00030 #include "gnss_synchro.h"
00031 #include "tracking_2nd_DLL_filter.h"
00032 #include "tracking_2nd_PLL_filter.h"
00033 #include <gnuradio/block.h>
00034 #include <volk_gnss_sdr/volk_gnss_sdr_alloc.h> // for volk_gnss_sdr::vector
00035 #include <fstream>
00036 #include <map>
00037 #include <string>
00038
00039
00040 /** \addtogroup Tracking
00041  * \{ */
00042 /** \addtogroup Tracking_gnuradio_blocks
00043  * \{ */
00044
00045
00046 class Glonass_L1_Ca_Dll_Pll_Tracking_cc;
00047
00048 using glonass_l1_ca_dll_pll_tracking_cc_sptr = gnss_shared_ptr<Glonass_L1_Ca_Dll_Pll_Tracking_cc>;
00049
00050 glonass_l1_ca_dll_pll_tracking_cc_sptr
00051 glonass_l1_ca_dll_pll_make_tracking_cc(
00052     int64_t fs_in, uint32_t vector_length,
00053     bool dump,
00054     const std::string& dump_filename,
00055     float pll_bw_hz,
00056     float dll_bw_hz,
00057     float early_late_space_chips);
00058
00059
00060 /*!
00061  * \brief This class implements a DLL + PLL tracking loop block
00062  */
00063 class Glonass_L1_Ca_Dll_Pll_Tracking_cc : public gr::block
00064 {
00065 public:
00066     ~Glonass_L1_Ca_Dll_Pll_Tracking_cc();
00067
00068     void set_channel(uint32_t channel);
00069     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro);
00070     void start_tracking();
00071
00072     int general_work(int noutput_items, gr_vector_int& ninput_items,
00073         gr_vector_const_void_star& input_items, gr_vector_void_star& output_items);
00074
00075     void forecast(int noutput_items, gr_vector_int& ninput_items_required);
00076
00077 private:
00078     friend glonass_l1_ca_dll_pll_tracking_cc_sptr
00079     glonass_l1_ca_dll_pll_make_tracking_cc(
00080         int64_t fs_in, uint32_t vector_length,
00081         bool dump,
00082         const std::string& dump_filename,
00083         float pll_bw_hz,

```

```

00084         float dll_bw_hz,
00085         float early_late_space_chips);
00086
00087     Glonass_L1_Ca_Dll_Pll_Tracking_cc(
00088         int64_t fs_in, uint32_t vector_length,
00089         bool dump,
00090         const std::string& dump_filename,
00091         float pll_bw_hz,
00092         float dll_bw_hz,
00093         float early_late_space_chips);
00094
00095     void check_carrier_phase_coherent_initialization();
00096
00097     int32_t save_matfile() const;
00098
00099     volk_gnssssdr::vector<gr_complex> d_ca_code;
00100     volk_gnssssdr::vector<float> d_local_code_shift_chips;
00101     volk_gnssssdr::vector<gr_complex> d_correlator_outs;
00102     volk_gnssssdr::vector<gr_complex> d_Prompt_buffer;
00103
00104     Cpu_Multicorrelator multicorrelator_cpu;
00105
00106     // PLL and DLL filter library
00107     Tracking_2nd_DLL_filter d_code_loop_filter;
00108     Tracking_2nd_PLL_filter d_carrier_loop_filter;
00109
00110     Gnss_Synchro* d_acquisition_gnss_synchro;
00111
00112     // file dump
00113     std::string d_dump_filename;
00114     std::ofstream d_dump_file;
00115
00116     std::map<std::string, std::string> systemName;
00117     std::string sys;
00118
00119     // tracking configuration vars
00120     int64_t d_fs_in;
00121     int64_t d_glonass_freq_ch;
00122     double d_early_late_spc_chips;
00123     uint32_t d_vector_length;
00124     uint32_t d_channel;
00125
00126     // remaining code phase and carrier phase between tracking loops
00127     double d_rem_code_phase_samples;
00128     double d_rem_code_phase_chips;
00129     float d_rem_carr_phase_rad;
00130
00131     // acquisition
00132     double d_acq_code_phase_samples;
00133     double d_acq_carrier_doppler_hz;
00134
00135     // tracking vars
00136     double d_code_freq_chips;
00137     double d_code_phase_step_chips;
00138     double d_carrier_doppler_hz;
00139     double d_carrier_doppler_phase_step_rad;
00140     double d_carrier_frequency_hz;
00141     double d_carrier_phase_step_rad;
00142     double d_acc_carrier_phase_rad;
00143     double d_code_phase_samples;
00144
00145     // correlator
00146     int32_t d_n_correlator_taps;
00147
00148     // PRN period in samples
00149     int32_t d_current_prn_length_samples;
00150
00151     // processing samples counters
00152     uint64_t d_sample_counter;
00153     uint64_t d_acq_sample_stamp;
00154
00155     // CN0 estimation and lock detector
00156     double d_carrier_lock_test;
00157     double d_CN0_SNV_dB_Hz;
00158     double d_carrier_lock_threshold;
00159     int32_t d_carrier_lock_fail_counter;
00160     int32_t d_cn0_estimation_counter;
00161
00162     // control vars
00163     bool d_enable_tracking;
00164     bool d_pull_in;
00165     bool d_acc_carrier_phase_initialized;
00166
00167     bool d_dump;
00168 };
00169
00170

```



float pll\_bw\_narrow\_hz, float dll\_bw\_narrow\_hz, int32\_t extend\_correlation\_ms, float early\_late\_space\_↵  
chips)

### 11.538.1 Detailed Description

Implementation of a code DLL + carrier PLL tracking block.

Author

Damian Miralles, 2018. dmiralles2009(at)gmail.com

Code DLL + carrier PLL according to the algorithms described in: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen, A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach, Birkha user, 2007  
GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [glonass\\_l2\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_cc.h](#).

## 11.539 glonass\_l2\_ca\_dll\_pll\_c\_aid\_tracking\_cc.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file glonass_l2_ca_dll_pll_c_aid_tracking_cc.h
00003  * \brief Implementation of a code DLL + carrier PLL tracking block
00004  * \author Damian Miralles, 2018. dmiralles2009(at)gmail.com
00005  *
00006  *
00007  * Code DLL + carrier PLL according to the algorithms described in:
00008  * K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen,
00009  * A Software-Defined GPS and Galileo Receiver. A Single-Frequency
00010  * Approach, Birkha user, 2007
00011  *
00012  * -----
00013  *
00014  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00015  * This file is part of GNSS-SDR.
00016  *
00017  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00018  * SPDX-License-Identifier: GPL-3.0-or-later
00019  *
00020  * -----
00021  */
00022
00023 #ifndef GNSS_SDR_GLO_NASS_L2_CA_DLL_PLL_C_AID_TRACKING_CC_H
00024 #define GNSS_SDR_GLO_NASS_L2_CA_DLL_PLL_C_AID_TRACKING_CC_H
00025
00026 #include "cpu_multicorrelator.h"
00027 #include "gnss_block_interface.h"
00028 #include "gnss_synchro.h"
00029 #include "tracking_2nd_DLL_filter.h"
00030 #include "tracking_FLL_PLL_filter.h"
00031 #include <gnuradio/block.h>
00032 #include <pmt/pmt.h>
00033 #include <volk_gnss_sdr/volk_gnss_sdr_alloc.h> // for volk_gnss_sdr::vector
00034 #include <deque>
00035 #include <fstream>
00036 #include <map>
00037 #include <string>
00038
00039 /** \addtogroup Tracking
00040  * \{ */
00041 /** \addtogroup Tracking_gnuradio_blocks
00042  * \{ */
00043
00044
00045 class glonass_l2_ca_dll_pll_c_aid_tracking_cc;
00046
00047 using glonass_l2_ca_dll_pll_c_aid_tracking_cc_sptr =
00048     gnss_shared_ptr<glonass_l2_ca_dll_pll_c_aid_tracking_cc>;
00049
00049 glonass_l2_ca_dll_pll_c_aid_tracking_cc_sptr
00050 glonass_l2_ca_dll_pll_c_aid_make_tracking_cc(
00051     int64_t fs_in, uint32_t vector_length,
00052     bool dump,
00053     const std::string& dump_filename,
00054     float pll_bw_hz,
00055     float dll_bw_hz,
00056     float pll_bw_narrow_hz,
00057     float dll_bw_narrow_hz,
00058     int32_t extend_correlation_ms,
```

```

00059     float early_late_space_chips);
00060
00061
00062  /*!
00063   * \brief This class implements a DLL + PLL tracking loop block
00064   */
00065  class glonass_l2_ca_dll_pll_c_aid_tracking_cc : public gr::block
00066  {
00067  public:
00068      ~glonass_l2_ca_dll_pll_c_aid_tracking_cc();
00069
00070      void set_channel(uint32_t channel);
00071      void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro);
00072      void start_tracking();
00073
00074      int general_work(int noutput_items, gr_vector_int& ninput_items,
00075                      gr_vector_const_void_star& input_items, gr_vector_void_star& output_items);
00076
00077      void forecast(int noutput_items, gr_vector_int& ninput_items_required);
00078
00079  private:
00080      friend glonass_l2_ca_dll_pll_c_aid_tracking_cc_sptr
00081      glonass_l2_ca_dll_pll_c_aid_make_tracking_cc(
00082          int64_t fs_in, uint32_t vector_length,
00083          bool dump,
00084          const std::string& dump_filename,
00085          float pll_bw_hz,
00086          float dll_bw_hz,
00087          float pll_bw_narrow_hz,
00088          float dll_bw_narrow_hz,
00089          int32_t extend_correlation_ms,
00090          float early_late_space_chips);
00091
00092      glonass_l2_ca_dll_pll_c_aid_tracking_cc(
00093          int64_t fs_in, uint32_t vector_length,
00094          bool dump,
00095          const std::string& dump_filename,
00096          float pll_bw_hz,
00097          float dll_bw_hz,
00098          float pll_bw_narrow_hz,
00099          float dll_bw_narrow_hz,
00100          int32_t extend_correlation_ms,
00101          float early_late_space_chips);
00102
00103      void msg_handler_preamble_index(const pmt::pmt_t& msg);
00104
00105      void check_carrier_phase_coherent_initialization();
00106
00107      int32_t save_matfile() const;
00108
00109      volk_gnssssdr::vector<gr_complex> d_ca_code;
00110      volk_gnssssdr::vector<float> d_local_code_shift_chips;
00111      volk_gnssssdr::vector<gr_complex> d_correlator_outs;
00112      volk_gnssssdr::vector<gr_complex> d_Prompt_buffer;
00113
00114      Cpu_Multicorrelator multicorrelator_cpu;
00115
00116      // PLL and DLL filter library
00117      Tracking_2nd_DLL_filter d_code_loop_filter;
00118      Tracking_2nd_DLL_filter d_code_loop_filter;
00119      Tracking_FLL_PLL_filter d_carrier_loop_filter;
00120
00121      // symbol history to detect bit transition
00122      std::deque<gr_complex> d_E_history;
00123      std::deque<gr_complex> d_P_history;
00124      std::deque<gr_complex> d_L_history;
00125
00126      // file dump
00127      std::string d_dump_filename;
00128      std::ofstream d_dump_file;
00129
00130      std::map<std::string, std::string> systemName;
00131      std::string sys;
00132
00133      Gnss_Synchro* d_acquisition_gnss_synchro;
00134
00135      // tracking configuration vars
00136      int64_t d_fs_in;
00137      double d_glonass_freq_ch;
00138      double d_early_late_spc_chips;
00139      uint32_t d_vector_length;
00140      uint32_t d_channel;
00141      int32_t d_n_correlator_taps;
00142
00143      // remaining code phase and carrier phase between tracking loops
00144      double d_rem_code_phase_samples;
00145      double d_rem_code_phase_chips;

```



```

00146     double d_rem_carrier_phase_rad;
00147     int32_t d_rem_code_phase_integer_samples;
00148
00149     // acquisition
00150     double d_acq_code_phase_samples;
00151     double d_acq_carrier_doppler_hz;
00152
00153     // tracking vars
00154     float d_dll_bw_hz;
00155     float d_pll_bw_hz;
00156     float d_dll_bw_narrow_hz;
00157     float d_pll_bw_narrow_hz;
00158     double d_code_freq_chips;
00159     double d_code_phase_step_chips;
00160     double d_carrier_doppler_hz;
00161     double d_carrier_frequency_hz;
00162     double d_carrier_frequency_old_hz;
00163     double d_carrier_phase_step_rad;
00164     double d_acc_carrier_phase_cycles;
00165     double d_code_phase_samples;
00166     double d_pll_to_dll_assist_secs_Ti;
00167     double d_code_error_chips_Ti;
00168     double d_code_error_filt_chips_s;
00169     double d_code_error_filt_chips_Ti;
00170     double d_carr_phase_error_secs_Ti;
00171     double d_preamble_timestamp_samples;
00172     int32_t d_extend_correlation_ms;
00173
00174     // Integration period in samples
00175     int32_t d_correlation_length_samples;
00176
00177     // processing samples counters
00178     uint64_t d_sample_counter;
00179     uint64_t d_acq_sample_stamp;
00180
00181     // CN0 estimation and lock detector
00182     int32_t d_cn0_estimation_counter;
00183     double d_carrier_lock_test;
00184     double d_CN0_SNV_dB_Hz;
00185     double d_carrier_lock_threshold;
00186     int32_t d_carrier_lock_fail_counter;
00187
00188     bool d_enable_extended_integration;
00189     bool d_preamble_synchronized;
00190
00191     // control vars
00192     bool d_enable_tracking;
00193     bool d_pull_in;
00194     bool d_acc_carrier_phase_initialized;
00195
00196     bool d_dump;
00197 };
00198
00199
00200 /** \} */
00201 /** \} */
00202 #endif // GNSS_SDR_GLOPASS_L1_CA_DLL_PLL_C_AID_TRACKING_CC_H

```

## 11.540 glonass\_l2\_ca\_dll\_pll\_c\_aid\_tracking\_sc.h File Reference

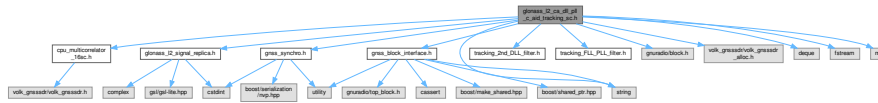
Implementation of a code DLL + carrier PLL tracking block.

```

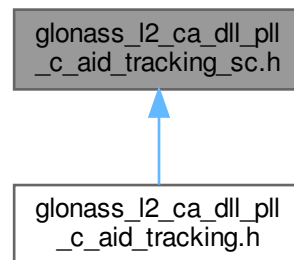
#include "cpu_multicorrelator_16sc.h"
#include "glonass_l2_signal_replica.h"
#include "gnss_block_interface.h"
#include "gnss_synchro.h"
#include "tracking_2nd_DLL_filter.h"
#include "tracking_FLL_PLL_filter.h"
#include <gnuradio/block.h>
#include <volk_gnssssdr/volk_gnssssdr_alloc.h>
#include <deque>
#include <fstream>
#include <map>
#include <string>

```

Include dependency graph for `glonass_l2_ca_dll_pll_c_aid_tracking_sc.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [glonass\\_l2\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_sc](#)  
*This class implements a DLL + PLL tracking loop block.*

## Typedefs

- using [glonass\\_l2\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_sc\\_sptr](#) = `gnss_shared_ptr<glonass_l2_ca_dll_pll_c_aid_tracking_sc>`

## Functions

- `glonass_l2_ca_dll_pll_c_aid_tracking_sc_sptr` **glonass\_l2\_ca\_dll\_pll\_c\_aid\_make\_tracking\_sc** (`int64_t` `fs_in`, `uint32_t` `vector_length`, `bool` `dump`, `const std::string&` `dump_filename`, `float` `pll_bw_hz`, `float` `dll_bw_hz`, `float` `pll_bw_narrow_hz`, `float` `dll_bw_narrow_hz`, `int32_t` `extend_correlation_ms`, `float` `early_late_space_chips`)

### 11.540.1 Detailed Description

Implementation of a code DLL + carrier PLL tracking block.

#### Author

Damian Miralles, 2018. [dmiralles2009\(at\)gmail.com](mailto:dmiralles2009(at)gmail.com)

Code DLL + carrier PLL according to the algorithms described in: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen, A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach, Birkha user, 2007  
GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [glonass\\_l2\\_ca\\_dll\\_pll\\_c\\_aid\\_tracking\\_sc.h](#).

## 11.541 glonass\_l2\_ca\_dll\_pll\_c\_aid\_tracking\_sc.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file glonass_l2_ca_dll_pll_c_aid_tracking_sc.h
00003  * \brief Implementation of a code DLL + carrier PLL tracking block
00004  * \author Damian Miralles, 2018. dmiralles2009(at)gmail.com
00005  *
00006  *
00007  * Code DLL + carrier PLL according to the algorithms described in:
00008  * K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen,
00009  * A Software-Defined GPS and Galileo Receiver. A Single-Frequency
00010  * Approach, Birkha user, 2007
00011  *
00012  * -----
00013  *
00014  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00015  * This file is part of GNSS-SDR.
00016  *
00017  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00018  * SPDX-License-Identifier: GPL-3.0-or-later
00019  *
00020  * -----
00021  */
00022
00023 #ifndef GNSS_SDR_GLONASS_L2_CA_DLL_PLL_C_AID_TRACKING_SC_H
00024 #define GNSS_SDR_GLONASS_L2_CA_DLL_PLL_C_AID_TRACKING_SC_H
00025
00026 #include "cpu_multicorrelator_16sc.h"
00027 #include "glonass_l2_signal_replica.h"
00028 #include "gnss_block_interface.h"
00029 #include "gnss_synchro.h"
00030 #include "tracking_2nd_DLL_filter.h"
00031 #include "tracking_FLL_PLL_filter.h"
00032 #include <gnuradio/block.h>
00033 #include <volk_gnssssdr/volk_gnssssdr_alloc.h> // for volk_gnssssdr::vector
00034 #include <deque>
00035 #include <fstream>
00036 #include <map>
00037 #include <string>
00038
00039 /** \addtogroup Tracking
00040  * \{ */
00041 /** \addtogroup Tracking_gnuradio_blocks
00042  * \{ */
00043
00044
00045 class glonass_l2_ca_dll_pll_c_aid_tracking_sc;
00046
00047 using glonass_l2_ca_dll_pll_c_aid_tracking_sc_sptr =
    gnss_shared_ptr<glonass_l2_ca_dll_pll_c_aid_tracking_sc>;
00048
00049 glonass_l2_ca_dll_pll_c_aid_tracking_sc_sptr
00050 glonass_l2_ca_dll_pll_c_aid_make_tracking_sc(
00051     int64_t fs_in, uint32_t vector_length,
00052     bool dump,
00053     const std::string& dump_filename,
00054     float pll_bw_hz,
00055     float dll_bw_hz,
00056     float pll_bw_narrow_hz,
00057     float dll_bw_narrow_hz,
00058     int32_t extend_correlation_ms,
00059     float early_late_space_chips);
00060
00061
00062 /*!
00063  * \brief This class implements a DLL + PLL tracking loop block
00064  */
00065 class glonass_l2_ca_dll_pll_c_aid_tracking_sc : public gr::block
00066 {
00067 public:
00068     ~glonass_l2_ca_dll_pll_c_aid_tracking_sc();
00069
00070     void set_channel(uint32_t channel);
00071     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro);
00072     void start_tracking();
00073
00074     int general_work(int noutput_items, gr_vector_int& ninput_items,
00075         gr_vector_const_void_star& input_items, gr_vector_void_star& output_items);
00076
00077     void forecast(int noutput_items, gr_vector_int& ninput_items_required);
00078
00079 private:
00080     friend glonass_l2_ca_dll_pll_c_aid_tracking_sc_sptr
00081     glonass_l2_ca_dll_pll_c_aid_make_tracking_sc(
00082         int64_t fs_in, uint32_t vector_length,
```

```

00083         bool dump,
00084         const std::string& dump_filename,
00085         float pll_bw_hz,
00086         float dll_bw_hz,
00087         float pll_bw_narrow_hz,
00088         float dll_bw_narrow_hz,
00089         int32_t extend_correlation_ms,
00090         float early_late_space_chips);
00091
00092     glonass_l2_ca_dll_pll_c_aid_tracking_sc(
00093         int64_t fs_in, uint32_t vector_length,
00094         bool dump,
00095         const std::string& dump_filename,
00096         float pll_bw_hz,
00097         float dll_bw_hz,
00098         float pll_bw_narrow_hz,
00099         float dll_bw_narrow_hz,
00100         int32_t extend_correlation_ms,
00101         float early_late_space_chips);
00102
00103     void msg_handler_preamble_index(const pmt::pmt_t& msg);
00104
00105     void check_carrier_phase_coherent_initialization();
00106
00107     int32_t save_matfile() const;
00108
00109     volk_gnssssdr::vector<gr_complex> d_ca_code;
00110     volk_gnssssdr::vector<gr_complex> d_Prompt_buffer;
00111     volk_gnssssdr::vector<float> d_local_code_shift_chips;
00112     volk_gnssssdr::vector<lv_16sc_t> d_ca_code_16sc;
00113     volk_gnssssdr::vector<lv_16sc_t> d_correlator_outs_16sc;
00114
00115     Cpu_Multicorrelator_16sc multicorrelator_cpu_16sc;
00116
00117     // PLL and DLL filter library
00118     Tracking_2nd_DLL_filter d_code_loop_filter;
00119     Tracking_FLL_PLL_filter d_carrier_loop_filter;
00120
00121     // symbol history to detect bit transition
00122     std::deque<lv_16sc_t> d_E_history;
00123     std::deque<lv_16sc_t> d_P_history;
00124     std::deque<lv_16sc_t> d_L_history;
00125
00126     // file dump
00127     std::string d_dump_filename;
00128     std::ofstream d_dump_file;
00129
00130     std::map<std::string, std::string> systemName;
00131     std::string sys;
00132
00133     // tracking configuration vars
00134     Gnss_Synchro* d_acquisition_gnss_synchro;
00135     int64_t d_fs_in;
00136     int64_t d_glonass_freq_ch;
00137     double d_early_late_spc_chips;
00138     uint32_t d_vector_length;
00139     uint32_t d_channel;
00140     int32_t d_n_correlator_taps;
00141
00142     // remaining code phase and carrier phase between tracking loops
00143     double d_rem_code_phase_samples;
00144     double d_rem_code_phase_chips;
00145     double d_rem_carrier_phase_rad;
00146     int32_t d_rem_code_phase_integer_samples;
00147
00148     // acquisition
00149     double d_acq_code_phase_samples;
00150     double d_acq_carrier_doppler_hz;
00151
00152     // tracking vars
00153     float d_dll_bw_hz;
00154     float d_pll_bw_hz;
00155     float d_dll_bw_narrow_hz;
00156     float d_pll_bw_narrow_hz;
00157     double d_code_freq_chips;
00158     double d_code_phase_step_chips;
00159     double d_carrier_doppler_hz;
00160     double d_carrier_frequency_hz;
00161     double d_carrier_frequency_old_hz;
00162     double d_carrier_phase_step_rad;
00163     double d_acc_carrier_phase_cycles;
00164     double d_code_phase_samples;
00165     double d_pll_to_dll_assist_secs_Ti;
00166     double d_carr_phase_error_secs_Ti;
00167     double d_code_error_chips_Ti;
00168     double d_preamble_timestamp_samples;
00169     double d_code_error_filt_chips_s;

```

```

00170     double d_code_error_filt_chips_Ti;
00171     int32_t d_extend_correlation_ms;
00172
00173     // Integration period in samples
00174     int32_t d_correlation_length_samples;
00175
00176     // processing samples counters
00177     uint64_t d_sample_counter;
00178     uint64_t d_acq_sample_stamp;
00179
00180     // CN0 estimation and lock detector
00181     double d_carrier_lock_test;
00182     double d_CN0_SNV_dB_Hz;
00183     double d_carrier_lock_threshold;
00184     int32_t d_carrier_lock_fail_counter;
00185     int32_t d_cn0_estimation_counter;
00186
00187     bool d_enable_extended_integration;
00188     bool d_preamble_synchronized;
00189
00190     // control vars
00191     bool d_enable_tracking;
00192     bool d_pull_in;
00193     bool d_acc_carrier_phase_initialized;
00194
00195     bool d_dump;
00196 };
00197
00198
00199 /** \} */
00200 /** \} */
00201 #endif // GNSS_SDR_GLO_NASS_L2_CA_DLL_PLL_C_AID_TRACKING_SC_H

```

## 11.542 glonass\_l2\_ca\_dll\_pll\_tracking\_cc.h File Reference

Implementation of a code DLL + carrier PLL tracking block.

```

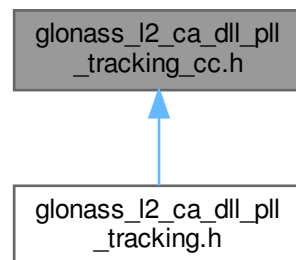
#include "cpu_multicorrelator.h"
#include "gnss_block_interface.h"
#include "gnss_synchro.h"
#include "tracking_2nd_DLL_filter.h"
#include "tracking_2nd_PLL_filter.h"
#include <gnuradio/block.h>
#include <volk_gnssssdr/volk_gnssssdr_alloc.h>
#include <fstream>
#include <map>
#include <string>

```

Include dependency graph for glonass\_l2\_ca\_dll\_pll\_tracking\_cc.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Glonass\\_L2\\_Ca\\_Dll\\_Pll\\_Tracking\\_cc](#)  
*This class implements a DLL + PLL tracking loop block.*

## Typedefs

- using [glonass\\_l2\\_ca\\_dll\\_pll\\_tracking\\_cc\\_sptr](#) = gnss\_shared\_ptr<[Glonass\\_L2\\_Ca\\_Dll\\_Pll\\_Tracking\\_cc](#)>

## Functions

- [glonass\\_l2\\_ca\\_dll\\_pll\\_tracking\\_cc\\_sptr glonass\\_l2\\_ca\\_dll\\_pll\\_make\\_tracking\\_cc](#) (int64\_t fs\_in, uint32\_t vector\_length, bool dump, const std::string &dump\_filename, float pll\_bw\_hz, float dll\_bw\_hz, float early\_↵late\_space\_chips)

### 11.542.1 Detailed Description

Implementation of a code DLL + carrier PLL tracking block.

#### Author

Damian Miralles, 2018. [dmiralles2009\(at\)gmail.com](mailto:dmiralles2009(at)gmail.com)

Code DLL + carrier PLL according to the algorithms described in: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen, A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach, Birkha user, 2007  
GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [glonass\\_l2\\_ca\\_dll\\_pll\\_tracking\\_cc.h](#).

### 11.543 glonass\_l2\_ca\_dll\_pll\_tracking\_cc.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file glonass_l2_ca_dll_pll_tracking_cc.h
00003  * \brief Implementation of a code DLL + carrier PLL tracking block
00004  * \author Damian Miralles, 2018. dmiralles2009\(at\)gmail.com
00005  *
00006  *
00007  * Code DLL + carrier PLL according to the algorithms described in:
00008  * K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen,
00009  * A Software-Defined GPS and Galileo Receiver. A Single-Frequency
00010  * Approach, Birkha user, 2007
00011  *
00012  * -----
00013  *

```

```

00014  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00015  * This file is part of GNSS-SDR.
00016  *
00017  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00018  * SPDX-License-Identifier: GPL-3.0-or-later
00019  *
00020  * -----
00021  */
00022
00023 #ifndef GNSS_SDR_GLOPASS_L2_CA_DLL_PLL_TRACKING_CC_H
00024 #define GNSS_SDR_GLOPASS_L2_CA_DLL_PLL_TRACKING_CC_H
00025
00026 #include "cpu_multicorrelator.h"
00027 #include "gnss_block_interface.h"
00028 #include "gnss_synchro.h"
00029 #include "tracking_2nd_DLL_filter.h"
00030 #include "tracking_2nd_PLL_filter.h"
00031 #include <gnuradio/block.h>
00032 #include <volk_gnssssdr/volk_gnssssdr_alloc.h> // for volk_gnssssdr::vector
00033 #include <fstream>
00034 #include <map>
00035 #include <string>
00036
00037 /** \addtogroup Tracking
00038  * \{ */
00039 /** \addtogroup Tracking_gnuradio_blocks
00040  * \{ */
00041
00042
00043 class Glonass_L2_Ca_Dll_Pll_Tracking_cc;
00044
00045 using glonass_l2_ca_dll_pll_tracking_cc_sptr = gnss_shared_ptr<Glonass_L2_Ca_Dll_Pll_Tracking_cc>;
00046
00047 glonass_l2_ca_dll_pll_tracking_cc_sptr
00048 glonass_l2_ca_dll_pll_make_tracking_cc(
00049     int64_t fs_in, uint32_t vector_length,
00050     bool dump,
00051     const std::string& dump_filename,
00052     float pll_bw_hz,
00053     float dll_bw_hz,
00054     float early_late_space_chips);
00055
00056
00057 /*!
00058  * \brief This class implements a DLL + PLL tracking loop block
00059  */
00060 class Glonass_L2_Ca_Dll_Pll_Tracking_cc : public gr::block
00061 {
00062 public:
00063     ~Glonass_L2_Ca_Dll_Pll_Tracking_cc();
00064
00065     void set_channel(uint32_t channel);
00066     void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro);
00067     void start_tracking();
00068
00069     int general_work(int noutput_items, gr_vector_int& ninput_items,
00070         gr_vector_const_void_star& input_items, gr_vector_void_star& output_items);
00071
00072     void forecast(int noutput_items, gr_vector_int& ninput_items_required);
00073
00074 private:
00075     friend glonass_l2_ca_dll_pll_tracking_cc_sptr
00076     glonass_l2_ca_dll_pll_make_tracking_cc(
00077         int64_t fs_in, uint32_t vector_length,
00078         bool dump,
00079         const std::string& dump_filename,
00080         float pll_bw_hz,
00081         float dll_bw_hz,
00082         float early_late_space_chips);
00083
00084     Glonass_L2_Ca_Dll_Pll_Tracking_cc(
00085         int64_t fs_in, uint32_t vector_length,
00086         bool dump,
00087         const std::string& dump_filename,
00088         float pll_bw_hz,
00089         float dll_bw_hz,
00090         float early_late_space_chips);
00091
00092     void check_carrier_phase_coherent_initialization();
00093
00094     int32_t save_matfile() const;
00095
00096     volk_gnssssdr::vector<gr_complex> d_ca_code;
00097     volk_gnssssdr::vector<gr_complex> d_correlator_outs;
00098     volk_gnssssdr::vector<gr_complex> d_Prompt_buffer;
00099     volk_gnssssdr::vector<float> d_local_code_shift_chips;
00100

```

```

00101     Cpu_Multicorrelator multicorrelator_cpu;
00102
00103     // PLL and DLL filter library
00104     Tracking_2nd_DLL_filter d_code_loop_filter;
00105     Tracking_2nd_PLL_filter d_carrier_loop_filter;
00106
00107     // file dump
00108     std::string d_dump_filename;
00109     std::ofstream d_dump_file;
00110
00111     std::map<std::string, std::string> systemName;
00112     std::string sys;
00113
00114     Gnss_Synchro* d_acquisition_gnss_synchro;
00115
00116     // tracking configuration vars
00117     int64_t d_fs_in;
00118     int64_t d_glonass_freq_ch;
00119     double d_early_late_spc_chips;
00120     uint32_t d_vector_length;
00121     uint32_t d_channel;
00122
00123     // remaining code phase and carrier phase between tracking loops
00124     double d_rem_code_phase_samples;
00125     double d_rem_code_phase_chips;
00126     float d_rem_carr_phase_rad;
00127
00128     // acquisition
00129     double d_acq_code_phase_samples;
00130     double d_acq_carrier_doppler_hz;
00131
00132     // correlator
00133     int32_t d_n_correlator_taps;
00134
00135     // tracking vars
00136     double d_code_freq_chips;
00137     double d_code_phase_step_chips;
00138     double d_carrier_doppler_hz;
00139     double d_carrier_doppler_phase_step_rad;
00140     double d_carrier_frequency_hz;
00141     double d_carrier_phase_step_rad;
00142     double d_acc_carrier_phase_rad;
00143     double d_code_phase_samples;
00144
00145     // PRN period in samples
00146     int32_t d_current_prn_length_samples;
00147
00148     // processing samples counters
00149     uint64_t d_sample_counter;
00150     uint64_t d_acq_sample_stamp;
00151
00152     // CN0 estimation and lock detector
00153     double d_carrier_lock_test;
00154     double d_CN0_SNV_dB_Hz;
00155     double d_carrier_lock_threshold;
00156     int32_t d_cn0_estimation_counter;
00157     int32_t d_carrier_lock_fail_counter;
00158
00159     // control vars
00160     bool d_enable_tracking;
00161     bool d_pull_in;
00162     bool d_acc_carrier_phase_initialized;
00163
00164     bool d_dump;
00165 };
00166
00167 /** */
00168 /** */
00169 /** */
00170 #endif // GNSS_SDR_GLO_NASS_L2_CA_DLL_PLL_TRACKING_CC_H

```

## 11.544 gps\_l1\_ca\_dll\_pll\_tracking\_gpu\_cc.h File Reference

Implementation of a code DLL + carrier PLL tracking block, GPU ACCELERATED.

```

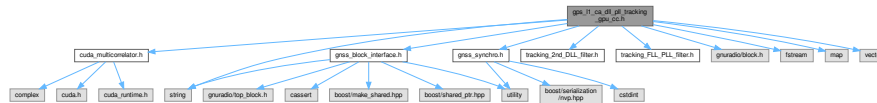
#include "cuda_multicorrelator.h"
#include "gnss_block_interface.h"
#include "gnss_synchro.h"
#include "tracking_2nd_DLL_filter.h"
#include "tracking_FLL_PLL_filter.h"
#include <gnuradio/block.h>
#include <fstream>

```

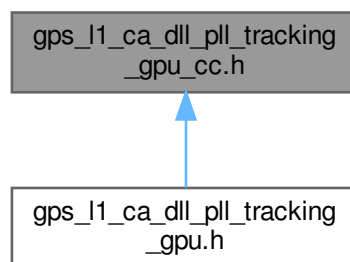


```
#include <map>
#include <string>
#include <vector>
```

Include dependency graph for `gps_l1_ca_dll_pll_tracking_gpu_cc.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Gps\\_L1\\_Ca\\_Dll\\_Pll\\_Tracking\\_GPU\\_cc](#)  
*This class implements a DLL + PLL tracking loop block.*

## Typedefs

- using [gps\\_l1\\_ca\\_dll\\_pll\\_tracking\\_gpu\\_cc\\_sptr](#) = `gnss_shared_ptr<Gps_L1_Ca_Dll_Pll_Tracking_GPU_cc>`

## Functions

- `gps_l1_ca_dll_pll_tracking_gpu_cc_sptr` [gps\\_l1\\_ca\\_dll\\_pll\\_make\\_tracking\\_gpu\\_cc](#) (`int64_t` fs\_in, `uint32_t` vector\_length, `bool` dump, `std::string` dump\_filename, `float` pll\_bw\_hz, `float` dll\_bw\_hz, `float` early\_↔late\_space\_chips)

### 11.544.1 Detailed Description

Implementation of a code DLL + carrier PLL tracking block, GPU ACCELERATED.

#### Author

Javier Arribas, 2015. [jarribas@cttc.es](mailto:jarribas@cttc.es)

Code DLL + carrier PLL according to the algorithms described in: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen, A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach, Birkhauser, 2007  
GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [gps\\_l1\\_ca\\_dll\\_pll\\_tracking\\_gpu\\_cc.h](#).

## 11.545 gps\_l1\_ca\_dll\_pll\_tracking\_gpu\_cc.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file gps_l1_ca_dll_pll_tracking_gpu_cc.h
00003  * \brief Implementation of a code DLL + carrier PLL tracking block, GPU ACCELERATED
00004  * \author Javier Arribas, 2015. jarribas(at)cttc.es
00005  *
00006  * Code DLL + carrier PLL according to the algorithms described in:
00007  * K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen,
00008  * A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach,
00009  * Birkhauser, 2007
00010  *
00011  * -----
00012  *
00013  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00014  * This file is part of GNSS-SDR.
00015  *
00016  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00017  * SPDX-License-Identifier: GPL-3.0-or-later
00018  *
00019  * -----
00020  */
00021
00022 #ifndef GNSS_SDR_GPS_L1_CA_DLL_PLL_TRACKING_GPU_CC_H
00023 #define GNSS_SDR_GPS_L1_CA_DLL_PLL_TRACKING_GPU_CC_H
00024
00025 #include "cuda_multicorrelator.h"
00026 #include "gnss_block_interface.h"
00027 #include "gnss_synchro.h"
00028 #include "tracking_2nd_DLL_filter.h"
00029 #include "tracking_FLL_PLL_filter.h"
00030 #include <gnuradio/block.h>
00031 #include <fstream>
00032 #include <map>
00033 #include <string>
00034 #include <vector>
00035
00036 /** \addtogroup Tracking
00037  * \{ */
00038 /** \addtogroup Tracking_gnuradio_blocks
00039  * \{ */
00040
00041
00042 class Gps_L1_Ca_Dll_Pll_Tracking_GPU_cc;
00043
00044 using gps_l1_ca_dll_pll_tracking_gpu_cc_sptr = gnss_shared_ptr<Gps_L1_Ca_Dll_Pll_Tracking_GPU_cc>;
00045
00046 gps_l1_ca_dll_pll_tracking_gpu_cc_sptr
00047 gps_l1_ca_dll_pll_make_tracking_gpu_cc(
00048     int64_t fs_in,
00049     uint32_t vector_length,
00050     bool dump,
00051     std::string dump_filename,
00052     float pll_bw_hz,
00053     float dll_bw_hz,
00054     float early_late_space_chips);
00055
00056
00057 /*!
00058 * \brief This class implements a DLL + PLL tracking loop block
00059 */
00060 class Gps_L1_Ca_Dll_Pll_Tracking_GPU_cc : public gr::block
00061 {
00062 public:
00063     ~Gps_L1_Ca_Dll_Pll_Tracking_GPU_cc();
00064
00065     void set_channel(uint32_t channel);
00066     void set_gnss_synchro(Gnss_Synchro *p_gnss_synchro);
00067     void start_tracking();
00068
00069     int general_work(int noutput_items, gr_vector_int &ninput_items,
00070         gr_vector_const_void_star &input_items, gr_vector_void_star &output_items);
00071
00072     void forecast(int noutput_items, gr_vector_int &ninput_items_required);
00073 private:
00074     friend gps_l1_ca_dll_pll_tracking_gpu_cc_sptr
00075     gps_l1_ca_dll_pll_make_tracking_gpu_cc(
00076         int64_t fs_in,
00077         uint32_t vector_length,
00078         bool dump,
00079         std::string dump_filename,
00080         float pll_bw_hz,
00081         float dll_bw_hz,
00082         float early_late_space_chips);
00083

```

```

00084
00085     Gps_L1_Ca_Dll_Pll_Tracking_GPU_cc(
00086         int64_t fs_in,
00087         uint32_t vector_length,
00088         bool dump,
00089         std::string dump_filename,
00090         float pll_bw_hz,
00091         float dll_bw_hz,
00092         float early_late_space_chips);
00093 void update_local_code();
00094 void update_local_carrier();
00095 void check_carrier_phase_coherent_initialization();
00096
00097 // PLL and DLL filter library
00098 Tracking_2nd_DLL_filter d_code_loop_filter;
00099 Tracking_FLL_PLL_filter d_carrier_loop_filter;
00100
00101 Gnss_Synchro *d_acquisition_gnss_synchro;
00102
00103 std::vector<gr_complex> d_Prompt_buffer;
00104
00105 // file dump
00106 std::string d_dump_filename;
00107 std::ofstream d_dump_file;
00108
00109 std::map<std::string, std::string> systemName;
00110 std::string sys;
00111
00112 // tracking configuration vars
00113 int64_t d_if_freq;
00114 int64_t d_fs_in;
00115 double d_early_late_spc_chips;
00116 uint32_t d_vector_length;
00117 uint32_t d_channel;
00118 int32_t d_n_correlator_taps;
00119
00120 // GPU HOST PINNED MEMORY IN/OUT VECTORS
00121 cuda_multicorrelator *multicorrelator_gpu;
00122 gr_complex *in_gpu;
00123 gr_complex *d_correlator_outs;
00124 gr_complex *d_ca_code;
00125 float *d_local_code_shift_chips;
00126
00127 gr_complex *d_Early;
00128 gr_complex *d_Prompt;
00129 gr_complex *d_Late;
00130
00131 // remaining code phase and carrier phase between tracking loops
00132 double d_rem_code_phase_samples;
00133 double d_rem_code_phase_chips;
00134 double d_rem_carrier_phase_rad;
00135
00136 // acquisition
00137 double d_acq_code_phase_samples;
00138 double d_acq_carrier_doppler_hz;
00139
00140 // tracking vars
00141 double d_code_freq_chips;
00142 double d_code_phase_step_chips;
00143 double d_carrier_doppler_hz;
00144 double d_carrier_phase_step_rad;
00145 double d_acc_carrier_phase_cycles;
00146 double d_code_phase_samples;
00147 double d_pll_to_dll_assist_secs_Ti;
00148
00149 // Integration period in samples
00150 int32_t d_correlation_length_samples;
00151
00152 // processing samples counters
00153 uint64_t d_sample_counter;
00154 uint64_t d_acq_sample_stamp;
00155
00156 // CN0 estimation and lock detector
00157 double d_carrier_lock_test;
00158 double d_CN0_SNV_dB_Hz;
00159 double d_carrier_lock_threshold;
00160 int32_t d_carrier_lock_fail_counter;
00161 int32_t d_cn0_estimation_counter;
00162
00163 // control vars
00164 bool d_acc_carrier_phase_initialized;
00165 bool d_enable_tracking;
00166 bool d_pull_in;
00167 bool d_dump;
00168 };
00169
00170

```

```

00171 /** \} */
00172 /** \} */
00173 #endif // GNSS_SDR_GPS_L1_CA_DLL_PLL_TRACKING_GPU_CC_H

```

## 11.546 gps\_l1\_ca\_gaussian\_tracking\_cc.h File Reference

Interface of a processing block of a DLL + Kalman carrier tracking loop for GPS L1 C/A signals.

```

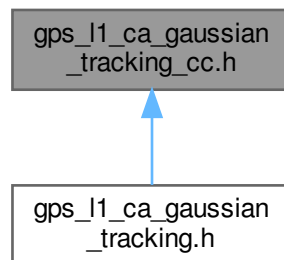
#include "bayesian_estimation.h"
#include "cpu_multicorrelator_real_codes.h"
#include "gnss_block_interface.h"
#include "gnss_synchro.h"
#include "tracking_2nd_DLL_filter.h"
#include "tracking_2nd_PLL_filter.h"
#include <armadillo>
#include <gnuradio/block.h>
#include <volk_gnssssdr/volk_gnssssdr_alloc.h>
#include <fstream>
#include <map>
#include <string>

```

Include dependency graph for `gps_l1_ca_gaussian_tracking_cc.h`:



This graph shows which files directly or indirectly include this file:



### Classes

- class [Gps\\_L1\\_Ca\\_Gaussian\\_Tracking\\_cc](#)  
This class implements a DLL + PLL tracking loop block.

### Typedefs

- using [gps\\_l1\\_ca\\_gaussian\\_tracking\\_cc\\_sptr](#) = `gnss_shared_ptr<Gps_L1_Ca_Gaussian_Tracking_cc>`

## Functions

- `gps_l1_ca_gaussian_tracking_cc_sptr` **gps\_l1\_ca\_gaussian\_make\_tracking\_cc** (uint32\_t order, int64\_t fs\_in, uint32\_t vector\_length, bool dump, const std::string &dump\_filename, float dll\_bw\_hz, float early\_late\_space\_chips, bool bce\_run, uint32\_t bce\_pttrans, uint32\_t bce\_strans, int32\_t bce\_nu, int32\_t bce\_kappa)

### 11.546.1 Detailed Description

Interface of a processing block of a DLL + Kalman carrier tracking loop for GPS L1 C/A signals.

#### Author

Javier Arribas, 2018. jarribas(at)cttc.es

Jordi Vila-Valls 2018. jvila(at)cttc.es

Carles Fernandez-Prades 2018. cfernandez(at)cttc.es

Reference: J. Vila-Valls, P. Closas, M. Navarro and C. Fernandez-Prades, "Are PLLs Dead? A Tutorial on Kalman Filter-based Techniques for Digital Carrier Synchronization", IEEE Aerospace and Electronic Systems Magazine, Vol. 32, No. 7, pp. 28–45, July 2017. DOI: 10.1109/MAES.2017.150260

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [gps\\_l1\\_ca\\_gaussian\\_tracking\\_cc.h](#).

## 11.547 gps\_l1\_ca\_gaussian\_tracking\_cc.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file gps_l1_ca_gaussian_tracking_cc.h
00003  * \brief Interface of a processing block of a DLL + Kalman carrier
00004  * \tracking loop for GPS L1 C/A signals
00005  * \author Javier Arribas, 2018. jarribas(at)cttc.es
00006  * \author Jordi Vila-Valls 2018. jvila(at)cttc.es
00007  * \author Carles Fernandez-Prades 2018. cfernandez(at)cttc.es
00008  *
00009  * Reference:
00010  * J. Vila-Valls, P. Closas, M. Navarro and C. Fernandez-Prades,
00011  * "Are PLLs Dead? A Tutorial on Kalman Filter-based Techniques for Digital
00012  * Carrier Synchronization", IEEE Aerospace and Electronic Systems Magazine,
00013  * Vol. 32, No. 7, pp. 28-45, July 2017. DOI: 10.1109/MAES.2017.150260
00014  *
00015  * -----
00016  *
00017  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00018  * This file is part of GNSS-SDR.
00019  *
00020  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00021  * SPDX-License-Identifier: GPL-3.0-or-later
00022  *
00023  * -----
00024  */
00025
00026 #ifndef GNSS_SDR_GPS_L1_CA_GAUSSIAN_TRACKING_CC_H
00027 #define GNSS_SDR_GPS_L1_CA_GAUSSIAN_TRACKING_CC_H
00028
00029 #if ARMA_NO_BOUND_CHECKING
00030 #define ARMA_NO_DEBUG 1
00031 #endif
00032
00033 #include "bayesian_estimation.h"
00034 #include "cpu_multicorrelator_real_codes.h"
00035 #include "gnss_block_interface.h"
00036 #include "gnss_synchro.h"
00037 #include "tracking_2nd_DLL_filter.h"
00038 #include "tracking_2nd_PLL_filter.h"
00039 #include <armadillo>
00040 #include <gnuradio/block.h>
00041 #include <volk_gnssdr/volk_gnssdr_alloc.h> // for volk_gnssdr::vector
00042 #include <fstream>
00043 #include <map>
00044 #include <string>
00045
00046 /** \addtogroup Tracking
00047  * \{ */
00048 /** \addtogroup Tracking_gnuradio_blocks
```

```

00049  * \{ */
00050
00051
00052  class Gps_L1_Ca_Gaussian_Tracking_cc;
00053
00054  using gps_l1_ca_gaussian_tracking_cc_sptr = gnss_shared_ptr<Gps_L1_Ca_Gaussian_Tracking_cc>;
00055
00056  gps_l1_ca_gaussian_tracking_cc_sptr
00057  gps_l1_ca_gaussian_make_tracking_cc(uint32_t order,
00058      int64_t fs_in,
00059      uint32_t vector_length,
00060      bool dump,
00061      const std::string& dump_filename,
00062      float dll_bw_hz,
00063      float early_late_space_chips,
00064      bool bce_run,
00065      uint32_t bce_ptrans,
00066      uint32_t bce_strans,
00067      int32_t bce_nu,
00068      int32_t bce_kappa);
00069
00070
00071  /*!
00072  * \brief This class implements a DLL + PLL tracking loop block
00073  */
00074  class Gps_L1_Ca_Gaussian_Tracking_cc : public gr::block
00075  {
00076  public:
00077      ~Gps_L1_Ca_Gaussian_Tracking_cc();
00078
00079      void set_channel(uint32_t channel);
00080      void set_gnss_synchro(Gnss_Synchro* p_gnss_synchro);
00081      void start_tracking();
00082
00083      int general_work(int noutput_items, gr_vector_int& ninput_items,
00084          gr_vector_const_void_star& input_items, gr_vector_void_star& output_items);
00085
00086      void forecast(int noutput_items, gr_vector_int& ninput_items_required);
00087
00088  private:
00089      friend gps_l1_ca_gaussian_tracking_cc_sptr
00090      gps_l1_ca_gaussian_make_tracking_cc(uint32_t order,
00091          int64_t fs_in,
00092          uint32_t vector_length,
00093          bool dump,
00094          const std::string& dump_filename,
00095          float dll_bw_hz,
00096          float early_late_space_chips,
00097          bool bce_run,
00098          uint32_t bce_ptrans,
00099          uint32_t bce_strans,
00100          int32_t bce_nu,
00101          int32_t bce_kappa);
00102
00103      Gps_L1_Ca_Gaussian_Tracking_cc(uint32_t order,
00104          int64_t fs_in,
00105          uint32_t vector_length,
00106          bool dump,
00107          const std::string& dump_filename,
00108          float dll_bw_hz,
00109          float early_late_space_chips,
00110          bool bce_run,
00111          uint32_t bce_ptrans,
00112          uint32_t bce_strans,
00113          int32_t bce_nu,
00114          int32_t bce_kappa);
00115
00116      int32_t save_matfile();
00117
00118      // tracking configuration vars
00119      uint32_t d_order;
00120      uint32_t d_vector_length;
00121      bool d_dump;
00122
00123      Gnss_Synchro* d_acquisition_gnss_synchro;
00124      uint32_t d_channel;
00125
00126      int64_t d_fs_in;
00127
00128      double d_early_late_spc_chips;
00129
00130      // remaining code phase and carrier phase between tracking loops
00131      double d_rem_code_phase_samples;
00132      double d_rem_code_phase_chips;
00133      float d_rem_carr_phase_rad;
00134
00135      // Kalman filter variables

```

```

00136     arma::mat kf_P_x_ini; // initial state error covariance matrix
00137     arma::mat kf_P_x;    // state error covariance matrix
00138     arma::mat kf_P_x_pre; // Predicted state error covariance matrix
00139     arma::mat kf_P_y;    // innovation covariance matrix
00140
00141     arma::mat kf_F; // state transition matrix
00142     arma::mat kf_H; // system matrix
00143     arma::mat kf_R; // measurement error covariance matrix
00144     arma::mat kf_Q; // system error covariance matrix
00145
00146     arma::colvec kf_x; // state vector
00147     arma::colvec kf_x_pre; // predicted state vector
00148     arma::colvec kf_y; // measurement vector
00149     arma::mat kf_K; // Kalman gain matrix
00150
00151     // Gaussian estimator
00152     Bayesian_estimator bayes_estimator;
00153     arma::mat kf_R_est; // measurement error covariance
00154     uint32_t bayes_ptrans;
00155     uint32_t bayes_strans;
00156     int32_t bayes_nu;
00157     int32_t bayes_kappa;
00158
00159     bool bayes_run;
00160     uint32_t kf_iter;
00161
00162     // PLL and DLL filter library
00163     Tracking_2nd_DLL_filter d_code_loop_filter;
00164     // Tracking_2nd_PLL_filter d_carrier_loop_filter;
00165
00166     // acquisition
00167     double d_acq_carrier_doppler_step_hz{};
00168     double d_acq_code_phase_samples;
00169     double d_acq_carrier_doppler_hz;
00170     // correlator
00171     int32_t d_n_correlator_taps;
00172     volk_gnssdr::vector<float> d_ca_code;
00173     volk_gnssdr::vector<float> d_local_code_shift_chips;
00174     volk_gnssdr::vector<gr_complex> d_correlator_outs;
00175     Cpu_Multicorrelator_Real_Codes multicorrelator_cpu;
00176
00177     // tracking vars
00178     double d_code_freq_chips;
00179     double d_code_phase_step_chips;
00180     double d_code_phase_rate_step_chips;
00181     double d_carrier_doppler_hz;
00182     double d_carrier_dopplerrate_hz2;
00183     double d_carrier_phase_step_rad;
00184     double d_acc_carrier_phase_rad;
00185     double d_carr_phase_error_rad{};
00186     double d_carr_phase_sigma2;
00187     double d_code_phase_samples;
00188     double code_error_chips;
00189     double code_error_filt_chips;
00190
00191     // PRN period in samples
00192     int32_t d_current_prn_length_samples;
00193
00194     // processing samples counters
00195     uint64_t d_sample_counter;
00196     uint64_t d_acq_sample_stamp;
00197
00198     // CN0 estimation and lock detector
00199     int32_t d_cn0_estimation_counter;
00200     volk_gnssdr::vector<gr_complex> d_Prompt_buffer;
00201     double d_carrier_lock_test;
00202     double d_CN0_SNV_dB_Hz;
00203     double d_carrier_lock_threshold;
00204     int32_t d_carrier_lock_fail_counter;
00205
00206     // control vars
00207     bool d_enable_tracking;
00208     bool d_pull_in;
00209
00210     // file dump
00211     std::string d_dump_filename;
00212     std::ofstream d_dump_file;
00213
00214     std::map<std::string, std::string> systemName;
00215     std::string sys;
00216 };
00217
00218
00219 /** \} */
00220 /** \} */
00221 #endif // GNSS_SDR_GPS_L1_CA_GAUSSIAN_TRACKING_CC_H

```

## 11.548 gps\_l1\_ca\_tcp\_connector\_tracking\_cc.h File Reference

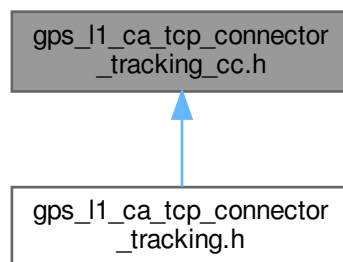
Interface of a TCP connector block based on code DLL + carrier PLL.

```
#include "cpu_multicorrelator.h"
#include "gnss_block_interface.h"
#include "gnss_synchro.h"
#include "tcp_communication.h"
#include <gnuradio/block.h>
#include <volk_gnssdr/volk_gnssdr_alloc.h>
#include <fstream>
#include <map>
#include <string>
```

Include dependency graph for `gps_l1_ca_tcp_connector_tracking_cc.h`:



This graph shows which files directly or indirectly include this file:



### Classes

- class [Gps\\_L1\\_Ca\\_Tcp\\_Connector\\_Tracking\\_cc](#)  
*This class implements a DLL + PLL tracking loop block.*

### Typedefs

- using [gps\\_l1\\_ca\\_tcp\\_connector\\_tracking\\_cc\\_sptr](#) = `gnss_shared_ptr<Gps\_L1\_Ca\_Tcp\_Connector\_Tracking\_cc>`

### Functions

- `gps_l1_ca_tcp_connector_tracking_cc_sptr` **gps\_l1\_ca\_tcp\_connector\_make\_tracking\_cc** (`int64_t` fs\_in, `uint32_t` vector\_length, `bool` dump, `const std::string` &dump\_filename, `float` early\_late\_space\_chips, `size_t` port\_ch0)

### 11.548.1 Detailed Description

Interface of a TCP connector block based on code DLL + carrier PLL.



## Author

David Pubill, 2012. dpubill(at)cttc.es Javier Arribas, 2011. jarribas(at)cttc.es

Code DLL + carrier PLL according to the algorithms described in: K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen, A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach, Birkhauser, 2007  
GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [gps\\_l1\\_ca\\_tcp\\_connector\\_tracking\\_cc.h](#).

## 11.549 gps\_l1\_ca\_tcp\_connector\_tracking\_cc.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file gps_l1_ca_tcp_connector_tracking_cc.h
00003  * \brief Interface of a TCP connector block based on code DLL + carrier PLL
00004  * \author David Pubill, 2012. dpubill(at)cttc.es
00005  *          Javier Arribas, 2011. jarribas(at)cttc.es
00006  *
00007  * Code DLL + carrier PLL according to the algorithms described in:
00008  * K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H.Jensen,
00009  * A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach,
00010  * Birkhauser, 2007
00011  *
00012  * -----
00013  *
00014  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00015  * This file is part of GNSS-SDR.
00016  *
00017  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00018  * SPDX-License-Identifier: GPL-3.0-or-later
00019  *
00020  * -----
00021  */
00022
00023 #ifndef GNSS_SDR_GPS_L1_CA_TCP_CONNECTOR_TRACKING_CC_H
00024 #define GNSS_SDR_GPS_L1_CA_TCP_CONNECTOR_TRACKING_CC_H
00025
00026 #include "cpu_multicorrelator.h"
00027 #include "gnss_block_interface.h"
00028 #include "gnss_synchro.h"
00029 #include "tcp_communication.h"
00030 #include <gnuradio/block.h>
00031 #include <volk_gnssdr/volk_gnssdr_alloc.h> // for volk_gnssdr::vector
00032 #include <fstream>
00033 #include <map>
00034 #include <string>
00035
00036 /** \addtogroup Tracking
00037  * \{ */
00038 /** \addtogroup Tracking_gnuradio_blocks
00039  * \{ */
00040
00041
00042 class Gps_L1_Ca_Tcp_Connector_Tracking_cc;
00043
00044 using gps_l1_ca_tcp_connector_tracking_cc_sptr = gnss_shared_ptr<Gps_L1_Ca_Tcp_Connector_Tracking_cc>;
00045
00046 gps_l1_ca_tcp_connector_tracking_cc_sptr
00047 gps_l1_ca_tcp_connector_make_tracking_cc(
00048     int64_t fs_in, uint32_t vector_length,
00049     bool dump,
00050     const std::string &dump_filename,
00051     float early_late_space_chips,
00052     size_t port_ch0);
00053
00054
00055 /*!
00056  * \brief This class implements a DLL + PLL tracking loop block
00057  */
00058 class Gps_L1_Ca_Tcp_Connector_Tracking_cc : public gr::block
00059 {
00060 public:
00061     ~Gps_L1_Ca_Tcp_Connector_Tracking_cc();
00062
00063     void set_channel(uint32_t channel);
00064     void set_gnss_synchro(Gnss_Synchro *p_gnss_synchro);
00065     void start_tracking();
00066
00067     /*
00068     * \brief just like gr_block::general_work, only this arranges to call consume_each for you
00069     */

```

```

00070      * The user must override work to define the signal processing code
00071      */
00072      int general_work(int noutput_items, gr_vector_int &ninput_items,
00073                      gr_vector_const_void_star &input_items, gr_vector_void_star &output_items);
00074
00075      void forecast(int noutput_items, gr_vector_int &ninput_items_required);
00076
00077  private:
00078      friend gps_ll_ca_tcp_connector_tracking_cc_sptr
00079      gps_ll_ca_tcp_connector_make_tracking_cc(
00080          int64_t fs_in, uint32_t vector_length,
00081          bool dump,
00082          const std::string &dump_filename,
00083          float early_late_space_chips,
00084          size_t port_ch0);
00085
00086      Gps_Ll_Ca_Tcp_Connector_Tracking_cc(
00087          int64_t fs_in, uint32_t vector_length,
00088          bool dump,
00089          const std::string &dump_filename,
00090          float early_late_space_chips,
00091          size_t port_ch0);
00092
00093      volk_gnssssdr::vector<gr_complex> d_ca_code;
00094      // correlator
00095      volk_gnssssdr::vector<float> d_local_code_shift_chips;
00096      volk_gnssssdr::vector<gr_complex> d_correlator_outs;
00097      volk_gnssssdr::vector<gr_complex> d_Prompt_buffer;
00098      Cpu_Multicorrelator multicorrelator_cpu;
00099      Tcp_Communication d_tcp_com;
00100      Gnss_Synchro *d_acquisition_gnss_synchro;
00101      // tracking configuration vars
00102
00103      gr_complex *d_Early;
00104      gr_complex *d_Prompt;
00105      gr_complex *d_Late;
00106
00107      // file dump
00108      std::string d_dump_filename;
00109      std::ofstream d_dump_file;
00110
00111      std::map<std::string, std::string> systemName;
00112      std::string sys;
00113
00114      double d_early_late_spc_chips;
00115      double d_code_phase_step_chips;
00116      double d_rem_code_phase_samples;
00117      double d_next_rem_code_phase_samples;
00118      double d_code_freq_hz;
00119      double d_carrier_doppler_hz;
00120      double d_acc_carrier_phase_rad;
00121      double d_code_phase_samples;
00122      double d_sample_counter_seconds;
00123
00124      int64_t d_fs_in;
00125      uint64_t d_sample_counter;
00126      uint64_t d_acq_sample_stamp;
00127
00128      size_t d_port_ch0;
00129      size_t d_port;
00130
00131      uint32_t d_vector_length;
00132      uint32_t d_channel;
00133
00134      int32_t d_correlation_length_samples;
00135      int32_t d_n_correlator_taps;
00136      int32_t d_listen_connection;
00137      int32_t d_current_prn_length_samples;
00138      int32_t d_next_prn_length_samples;
00139      int32_t d_cn0_estimation_counter;
00140      int32_t d_carrier_lock_fail_counter;
00141
00142      float d_rem_carr_phase_rad;
00143      float d_acq_code_phase_samples;
00144      float d_acq_carrier_doppler_hz;
00145      float d_carrier_lock_test;
00146      float d_CN0_SNV_dB_Hz;
00147      float d_carrier_lock_threshold;
00148      float d_control_id;
00149
00150      bool d_enable_tracking;
00151      bool d_pull_in;
00152      bool d_dump;
00153  };
00154
00155
00156  /** \} */

```

```

00157 /** \} */
00158 #endif // GNSS_SDR_GPS_L1_CA_TCP_CONNECTOR_TRACKING_CC_H

```

## 11.550 kf\_tracking.h

```

00001 /*!
00002  * \file kf_tracking.cc
00003  * \brief Implementation of a Kalman filter based tracking with optional Vector
00004  * Tracking Loop message receiver block.
00005  * \author Javier Arribas, 2020. jarribas(at)cttc.es
00006  *
00007  * -----
00008  *
00009  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00010  *
00011  * GNSS-SDR is a software defined Global Navigation
00012  * Satellite Systems receiver
00013  *
00014  * This file is part of GNSS-SDR.
00015  *
00016  * SPDX-License-Identifier: GPL-3.0-or-later
00017  *
00018  * -----
00019  */
00020
00021 #ifndef GNSS_SDR_KF_TRACKING_H
00022 #define GNSS_SDR_KF_TRACKING_H
00023
00024 #if ARMA_NO_BOUND_CHECKING
00025 #define ARMA_NO_DEBUG 1
00026 #endif
00027
00028 #include "cpu_multicorrelator_real_codes.h"
00029 #include "exponential_smoother.h"
00030 #include "gnss_block_interface.h"
00031 #include "gnss_time.h" // for timetags produced by File_Timestamp_Signal_Source
00032 #include "kf_conf.h"
00033 #include "tracking_FLL_PLL_filter.h" // for PLL/FLL filter
00034 #include "tracking_loop_filter.h" // for DLL filter
00035 #include <armadillo>
00036 #include <boost/circular_buffer.hpp>
00037 #include <gnuradio/block.h> // for block
00038 #include <gnuradio/gr_complex.h> // for gr_complex
00039 #include <gnuradio/types.h> // for gr_vector_int, gr_vector...
00040 #include <pmt/pmt.h> // for pmt_t
00041 #include <volk_gnssssdr/volk_gnssssdr_alloc.h> // for volk_gnssssdr::vector
00042 #include <cstdlib> // for size_t
00043 #include <stdint.h> // for int32_t
00044 #include <fstream> // for ofstream
00045 #include <memory>
00046 #include <string> // for string
00047 #include <typeinfo> // for typeid
00048 #include <utility> // for pair
00049
00050 class Gnss_Synchro;
00051 class kf_tracking;
00052
00053 using kf_tracking_sptr = gnss_shared_ptr<kf_tracking>;
00054
00055 kf_tracking_sptr kf_make_tracking(const Kf_Conf &conf_);
00056
00057 /*!
00058  * \brief This class implements a code DLL + carrier PLL tracking block.
00059  */
00060 class kf_tracking : public gr::block
00061 {
00062 public:
00063     ~kf_tracking();
00064
00065     void set_channel(uint32_t channel);
00066     void set_gnss_synchro(Gnss_Synchro *p_gnss_synchro);
00067     void start_tracking();
00068     void stop_tracking();
00069
00070     int general_work(int noutput_items, gr_vector_int &ninput_items,
00071         gr_vector_const_void_star &input_items, gr_vector_void_star &output_items);
00072
00073     void forecast(int noutput_items, gr_vector_int &ninput_items_required);
00074
00075 private:
00076     friend kf_tracking_sptr kf_make_tracking(const Kf_Conf &conf_);
00077     explicit kf_tracking(const Kf_Conf &conf_);
00078
00079     void init_kf(double acq_code_phase_chips, double acq_doppler_hz);

```

```

00080 void update_kf_narrow_integration_time();
00081 void update_kf_cn0(double current_cn0_dbhz);
00082 void run_Kf();
00083
00084 void msg_handler_telemetry_to_trk(const pmt::pmt_t &msg);
00085 void msg_handler_pvt_to_trk(const pmt::pmt_t &msg);
00086 void do_correlation_step(const gr_complex *input_samples);
00087
00088 void check_carrier_phase_coherent_initialization();
00089 void update_tracking_vars();
00090 void clear_tracking_vars();
00091 void save_correlation_results();
00092 void log_data();
00093 bool cn0_and_tracking_lock_status(double coh_integration_time_s);
00094 bool acquire_secondary();
00095 int32_t save_matfile() const;
00096
00097 Cpu_Multicorrelator_Real_Codes d_multicorrelator_cpu;
00098 Cpu_Multicorrelator_Real_Codes d_correlator_data_cpu; // for data channel
00099
00100 Kf_Conf d_trk_parameters;
00101
00102 Exponential_Smoother d_cn0_smoother;
00103 Exponential_Smoother d_carrier_lock_test_smoother;
00104
00105 Gnss_Synchro *d_acquisition_gnss_synchro;
00106
00107 volk_gnssdr::vector<float> d_tracking_code;
00108 volk_gnssdr::vector<float> d_data_code;
00109 volk_gnssdr::vector<float> d_local_code_shift_chips;
00110 volk_gnssdr::vector<gr_complex> d_correlator_outs;
00111 volk_gnssdr::vector<gr_complex> d_Prompt_Data;
00112 volk_gnssdr::vector<gr_complex> d_Prompt_buffer;
00113
00114 boost::circular_buffer<gr_complex> d_Prompt_circular_buffer;
00115 boost::circular_buffer<std::pair<double, double> d_code_ph_history;
00116 boost::circular_buffer<std::pair<double, double> d_carr_ph_history;
00117
00118 const size_t d_int_type_hash_code = typeid(int).hash_code();
00119
00120 // Kalman Filter class variables
00121 arma::mat d_F;
00122 arma::mat d_H;
00123 arma::mat d_R;
00124 arma::mat d_Q;
00125 arma::mat d_P_old_old;
00126 arma::mat d_P_new_old;
00127 arma::mat d_P_new_new;
00128 arma::vec d_x_old_old;
00129 arma::vec d_x_new_old;
00130 arma::vec d_x_new_new;
00131
00132 std::string d_secondary_code_string;
00133 std::string d_data_secondary_code_string;
00134 std::string d_systemName;
00135 std::string d_signal_type;
00136 std::string d_signal_pretty_name;
00137 std::string d_dump_filename;
00138
00139 std::ofstream d_dump_file;
00140
00141 gr_complex *d_Very_Early;
00142 gr_complex *d_Early;
00143 gr_complex *d_Prompt;
00144 gr_complex *d_Late;
00145 gr_complex *d_Very_Late;
00146
00147 gr_complex d_VE_accu;
00148 gr_complex d_E_accu;
00149 gr_complex d_P_accu;
00150 gr_complex d_P_accu_old;
00151 gr_complex d_L_accu;
00152 gr_complex d_VL_accu;
00153 gr_complex d_P_data_accu;
00154
00155 // nominal signal parameters
00156 double d_signal_carrier_freq;
00157 double d_code_period;
00158 double d_code_chip_rate;
00159
00160 // acquisition
00161 double d_acq_code_phase_samples;
00162 double d_acq_carrier_doppler_hz;
00163 double d_current_correlation_time_s;
00164
00165 // carrier and code discriminators output
00166 double d_carr_phase_error_disc_hz;

```

```

00167     double d_code_error_disc_chips;
00168
00169     // estimated parameters
00170     // code
00171     double d_code_error_kf_chips;
00172     double d_code_freq_kf_chips_s;
00173     // carrier
00174     double d_carrier_phase_kf_rad;
00175     double d_carrier_doppler_kf_hz;
00176     double d_carrier_doppler_rate_kf_hz_s;
00177
00178     double d_acc_carrier_phase_rad;
00179
00180     double d_T_chip_seconds;
00181     double d_T_prn_seconds;
00182     double d_T_prn_samples;
00183     double d_K_blk_samples;
00184     double d_carrier_lock_test;
00185     double d_CN0_SNV_dB_Hz;
00186     double d_carrier_lock_threshold;
00187
00188     // carrier NCO
00189     double d_carrier_phase_step_rad;
00190     double d_carrier_phase_rate_step_rad;
00191
00192     // code NCO
00193     double d_code_phase_step_chips;
00194     double d_code_phase_rate_step_chips;
00195     double d_rem_code_phase_chips;
00196     double d_rem_code_phase_samples;
00197
00198     double d_beta;
00199
00200     uint64_t d_sample_counter;
00201     uint64_t d_acq_sample_stamp;
00202
00203     float *d_prompt_data_shift;
00204     float d_rem_carr_phase_rad;
00205
00206     uint32_t d_channel;
00207     uint32_t d_secondary_code_length;
00208     uint32_t d_data_secondary_code_length;
00209
00210     int32_t d_symbols_per_bit;
00211     int32_t d_state;
00212     int32_t d_correlation_length_ms;
00213     int32_t d_n_correlator_taps;
00214     int32_t d_current_prn_length_samples;
00215     int32_t d_extend_correlation_symbols_count;
00216     int32_t d_current_symbol;
00217     int32_t d_current_data_symbol;
00218     int32_t d_cn0_estimation_counter;
00219     int32_t d_carrier_lock_fail_counter;
00220     int32_t d_code_lock_fail_counter;
00221     int32_t d_code_samples_per_chip; // All signals have 1 sample per chip code except Gal. E1 which
has 2 (CBOC disabled) or 12 (CBOC enabled)
00222     int32_t d_code_length_chips;
00223
00224     bool d_pull_in_transitory;
00225     bool d_corrected_doppler;
00226     bool d_interchange_iq;
00227     bool d_veml;
00228     bool d_cloop;
00229     bool d_secondary;
00230     bool d_dump;
00231     bool d_dump_mat;
00232     bool d_acc_carrier_phase_initialized;
00233     bool d_enable_extended_integration;
00234 };
00235
00236 #endif // GNSS_SDR_KF_TRACKING_H

```

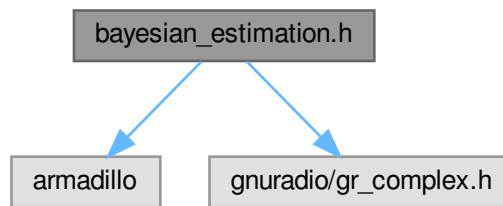
## 11.551 bayesian\_estimation.h File Reference

Interface of a library with Bayesian noise statistic estimation.

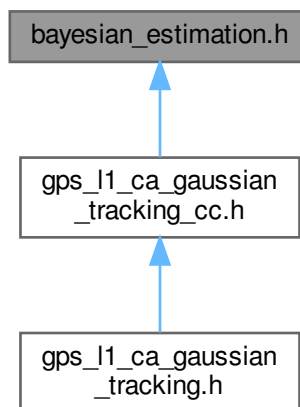
```
#include <armadillo>
```

```
#include <gnuradio/gr_complex.h>
```

Include dependency graph for `bayesian_estimation.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class `Bayesian_estimator`  
*`Bayesian_estimator` is an estimator of noise characteristics (i.e. mean, covariance)*

### 11.551.1 Detailed Description

Interface of a library with Bayesian noise statistic estimation.

`Bayesian_estimator` is a Bayesian estimator which attempts to estimate the properties of a stochastic process based on a sequence of discrete samples of the sequence.

[1]: LaMountain, Gerald, Vilà-Valls, Jordi, Closas, Pau, "Bayesian Covariance Estimation for Kalman Filter based Digital Carrier Synchronization," Proceedings of the 31st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2018), Miami, Florida, September 2018, pp. 3575-3586. <https://doi.org/10.33012/2018.15911>

## Authors

- Gerald LaMountain, 2018. [gerald\(at\)ece.neu.edu](mailto:gerald(at)ece.neu.edu)
- Jordi Vila-Valls 2018. [jvila\(at\)cttc.es](mailto:jvila(at)cttc.es)

## 11.551.1.1

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [bayesian\\_estimation.h](#).

## 11.552 bayesian\_estimation.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file bayesian_estimation.h
00003  * \brief Interface of a library with Bayesian noise statistic estimation
00004  *
00005  * Bayesian_estimator is a Bayesian estimator which attempts to estimate
00006  * the properties of a stochastic process based on a sequence of
00007  * discrete samples of the sequence.
00008  *
00009  * [1]: LaMountain, Gerald, Vilà-Valls, Jordi, Closas, Pau, "Bayesian
00010  * Covariance Estimation for Kalman Filter based Digital Carrier
00011  * Synchronization," Proceedings of the 31st International Technical Meeting
00012  * of the Satellite Division of The Institute of Navigation
00013  * (ION GNSS+ 2018), Miami, Florida, September 2018, pp. 3575-3586.
00014  * https://doi.org/10.33012/2018.15911
00015  *
00016  * \authors <ul>
00017  *     <li> Gerald LaMountain, 2018. gerald(at)ece.neu.edu
00018  *     <li> Jordi Vila-Valls 2018. jvila(at)cttc.es
00019  * </ul>
00020  * -----
00021  *
00022  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00023  * This file is part of GNSS-SDR.
00024  *
00025  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00026  * SPDX-License-Identifier: GPL-3.0-or-later
00027  *
00028  * -----
00029  */
00030
00031 #ifndef GNSS_SDR_BAYESIAN_ESTIMATION_H
00032 #define GNSS_SDR_BAYESIAN_ESTIMATION_H
00033
00034 #if ARMA_NO_BOUND_CHECKING
00035 #define ARMA_NO_DEBUG 1
00036 #endif
00037
00038 #include <armadillo>
00039 #include <gnuradio/gr_complex.h>
00040
00041 /** \addtogroup Tracking
00042  * \{ */
00043 /** \addtogroup Tracking_libs
00044  * \{ */
00045
00046
00047 /*! \brief Bayesian_estimator is an estimator of noise characteristics (i.e. mean, covariance)
00048  *
00049  * Bayesian_estimator is an estimator which performs estimation of noise characteristics from
00050  * a sequence of identically and independently distributed (IID) samples of a stationary
00051  * stochastic process by way of Bayesian inference using conjugate priors. The posterior
00052  * distribution is assumed to be Gaussian with mean  $\mu$  and covariance  $\hat{C}$ ,
00053  * which has a conjugate prior given by a normal-inverse-Wishart distribution with parameters
00054  *  $\mu_0$ ,  $\kappa_0$ ,  $\nu_0$ , and  $\Psi$ .
00055  *
00056  * [1] TODO: Ref1
00057  *
00058  */
00059
00060 class Bayesian_estimator
00061 {
00062 public:
00063     Bayesian_estimator();
00064     explicit Bayesian_estimator(int ny);
00065     Bayesian_estimator(const arma::vec& mu_prior_0, int kappa_prior_0, int nu_prior_0, const
arma::mat& Psi_prior_0);
00066     ~Bayesian_estimator() = default;
00067
00068     void init(const arma::mat& mu_prior_0, int kappa_prior_0, int nu_prior_0, const arma::mat&
Psi_prior_0);
00069
00070     void update_sequential(const arma::vec& data);
00071     void update_sequential(const arma::vec& data, const arma::vec& mu_prior_0, int kappa_prior_0, int
nu_prior_0, const arma::mat& Psi_prior_0);

```

```

00072
00073     arma::mat get_mu_est() const;
00074     arma::mat get_Psi_est() const;
00075
00076 private:
00077     arma::vec mu_est;
00078     arma::mat Psi_est;
00079     arma::vec mu_prior;
00080     arma::mat Psi_prior;
00081     int kappa_prior;
00082     int nu_prior;
00083 };
00084
00085
00086 /** \} */
00087 /** \} */
00088 #endif // GNSS_SDR_BAYESIAN_ESTIMATION_H

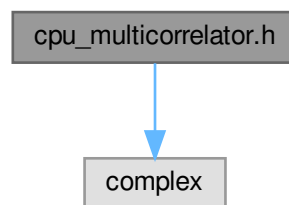
```

## 11.553 cpu\_multicorrelator.h File Reference

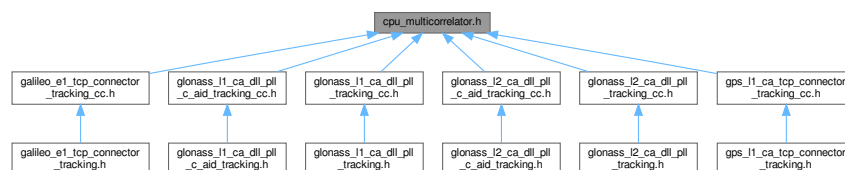
High optimized CPU vector multiTAP correlator class.

#include <complex>

Include dependency graph for cpu\_multicorrelator.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [Cpu\\_Multicorrelator](#)

*Class that implements carrier wipe-off and correlators.*

### 11.553.1 Detailed Description

High optimized CPU vector multiTAP correlator class.



## Authors

- Javier Arribas, 2015. jarribas(at)cttc.es

Class that implements a high optimized vector multiTAP correlator class for CPUs

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [cpu\\_multicorrelator.h](#).

## 11.554 cpu\_multicorrelator.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002   * \file cpu_multicorrelator.h
00003   * \brief High optimized CPU vector multiTAP correlator class
00004   * \authors <ul>
00005   *         <li> Javier Arribas, 2015. jarribas(at)cttc.es
00006   *         </ul>
00007   *
00008   * Class that implements a high optimized vector multiTAP correlator class for CPUs
00009   *
00010   * -----
00011   *
00012   * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00013   * This file is part of GNSS-SDR.
00014   *
00015   * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00016   * SPDX-License-Identifier: GPL-3.0-or-later
00017   *
00018   * -----
00019   */
00020
00021 #ifndef GNSS_SDR_CPU_MULTICORRELATOR_H
00022 #define GNSS_SDR_CPU_MULTICORRELATOR_H
00023
00024
00025 #include <complex>
00026
00027 /** \addtogroup Tracking
00028  * \{ */
00029 /** \addtogroup Tracking_libs tracking_libs
00030  * Utilities for GNSS signal tracking.
00031  * \{ */
00032
00033
00034  /*!
00035   * \brief Class that implements carrier wipe-off and correlators.
00036   */
00037  class Cpu_Multicorrelator
00038  {
00039  public:
00040      Cpu_Multicorrelator() = default;
00041      ~Cpu_Multicorrelator();
00042      bool init(int max_signal_length_samples, int n_correlators);
00043      bool set_local_code_and_taps(int code_length_chips, const std::complex<float> *local_code_in,
00044      float *shifts_chips);
00044      bool set_input_output_vectors(std::complex<float> *corr_out, const std::complex<float> *sig_in);
00045      void update_local_code(int correlator_length_samples, float rem_code_phase_chips, float
00046      code_phase_step_chips);
00046      bool Carrier_wipeoff_multicorrelator_resampler(float rem_carrier_phase_in_rad, float
00047      phase_step_rad, float rem_code_phase_chips, float code_phase_step_chips, int signal_length_samples);
00047      bool free();
00048
00049  private:
00050      // Allocate the device input vectors
00051      const std::complex<float> *d_sig_in{nullptr};
00052      const std::complex<float> *d_local_code_in{nullptr};
00053      std::complex<float> **d_local_codes_resampled{nullptr};
00054      std::complex<float> *d_corr_out{nullptr};
00055      float *d_shifts_chips{nullptr};
00056      int d_code_length_chips{0};
00057      int d_n_correlators{0};
00058  };
00059
00060
00061  /** \} */
00062  /** \} */
00063  #endif // GNSS_SDR_CPU_MULTICORRELATOR_H

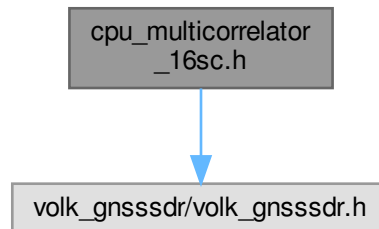
```

## 11.555 cpu\_multicorrelator\_16sc.h File Reference

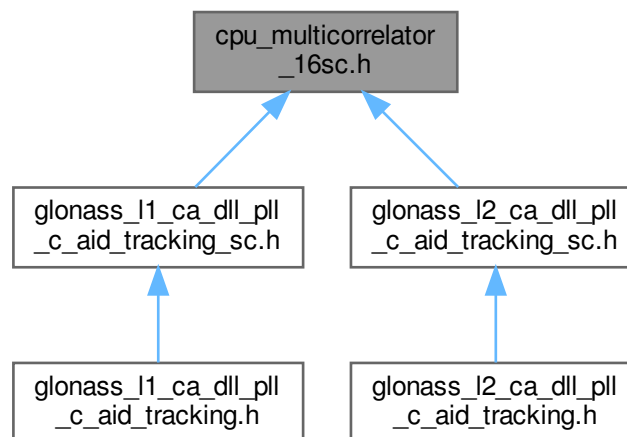
Highly optimized CPU vector multiTAP correlator class for lv\_16sc\_t (short int complex)

```
#include <volk_gnssdr/volk_gnssdr.h>
```

Include dependency graph for cpu\_multicorrelator\_16sc.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [Cpu\\_Multicorrelator\\_16sc](#)  
Class that implements carrier wipe-off and correlators.

### 11.555.1 Detailed Description

Highly optimized CPU vector multiTAP correlator class for lv\_16sc\_t (short int complex)

## Authors

- Javier Arribas, 2016. jarribas(at)cttc.es

Class that implements a highly optimized vector multiTAP correlator class for CPUs

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [cpu\\_multicorrelator\\_16sc.h](#).

## 11.556 cpu\_multicorrelator\_16sc.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file cpu_multicorrelator_16sc.h
00003  * \brief Highly optimized CPU vector multiTAP correlator class for lv_16sc_t (short int complex)
00004  * \authors <ul>
00005  *         <li> Javier Arribas, 2016. jarribas(at)cttc.es
00006  *         </li>
00007  *
00008  * Class that implements a highly optimized vector multiTAP correlator class for CPUs
00009  *
00010  * -----
00011  *
00012  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00013  * This file is part of GNSS-SDR.
00014  *
00015  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00016  * SPDX-License-Identifier: GPL-3.0-or-later
00017  *
00018  * -----
00019  */
00020
00021 #ifndef GNSS_SDR_CPU_MULTICORRELATOR_16SC_H
00022 #define GNSS_SDR_CPU_MULTICORRELATOR_16SC_H
00023
00024 #include <volk_gnssdr/volk_gnssdr.h>
00025
00026 /** \addtogroup Tracking
00027  * \{ */
00028 /** \addtogroup Tracking_libs
00029  * \{ */
00030
00031
00032  /*!
00033  * \brief Class that implements carrier wipe-off and correlators.
00034  */
00035  class Cpu_Multicorrelator_16sc
00036  {
00037  public:
00038      Cpu_Multicorrelator_16sc() = default;
00039      ~Cpu_Multicorrelator_16sc();
00040      bool init(int max_signal_length_samples, int n_correlators);
00041      bool set_local_code_and_taps(int code_length_chips, const lv_16sc_t *local_code_in, float
00042  *shifts_chips);
00043      bool set_input_output_vectors(lv_16sc_t *corr_out, const lv_16sc_t *sig_in);
00044      void update_local_code(int correlator_length_samples, float rem_code_phase_chips, float
00045  code_phase_step_chips);
00046      bool Carrier_wipeoff_multicorrelator_resampler(float rem_carrier_phase_in_rad, float
00047  phase_step_rad, float rem_code_phase_chips, float code_phase_step_chips, int signal_length_samples);
00048      bool free();
00049
00050  private:
00051      // Allocate the device input vectors
00052      const lv_16sc_t *d_sig_in{nullptr};
00053      const lv_16sc_t *d_local_code_in{nullptr};
00054      lv_16sc_t **d_local_codes_resampled{nullptr};
00055      lv_16sc_t *d_corr_out{nullptr};
00056      float *d_shifts_chips{nullptr};
00057      int d_code_length_chips{0};
00058      int d_n_correlators{0};
00059  };
00060
00061 /** \} */
00062 /** \} */
00063 #endif // GNSS_SDR_CPU_MULTICORRELATOR_H

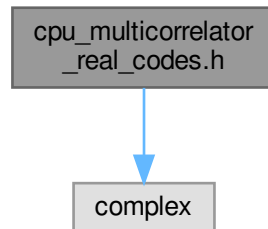
```

## 11.557 cpu\_multicorrelator\_real\_codes.h File Reference

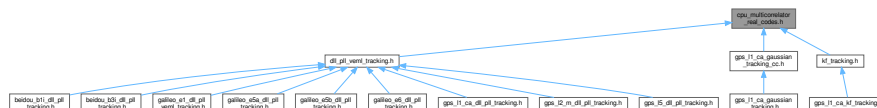
Highly optimized CPU vector multiTAP correlator class using real-valued local codes.

```
#include <complex>
```

Include dependency graph for `cpu_multicorrelator_real_codes.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Cpu\\_Multicorrelator\\_Real\\_Codes](#)  
Class that implements carrier wipe-off and correlators.

### 11.557.1 Detailed Description

Highly optimized CPU vector multiTAP correlator class using real-valued local codes.

#### Authors

- Javier Arribas, 2015. jarribas(at)cttc.es
- Cillian O'Driscoll, 2017, cillian.odriscoll(at)gmail.com

Class that implements a highly optimized vector multiTAP correlator class for CPUs

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [cpu\\_multicorrelator\\_real\\_codes.h](#).

### 11.558 cpu\_multicorrelator\_real\_codes.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file cpu_multicorrelator_real_codes.h
00003  * \brief Highly optimized CPU vector multiTAP correlator class using real-valued local codes
00004  * \authors <ul>
00005  *         <li> Javier Arribas, 2015. jarribas(at)cttc.es
00006  *         <li> Cillian O'Driscoll, 2017, cillian.odriscoll(at)gmail.com
00007  *         </li>
00008  *
00009  * Class that implements a highly optimized vector multiTAP correlator class for CPUs
00010  *
00011  * -----
00012  *
00013  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
  
```

```

00014  * This file is part of GNSS-SDR.
00015  *
00016  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00017  * SPDX-License-Identifier: GPL-3.0-or-later
00018  *
00019  * -----
00020  */
00021
00022 #ifndef GNSS_SDR_CPU_MULTICORRELATOR_REAL_CODES_H
00023 #define GNSS_SDR_CPU_MULTICORRELATOR_REAL_CODES_H
00024
00025
00026 #include <complex>
00027
00028 /** \addtogroup Tracking
00029  * \{ */
00030 /** \addtogroup Tracking_libs
00031  * \{ */
00032
00033
00034 /*!
00035  * \brief Class that implements carrier wipe-off and correlators.
00036  */
00037 class Cpu_Multicorrelator_Real_Codes
00038 {
00039 public:
00040     Cpu_Multicorrelator_Real_Codes() = default;
00041     void set_high_dynamics_resampler(bool use_high_dynamics_resampler);
00042     ~Cpu_Multicorrelator_Real_Codes();
00043     bool init(int max_signal_length_samples, int n_correlators);
00044     bool set_local_code_and_taps(int code_length_chips, const float *local_code_in, float
00045 *shifts_chips);
00046     bool set_input_output_vectors(std::complex<float> *corr_out, const std::complex<float> *sig_in);
00047     void update_local_code(int correlator_length_samples, float rem_code_phase_chips, float
00048 code_phase_step_chips, float code_phase_rate_step_chips = 0.0);
00049     bool Carrier_wipeoff_multicorrelator_resampler(float rem_carrier_phase_in_rad, float
00050 phase_step_rad, float phase_rate_step_rad, float rem_code_phase_chips, float code_phase_step_chips,
00051 float code_phase_rate_step_chips, int signal_length_samples);
00052     bool Carrier_wipeoff_multicorrelator_resampler(float rem_carrier_phase_in_rad, float
00053 phase_step_rad, float rem_code_phase_chips, float code_phase_step_chips, float
00054 code_phase_rate_step_chips, int signal_length_samples);
00055     bool free();
00056
00057 private:
00058     // Allocate the device input vectors
00059     const std::complex<float> *d_sig_in{nullptr};
00060     const float *d_local_code_in{nullptr};
00061     std::complex<float> *d_corr_out{nullptr};
00062     float **d_local_codes_resampled{nullptr};
00063     float *d_shifts_chips{nullptr};
00064     int d_code_length_chips{0};
00065     int d_n_correlators{0};
00066     bool d_use_high_dynamics_resampler{true};
00067 };
00068
00069 /** \} */
00070 /** \} */
00071 #endif // GNSS_SDR_CPU_MULTICORRELATOR_REAL_CODES_H

```

## 11.559 cuda\_multicorrelator.h File Reference

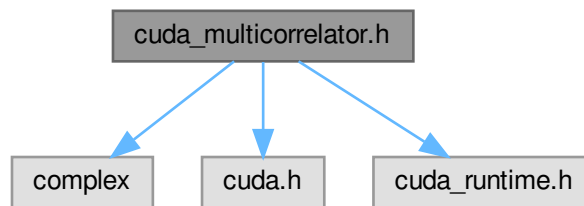
Highly optimized CUDA GPU vector multiTAP correlator class.

```

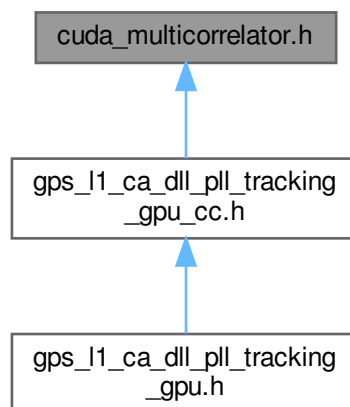
#include <complex>
#include <cuda.h>
#include <cuda_runtime.h>

```

Include dependency graph for `cuda_multicorrelator.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- struct [GPU\\_Complex](#)
- struct [GPU\\_Complex\\_Short](#)
- class [cuda\\_multicorrelator](#)

*Class that implements carrier wipe-off and correlators using NVIDIA CUDA GPU accelerators.*

### 11.559.1 Detailed Description

Highly optimized CUDA GPU vector multiTAP correlator class.

#### Authors

- Javier Arribas, 2015. [jarribas@cttc.es](mailto:jarribas@cttc.es)

Class that implements a highly optimized vector multiTAP correlator class for NVIDIA CUDA GPUs  
 GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [cuda\\_multicorrelator.h](#).

## 11.559.2 Macro Definition Documentation

### 11.559.2.1 CUDA\_CALLABLE\_MEMBER\_DEVICE

#define CUDA\_CALLABLE\_MEMBER\_DEVICE

Definition at line 30 of file [cuda\\_multicorrelator.h](#).

### 11.559.2.2 CUDA\_CALLABLE\_MEMBER\_GLOBAL

#define CUDA\_CALLABLE\_MEMBER\_GLOBAL

Definition at line 29 of file [cuda\\_multicorrelator.h](#).

## 11.560 cuda\_multicorrelator.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002   * \file cuda_multicorrelator.h
00003   * \brief Highly optimized CUDA GPU vector multiTAP correlator class
00004   * \authors <ul>
00005   *         <li> Javier Arribas, 2015. jarribas(at)cttc.es
00006   *         </li>
00007   *         </ul>
00008   * Class that implements a highly optimized vector multiTAP correlator class for NVIDIA CUDA GPUs
00009   *
00010   * -----
00011   *
00012   * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00013   * This file is part of GNSS-SDR.
00014   *
00015   * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00016   * SPDX-License-Identifier: GPL-3.0-or-later
00017   *
00018   * -----
00019   */
00020
00021 #ifndef GNSS_SDR_CUDA_MULTICORRELATOR_H
00022 #define GNSS_SDR_CUDA_MULTICORRELATOR_H
00023
00024
00025 #ifdef __CUDACC__
00026 #define CUDA_CALLABLE_MEMBER_GLOBAL __global__
00027 #define CUDA_CALLABLE_MEMBER_DEVICE __device__
00028 #else
00029 #define CUDA_CALLABLE_MEMBER_GLOBAL
00030 #define CUDA_CALLABLE_MEMBER_DEVICE
00031 #endif
00032
00033 #include <complex>
00034 #include <cuda.h>
00035 #include <cuda_runtime.h>
00036
00037 /** \addtogroup Tracking
00038  * \{ */
00039 /** \addtogroup Tracking_libs
00040  * \{ */
00041
00042
00043 // GPU new internal data types for complex numbers
00044
00045 struct GPU_Complex
00046 {
00047     float r;
00048     float i;
00049     CUDA_CALLABLE_MEMBER_DEVICE GPU_Complex() {};
00050     CUDA_CALLABLE_MEMBER_DEVICE GPU_Complex(float a, float b) : r(a), i(b) {}
00051     CUDA_CALLABLE_MEMBER_DEVICE float magnitude2(void) { return r * r + i * i; }
00052     CUDA_CALLABLE_MEMBER_DEVICE GPU_Complex operator*(const GPU_Complex& a)
00053     {
00054 #ifdef __CUDACC__
00055         return GPU_Complex(__fmul_rn(r, a.r) - __fmul_rn(i, a.i), __fmul_rn(i, a.r) + __fmul_rn(r,
00056         a.i));
00057 #else
00057         return GPU_Complex(r * a.r - i * a.i, i * a.r + r * a.i);
00058 #endif
00059     }
00060     CUDA_CALLABLE_MEMBER_DEVICE GPU_Complex operator+(const GPU_Complex& a)
00061     {
00062         return GPU_Complex(r + a.r, i + a.i);
00063     }
00064     CUDA_CALLABLE_MEMBER_DEVICE void operator+=(const GPU_Complex& a)

```

```

00065     {
00066         r += a.r;
00067         i += a.i;
00068     }
00069     CUDA_CALLABLE_MEMBER_DEVICE void multiply_acc(const GPU_Complex& a, const GPU_Complex& b)
00070     {
00071         // c=a*b+c
00072         // real part
00073         // c.r=(a.r*b.r - a.i*b.i)+c.r
00074 #ifdef __CUDACC__
00075         r = __fmaf_rn(a.r, b.r, r);
00076         r = __fmaf_rn(-a.i, b.i, r);
00077         // imag part
00078         i = __fmaf_rn(a.i, b.r, i);
00079         i = __fmaf_rn(a.r, b.i, i);
00080 #else
00081         r = (a.r * b.r - a.i * b.i) + r;
00082         i = (a.i * b.r - a.r * b.i) + i;
00083 #endif
00084     }
00085 };
00086
00087
00088 struct GPU_Complex_Short
00089 {
00090     float r;
00091     float i;
00092     CUDA_CALLABLE_MEMBER_DEVICE GPU_Complex_Short(short int a, short int b) : r(a), i(b) {}
00093     CUDA_CALLABLE_MEMBER_DEVICE float magnitude2(void)
00094     {
00095         return r * r + i * i;
00096     }
00097     CUDA_CALLABLE_MEMBER_DEVICE GPU_Complex_Short operator*(const GPU_Complex_Short& a)
00098     {
00099         return GPU_Complex_Short(r * a.r - i * a.i, i * a.r + r * a.i);
00100     }
00101     CUDA_CALLABLE_MEMBER_DEVICE GPU_Complex_Short operator+(const GPU_Complex_Short& a)
00102     {
00103         return GPU_Complex_Short(r + a.r, i + a.i);
00104     }
00105 };
00106
00107
00108 /*!
00109 * \brief Class that implements carrier wipe-off and correlators using NVIDIA CUDA GPU accelerators.
00110 */
00111 class cuda_multicorrelator
00112 {
00113 public:
00114     cuda_multicorrelator();
00115     bool init_cuda_integrated_resampler(
00116         int signal_length_samples,
00117         int code_length_chips,
00118         int n_correlators);
00119     bool set_local_code_and_taps(
00120         int code_length_chips,
00121         const std::complex<float>* local_codes_in,
00122         float* shifts_chips,
00123         int n_correlators);
00124     bool set_input_output_vectors(
00125         std::complex<float>* corr_out,
00126         std::complex<float>* sig_in);
00127
00128     bool free_cuda();
00129     bool Carrier_wipeoff_multicorrelator_resampler_cuda(
00130         float rem_carrier_phase_in_rad,
00131         float phase_step_rad,
00132         float code_phase_step_chips,
00133         float rem_code_phase_chips,
00134         int signal_length_samples,
00135         int n_correlators);
00136 private:
00137     cudaStream_t stream1;
00138     // cudaStream_t stream2;
00139
00140     // Allocate the device input vectors
00141     GPU_Complex* d_sig_in;
00142     GPU_Complex* d_nco_in;
00143     GPU_Complex* d_sig_doppler_wiped;
00144     GPU_Complex* d_local_codes_in;
00145     GPU_Complex* d_corr_out;
00146
00147     std::complex<float>* d_sig_in_cpu;
00148     std::complex<float>* d_corr_out_cpu;
00149
00150     float* d_shifts_chips;

```



```

00152     int* d_shifts_samples;
00153     int d_code_length_chips;
00154
00155     int selected_gps_device;
00156     int threadsPerBlock;
00157     int blocksPerGrid;
00158
00159     int num_gpu_devices;
00160     int selected_device;
00161 };
00162
00163
00164 /** \} */
00165 /** \} */
00166 #endif // GNSS_SDR_CUDA_MULTICORRELATOR_H

```

## 11.561 dll\_pll\_conf.h File Reference

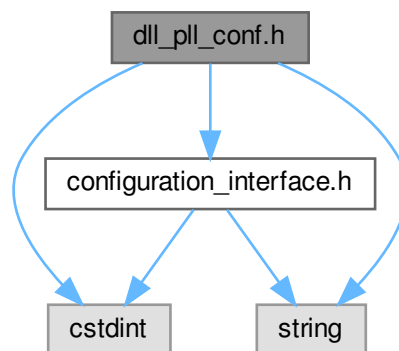
Class that contains all the configuration parameters for generic tracking block based on a DLL and a PLL.

```
#include "configuration_interface.h"
```

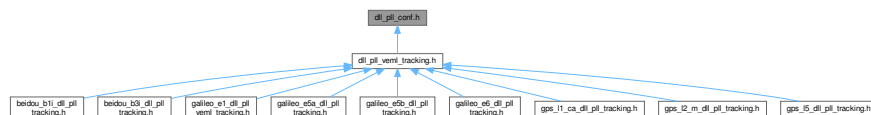
```
#include <stdint>
```

```
#include <string>
```

Include dependency graph for dll\_pll\_conf.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [Dll\\_Pll\\_Conf](#)

## 11.561.1 Detailed Description

Class that contains all the configuration parameters for generic tracking block based on a DLL and a PLL.

## Author

Javier Arribas, 2018. jarribas(at)cttc.es

Class that contains all the configuration parameters for generic tracking block based on a DLL and a PLL.  
GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [dll\\_pll\\_conf.h](#).

## 11.562 dll\_pll\_conf.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002   * \file dll_pll_conf.h
00003   * \brief Class that contains all the configuration parameters for generic tracking block based on a
00004   *        DLL and a PLL.
00005   * \author Javier Arribas, 2018. jarribas(at)cttc.es
00006   *
00007   * Class that contains all the configuration parameters for generic tracking block based on a DLL and
00008   * a PLL.
00009   *
00010   * -----
00011   * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00012   * This file is part of GNSS-SDR.
00013   *
00014   * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00015   * SPDX-License-Identifier: GPL-3.0-or-later
00016   * -----
00017  */
00018
00019  #ifndef GNSS_SDR_DLL_PLL_CONF_H
00020  #define GNSS_SDR_DLL_PLL_CONF_H
00021
00022  #include "configuration_interface.h"
00023  #include <stdint>
00024  #include <string>
00025
00026  /** \addtogroup Tracking
00027   * \{ */
00028  /** \addtogroup Tracking_libs
00029   * \{ */
00030
00031
00032  class Dll_Pll_Conf
00033  {
00034  public:
00035      Dll_Pll_Conf();
00036      void SetFromConfiguration(const ConfigurationInterface *configuration, const std::string &role);
00037
00038      /* DLL/PLL tracking configuration */
00039      std::string item_type{"gr_complex"};
00040      std::string dump_filename{"./dll_pll_dump.dat"};
00041      double fs_in{2000000.0};
00042      double carrier_lock_th{0.0};
00043      float pll_pull_in_bw_hz{50.0};
00044      float dll_pull_in_bw_hz{3.0};
00045      float fll_bw_hz{35.0};
00046      float pll_bw_hz{35.0};
00047      float dll_bw_hz{2.0};
00048      float pll_bw_narrow_hz{5.0};
00049      float dll_bw_narrow_hz{0.75};
00050      float early_late_space_chips{0.25};
00051      float very_early_late_space_chips{0.5};
00052      float early_late_space_narrow_chips{0.15};
00053      float very_early_late_space_narrow_chips{0.5};
00054      float slope{1.0};
00055      float spc{0.5};
00056      float y_intercept{1.0};
00057      float cn0_smoother_alpha{0.002};
00058      float carrier_lock_test_smoother_alpha{0.002};
00059      uint32_t pull_in_time_s{10U};
00060      uint32_t bit_synchronization_time_limit_s{20U};
00061      uint32_t vector_length{0U};
00062      uint32_t smoother_length{10U};
00063      int32_t fll_filter_order{1};
00064      int32_t pll_filter_order{3};
00065      int32_t dll_filter_order{2};
00066      int32_t extend_correlation_symbols{1};
00067      int32_t cn0_samples{0};
00068      int32_t cn0_smoother_samples{200};
00069      int32_t carrier_lock_test_smoother_samples{25};

```

```

00070     int32_t cn0_min{0};
00071     int32_t max_code_lock_fail{0};
00072     int32_t max_carrier_lock_fail{0};
00073     char signal[3]{};
00074     char system{'G'};
00075     bool enable_fll_pull_in{false};
00076     bool enable_fll_steady_state{false};
00077     bool track_pilot{true};
00078     bool enable_doppler_correction{false};
00079     bool carrier_aiding{true};
00080     bool high_dyn{false};
00081     bool dump{false};
00082     bool dump_mat{true};
00083 };
00084
00085
00086 /** \} */
00087 /** \} */
00088 #endif // GNSS_SDR_DLL_PLL_CONF_H

```

## 11.563 dll\_pll\_conf\_fpga.h File Reference

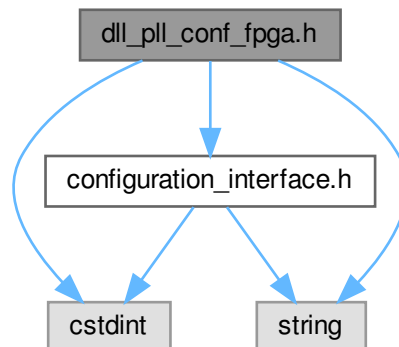
Class that contains all the configuration parameters for generic tracking block based on a DLL and a PLL for the FPGA.

```
#include "configuration_interface.h"
```

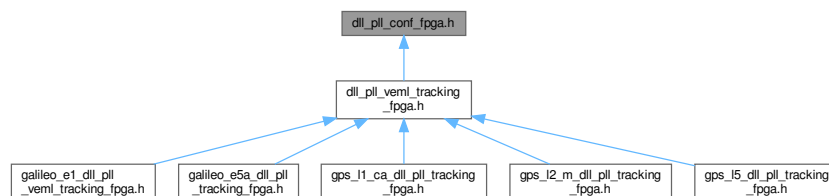
```
#include <cstdint>
```

```
#include <string>
```

Include dependency graph for `dll_pll_conf_fpga.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Dll\\_Pll\\_Conf\\_Fpga](#)

### 11.563.1 Detailed Description

Class that contains all the configuration parameters for generic tracking block based on a DLL and a PLL for the FPGA.

#### Author

Marc Majoral, 2019. mmajoral(at)cttc.cat

Javier Arribas, 2018. jarribas(at)cttc.es

Class that contains all the configuration parameters for generic tracking block based on a DLL and a PLL. GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [dll\\_pll\\_conf\\_fpga.h](#).

## 11.564 dll\_pll\_conf\_fpga.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file dll_pll_conf_fpga.h
00003  * \brief Class that contains all the configuration parameters for generic
00004  * tracking block based on a DLL and a PLL for the FPGA.
00005  * \author Marc Majoral, 2019. mmajoral(at)cttc.cat
00006  * \author Javier Arribas, 2018. jarribas(at)cttc.es
00007  *
00008  * Class that contains all the configuration parameters for generic tracking block based on a DLL and
00009  * a PLL.
00010  * -----
00011  *
00012  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00013  * This file is part of GNSS-SDR.
00014  *
00015  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00016  * SPDX-License-Identifier: GPL-3.0-or-later
00017  *
00018  * -----
00019  */
00020
00021 #ifndef GNSS_SDR_DLL_PLL_CONF_FPGA_H
00022 #define GNSS_SDR_DLL_PLL_CONF_FPGA_H
00023
00024 #include "configuration_interface.h"
00025 #include <stdint>
00026 #include <string>
00027
00028 /** \addtogroup Tracking
00029  * \{ */
00030 /** \addtogroup Tracking_libs
00031  * \{ */
00032
00033
00034 class Dll_Pll_Conf_Fpga
00035 {
00036 public:
00037     Dll_Pll_Conf_Fpga();
00038     void SetFromConfiguration(const ConfigurationInterface* configuration, const std::string& role);
00039
00040     /* DLL/PLL tracking configuration */
00041     std::string device_name{"/dev/uio"};
00042     std::string dump_filename{".dll_pll_dump.dat"};
00043
00044     double fs_in{12500000.0};
00045     double carrier_lock_th{0.0};
00046
00047     float pll_pull_in_bw_hz{50.0};
00048     float dll_pull_in_bw_hz{3.0};
00049     float fll_bw_hz{35.0};
00050     float pll_bw_hz{5.0};
00051     float dll_bw_hz{0.5};
00052     float pll_bw_narrow_hz{2.0};
00053     float dll_bw_narrow_hz{0.25};
00054     float early_late_space_chips{0.25};
00055     float very_early_late_space_chips{0.5};
00056     float early_late_space_narrow_chips{0.15};
00057     float very_early_late_space_narrow_chips{0.5};
00058     float slope{1.0};
00059     float spc{0.5};
00060     float y_intercept{1.0};
00061     float cn0_smoother_alpha{0.002};
```

```

00062     float carrier_lock_test_smoother_alpha{0.002};
00063
00064     uint32_t pull_in_time_s{10U}; // signed integer, when pull in time is not yet reached it has to
    be compared against a negative number
00065     uint32_t bit_synchronization_time_limit_s{70U};
00066     uint32_t vector_length{0U};
00067     uint32_t smoother_length{10U};
00068     uint32_t code_length_chips{0U};
00069     uint32_t code_samples_per_chip{0U};
00070     uint32_t extend_fpga_integration_periods{1};
00071     uint32_t fpga_integration_period{0};
00072
00073     int32_t fll_filter_order{1};
00074     int32_t pll_filter_order{3};
00075     int32_t dll_filter_order{2};
00076     int32_t extend_correlation_symbols{1};
00077     int32_t cn0_samples{0};
00078     int32_t cn0_min{0};
00079     int32_t max_code_lock_fail{0};
00080     int32_t max_carrier_lock_fail{0};
00081     int32_t cn0_smoother_samples{200};
00082     int32_t carrier_lock_test_smoother_samples{25};
00083     // int32_t max_lock_fail;
00084
00085     int32_t* ca_codes{nullptr};
00086     int32_t* data_codes{nullptr};
00087
00088     char signal[3]{};
00089     char system{'G'};
00090
00091     bool extended_correlation_in_fpga{false};
00092     bool track_pilot{true};
00093     bool enable_doppler_correction{false};
00094     bool enable_fll_pull_in{false};
00095     bool enable_fll_steady_state{false};
00096     bool carrier_aiding{true};
00097     bool high_dyn{false};
00098     bool dump{false};
00099     bool dump_mat{true};
00100 };
00101
00102
00103 /** \} */
00104 /** \} */
00105 #endif // GNSS_SDR_DLL_PLL_CONF_FPGA_H

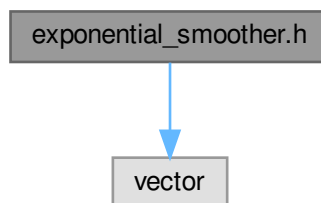
```

## 11.565 exponential\_smoother.h File Reference

Class that implements an exponential smoother.

#include <vector>

Include dependency graph for exponential\_smoother.h:





```

00046     Exponential_Smoothing operator=(Exponential_Smoothing&& /*other*/) = default;  //!< Move assignment
operator
00047
00048     void set_alpha(float alpha);  //!< 0 < alpha < 1. The higher, the most
responsive, but more variance. Default value: 0.001
00049     void set_samples_for_initialization(int num_samples);  //!< Number of samples averaged for
initialization. Default value: 200
00050     void reset();
00051     void set_min_value(float value);
00052     void set_offset(float offset);
00053     float smooth(float raw);
00054     double smooth(double raw);
00055
00056 private:
00057     std::vector<float> init_buffer_;
00058     float alpha_{0.001};
00059     float one_minus_alpha_{0.999};
00060     float old_value_{0.0};
00061     float min_value_{25.0};
00062     float offset_{12.0};
00063     int samples_for_initialization_{200};
00064     int init_counter_{0};
00065     bool initializing_{true};
00066 };
00067
00068
00069 /** \} */
00070 /** \} */
00071 #endif  // GNSS_SDR_EXPONENTIAL_SMOOTHER_H

```

## 11.567 fpga\_multicorrelator.h File Reference

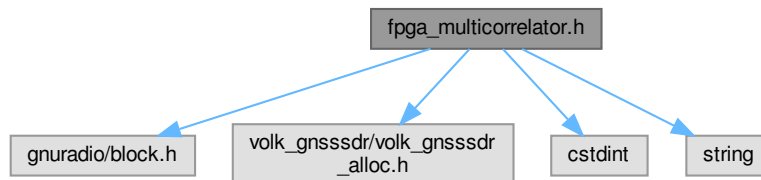
FPGA vector correlator class.

```

#include <gnuradio/block.h>
#include <volk_gnssssdr/volk_gnssssdr_alloc.h>
#include <cstdint>
#include <string>

```

Include dependency graph for fpga\_multicorrelator.h:



### Classes

- class [Fpga\\_Multicorrelator\\_8sc](#)  
Class that implements carrier wipe-off and correlators.

### 11.567.1 Detailed Description

FPGA vector correlator class.

#### Authors

- Marc Majoral, 2019. mmajoral(at)cttc.cat
- Javier Arribas, 2019. jarribas(at)cttc.es

Class that controls and executes a highly optimized vector correlator class in the FPGA  
GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [fpga\\_multicorrelator.h](#).

## 11.568 fpga\_multicorrelator.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file fpga_multicorrelator.h
00003  * \brief FPGA vector correlator class
00004  * \authors <ul>
00005  *         <li> Marc Majoral, 2019. mmajoral(at)cttc.cat
00006  *         <li> Javier Arribas, 2019. jarribas(at)cttc.es
00007  *         </ul>
00008  *
00009  * Class that controls and executes a highly optimized vector correlator
00010  * class in the FPGA
00011  *
00012  * -----
00013  *
00014  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00015  * This file is part of GNSS-SDR.
00016  *
00017  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00018  * SPDX-License-Identifier: GPL-3.0-or-later
00019  *
00020  * -----
00021  */
00022
00023 #ifndef GNSS_SDR_FPGA_MULTICORRELATOR_H
00024 #define GNSS_SDR_FPGA_MULTICORRELATOR_H
00025
00026 #include <gnuradio/block.h>
00027 #include <volk_gnssssdr/volk_gnssssdr_alloc.h>
00028 #include <cstdint>
00029 #include <string>
00030
00031 /** \addtogroup Tracking
00032  * \{ */
00033 /** \addtogroup Tracking_libs
00034  * \{ */
00035
00036
00037 /*!
00038  * \brief Class that implements carrier wipe-off and correlators.
00039  */
00040 class Fpga_Multicorrelator_8sc
00041 {
00042 public:
00043     /*!
00044     * \brief Constructor
00045     */
00046     Fpga_Multicorrelator_8sc(int32_t n_correlators,
00047                             int32_t *ca_codes,
00048                             int32_t *data_codes,
00049                             uint32_t code_length_chips,
00050                             bool track_pilot,
00051                             uint32_t code_samples_per_chip);
00052
00053     /*!
00054     * \brief Destructor
00055     */
00056     ~Fpga_Multicorrelator_8sc();
00057
00058     /*!
00059     * \brief Configure pointers to the FPGA multicorrelator results
00060     */
00061     void set_output_vectors(gr_complex *corr_out, gr_complex *Prompt_Data);
00062
00063     /*!
00064     * \brief Configure the local code in the FPGA multicorrelator
00065     */
00066     void set_local_code_and_taps(
00067         float *shifts_chips, float *prompt_data_shift, int32_t PRN);
00068
00069     /*!
00070     * \brief Configure code phase and code rate parameters in the FPGA
00071     */
00072     void update_local_code();
00073
00074     /*!
00075     * \brief Perform a multicorrelation
00076     */
00077     void Carrier_wipeoff_multicorrelator_resampler(
00078         float rem_carrier_phase_in_rad,
00079         float phase_step_rad,
00080         float carrier_phase_rate_step_rad,
00081         float rem_code_phase_chips,
00082         float code_phase_step_chips,
00083         float code_phase_rate_step_chips,

```



```

00084         int32_t signal_length_samples);
00085
00086     /*!
00087     * \brief Stop the correlation process in the FPGA and free code phase and code rate parameters
00088     */
00089     bool free();
00090
00091     /*!
00092     * \brief Open the FPGA device driver
00093     */
00094     void open_channel(const std::string &device_io_name, uint32_t channel);
00095
00096     /*!
00097     * \brief Set the initial sample number where the tracking process begins
00098     */
00099     void set_initial_sample(uint64_t samples_offset);
00100
00101     /*!
00102     * \brief Read the sample counter in the FPGA
00103     */
00104     uint64_t read_sample_counter();
00105
00106     /*!
00107     * \brief Start the tracking process in the FPGA
00108     */
00109     void lock_channel();
00110
00111     /*!
00112     * \brief finish the tracking process in the FPGA
00113     */
00114     void unlock_channel();
00115
00116     /*!
00117     * \brief Set the secondary code length in the FPGA. This is only used when extended coherent
00118     integration
00119     * is enabled in the FPGA. If tracking the pilot is enabled then secondary_code_0_length is the
00120     length of the pilot
00121     * secondary code and secondary_code_1_length is the length of the data secondary code. If
00122     tracking the pilot is disabled
00123     * then secondary_code_0_length is the length of the data secondary code, and
00124     secondary_code_1_length must be set to zero.
00125     */
00126     void set_secondary_code_lengths(uint32_t secondary_code_0_length, uint32_t
00127     secondary_code_1_length);
00128
00129     /*!
00130     * \brief Initialize the secondary code in the FPGA. If tracking the pilot is enabled then the
00131     pilot secondary code is
00132     * configured when secondary_code = 0 and the data secondary code is configured when
00133     secondary_code = 1. If tracking the
00134     * pilot is disabled then the data secondary code is configured when secondary code = 0.
00135     */
00136     void initialize_secondary_code(uint32_t secondary_code, std::string *secondary_code_string);
00137
00138     /*!
00139     * \brief Set the PRN length in the FPGA in number of samples. This function is only used then
00140     extended coherent integration is enabled in the
00141     * FPGA. The FPGA allows for the configuration of two PRN lengths. When the length of the extended
00142     coherent integration is bigger than the
00143     * length of the PRN code, the FPGA uses the first_length_secondary_code as the length of the PRN
00144     code immediately following the beginning
00145     * of the extended coherent integration, and the next_length_secondary_code as the length of the
00146     remaining PRN codes.
00147     * The purpose of this is to have the option to allow the FPGA to compensate for a possible
00148     deviation between the nominal value of the PRN
00149     * code length and the measured PRN code length in the PRN immediately following the start of the
00150     coherent integration only.
00151     * If this option is not used then write the same value to first_length_secondary_code and
00152     next_length_secondary_code.
00153     */
00154     void update_prn_code_length(uint32_t first_prn_length, uint32_t next_prn_length);
00155
00156     /*!
00157     * \brief Enable the use of secondary codes in the FPGA
00158     */
00159     void enable_secondary_codes();
00160
00161     /*!
00162     * \brief Disable the use of secondary codes in the FPGA
00163     */
00164     void disable_secondary_codes();
00165
00166 private:
00167     // FPGA register addresses
00168     // write addresses
00169     static const uint32_t code_phase_step_chips_num_reg_addr = 0;
00170     static const uint32_t initial_index_reg_base_addr = 1;

```

```

00157     static const uint32_t initial_interp_counter_reg_base_addr = 7;
00158     static const uint32_t nsamples_minus_1_reg_addr = 13;
00159     static const uint32_t code_length_minus_1_reg_addr = 14;
00160     static const uint32_t rem_carr_phase_rad_reg_addr = 15;
00161     static const uint32_t phase_step_rad_reg_addr = 16;
00162     static const uint32_t prog_mems_addr = 17;
00163     static const uint32_t drop_samples_reg_addr = 18;
00164     static const uint32_t initial_counter_value_reg_addr_lsw = 19;
00165     static const uint32_t initial_counter_value_reg_addr_msw = 20;
00166     static const uint32_t code_phase_step_chips_rate_reg_addr = 21;
00167     static const uint32_t phase_step_rate_reg_addr = 22;
00168     static const uint32_t stop_tracking_reg_addr = 23;
00169     static const uint32_t secondary_code_lengths_reg_addr = 25;
00170     static const uint32_t prog_secondary_code_0_data_reg_addr = 26;
00171     static const uint32_t prog_secondary_code_1_data_reg_addr = 27;
00172     static const uint32_t first_prn_length_minus_1_reg_addr = 28;
00173     static const uint32_t next_prn_length_minus_1_reg_addr = 29;
00174     static const uint32_t start_flag_addr = 30;
00175     // read-write addresses
00176     static const uint32_t test_reg_addr = 31;
00177     // read addresses
00178     static const uint32_t result_reg_real_base_addr = 1;
00179     static const uint32_t result_reg_imag_base_addr = 7;
00180     static const uint32_t sample_counter_reg_addr_lsw = 13;
00181     static const uint32_t sample_counter_reg_addr_msw = 14;
00182     // FPGA-related constants
00183     static const uint32_t secondary_code_word_size = 20;           // the secondary codes are written in
to the FPGA in words of secondary_code_word_size bits
00184     static const uint32_t secondary_code_wr_strobe = 0x800000;    // write strobe position in the
secondary code write register
00185     static const uint32_t secondary_code_addr_bits = 0x100000;    // memory address position in the
secondary code write register
00186     static const uint32_t drop_samples = 1;                      // bit 0 of drop_samples_reg_addr
00187     static const uint32_t enable_secondary_code = 2;             // bit 1 of drop_samples_reg_addr
00188     static const uint32_t init_secondary_code_addresses = 4;     // bit 2 of drop_samples_reg_addr
00189     static const uint32_t FPGA_PAGE_SIZE = 0x1000;
00190     static const uint32_t max_code_resampler_counter = 1 « 31;   // 2^(number of bits of precision of
the code resampler)
00191     static const uint32_t local_code_fpga_clear_address_counter = 0x10000000;
00192     static const uint32_t test_register_track_writeval = 0x55AA;
00193
00194     // private functions
00195     uint32_t fpga_acquisition_test_register(uint32_t writeval);
00196     void fpga_configure_tracking_gps_local_code(int32_t PRN);
00197     void fpga_compute_code_shift_parameters();
00198     void fpga_configure_code_parameters_in_fpga();
00199     void fpga_compute_signal_parameters_in_fpga();
00200     void fpga_configure_signal_parameters_in_fpga();
00201     void fpga_launch_multicorrelator_fpga();
00202     void read_tracking_gps_results();
00203     void close_device(void);
00204     void write_secondary_code(uint32_t secondary_code_length, std::string *secondary_code_string,
uint32_t reg_addr);
00205
00206     volk_gnssssdr::vector<uint32_t> d_initial_index;
00207     volk_gnssssdr::vector<uint32_t> d_initial_interp_counter;
00208
00209     uint64_t d_initial_sample_counter;
00210
00211     gr_complex *d_corr_out;
00212     gr_complex *d_Prompt_Data;
00213
00214     float *d_shifts_chips;
00215     float *d_prompt_data_shift;
00216
00217     float d_rem_code_phase_chips;
00218     float d_code_phase_step_chips;
00219     float d_code_phase_rate_step_chips;
00220     float d_rem_carrier_phase_in_rad;
00221     float d_phase_step_rad;
00222     float d_carrier_phase_rate_step_rad;
00223
00224     uint32_t d_code_length_samples;
00225     uint32_t d_n_correlators; // number of correlators
00226
00227     // data related to the hardware module and the driver
00228     int32_t d_device_descriptor; // driver descriptor
00229     volatile uint32_t *d_map_base; // driver memory map
00230
00231     // configuration data received from the interface
00232     uint32_t d_correlator_length_samples;
00233
00234     uint32_t d_code_phase_step_chips_num;
00235     uint32_t d_code_phase_rate_step_chips_num;
00236     int32_t d_rem_carr_phase_rad_int;
00237     int32_t d_phase_step_rad_int;
00238     int32_t d_carrier_phase_rate_step_rad_int;

```

```

00239
00240     // PRN codes
00241     int32_t *d_ca_codes;
00242     int32_t *d_data_codes;
00243
00244     // secondary code configuration
00245     uint32_t d_secondary_code_0_length;
00246     uint32_t d_secondary_code_1_length;
00247
00248     bool d_track_pilot;
00249     bool d_secondary_code_enabled;
00250 };
00251
00252
00253 /** \} */
00254 /** \} */
00255 #endif // GNSS_SDR_FPGA_MULTICORRELATOR_H

```

## 11.569 kf\_conf.h

```

00001 /*!
00002  * \file Kf_conf.h
00003  * \brief Class that contains all the configuration parameters for generic tracking block based on a
00004  * Kalman Filter.
00005  * \author Javier Arribas, 2020. jarribas(at)cttc.es
00006  * Class that contains all the configuration parameters for generic tracking block based on a DLL and
00007  * a PLL.
00008  * -----
00009  *
00010  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00011  *
00012  * GNSS-SDR is a software defined Global Navigation
00013  * Satellite Systems receiver
00014  *
00015  * This file is part of GNSS-SDR.
00016  *
00017  * SPDX-License-Identifier: GPL-3.0-or-later
00018  *
00019  * -----
00020  */
00021
00022 #ifndef GNSS_SDR_Kf_CONF_H
00023 #define GNSS_SDR_Kf_CONF_H
00024
00025 #include "configuration_interface.h"
00026 #include <stdint>
00027 #include <string>
00028
00029 class Kf_Conf
00030 {
00031 public:
00032     Kf_Conf();
00033     void SetFromConfiguration(const ConfigurationInterface *configuration, const std::string &role);
00034
00035     std::string item_type;
00036     std::string dump_filename;
00037     double fs_in;
00038     double carrier_lock_th;
00039
00040     // KF statistics
00041     // Measurement covariances (R)
00042     double code_disc_sd_chips;
00043     double carrier_disc_sd_rads;
00044
00045     // System covariances (Q)
00046     double code_phase_sd_chips;
00047     double carrier_phase_sd_rad;
00048     double carrier_freq_sd_hz;
00049     double carrier_freq_rate_sd_hz_s;
00050
00051     // initial Kalman covariance matrix (P)
00052     double init_code_phase_sd_chips;
00053     double init_carrier_phase_sd_rad;
00054     double init_carrier_freq_sd_hz;
00055     double init_carrier_freq_rate_sd_hz_s;
00056
00057     float early_late_space_chips;
00058     float very_early_late_space_chips;
00059     float early_late_space_narrow_chips;
00060     float very_early_late_space_narrow_chips;
00061     float slope;
00062     float spc;

```

```

00063     float y_intercept;
00064     float cn0_smoother_alpha;
00065     float carrier_lock_test_smoother_alpha;
00066     uint32_t pull_in_time_s;
00067     uint32_t bit_synchronization_time_limit_s;
00068     uint32_t vector_length;
00069     uint32_t smoother_length;
00070     int32_t extend_correlation_symbols;
00071     int32_t cn0_samples;
00072     int32_t cn0_smoother_samples;
00073     int32_t carrier_lock_test_smoother_samples;
00074     int32_t cn0_min;
00075     int32_t max_code_lock_fail;
00076     int32_t max_carrier_lock_fail;
00077     char signal[3]{};
00078     char system;
00079     bool track_pilot;
00080     bool enable_doppler_correction;
00081     bool high_dyn;
00082     bool dump;
00083     bool dump_mat;
00084 };
00085
00086 #endif

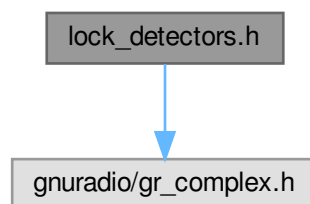
```

## 11.570 lock\_detectors.h File Reference

Interface of a library with a set of code and carrier phase lock detectors.

```
#include <gnuradio/gr_complex.h>
```

Include dependency graph for lock\_detectors.h:



### Functions

- float [cn0\\_svn\\_estimator](#) (const gr\_complex \*Prompt\_buffer, int length, float coh\_integration\_time\_s)  
*cn0\_svn\_estimator is a Carrier-to-Noise (CN0) estimator based on the Signal-to-Noise Variance (SNV) estimator*
- float [cn0\\_m2m4\\_estimator](#) (const gr\_complex \*Prompt\_buffer, int length, float coh\_integration\_time\_s)  
*cn0\_m2m4\_estimator is a Carrier-to-Noise (CN0) estimator based on the Second- and Fourth-Order Moments Method (M2M4)*
- float [carrier\\_lock\\_detector](#) (const gr\_complex \*Prompt\_buffer, int length)  
*A carrier lock detector.*

### 11.570.1 Detailed Description

Interface of a library with a set of code and carrier phase lock detectors.

SNV\_CN0 is a Carrier-to-Noise (CN0) estimator based on the Signal-to-Noise Variance (SNV) estimator [1]. Carrier lock detector using normalised estimate of the cosine of twice the carrier phase error [2].

[1] Marco Pini, Emanuela Falletti and Maurizio Fantino, "Performance Evaluation of C/N0 Estimators using a Real Time GNSS Software Receiver," IEEE 10th International Symposium on Spread Spectrum Techniques and Applications, pp.28-30, August 2008.

[2] Van Dierendonck, A.J. (1996), Global Positioning System: Theory and Applications, Volume I, Chapter 8: GPS Receivers, AJ Systems, Los Altos, CA 94024. Inc.: 329-407.

#### Authors

- Javier Arribas, 2011. jarribas(at)cttc.es
- Luis Esteve, 2012. luis(at)epsilon-formacion.com

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [lock\\_detectors.h](#).

---

## 11.571 lock\_detectors.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file lock_detectors.h
00003  * \brief Interface of a library with a set of code and carrier phase lock detectors.
00004  *
00005  * SNV_CN0 is a Carrier-to-Noise (CN0) estimator
00006  * based on the Signal-to-Noise Variance (SNV) estimator [1].
00007  * Carrier lock detector using normalised estimate of the cosine
00008  * of twice the carrier phase error [2].
00009  *
00010  * [1] Marco Pini, Emanuela Falletti and Maurizio Fantino, "Performance
00011  * Evaluation of C/N0 Estimators using a Real Time GNSS Software Receiver,"
00012  * IEEE 10th International Symposium on Spread Spectrum Techniques and
00013  * Applications, pp.28-30, August 2008.
00014  *
00015  * [2] Van Dierendonck, A.J. (1996), Global Positioning System: Theory and
00016  * Applications,
00017  * Volume I, Chapter 8: GPS Receivers, AJ Systems, Los Altos, CA 94024.
00018  * Inc.: 329-407.
00019  * \authors <ul>
00020  * <li> Javier Arribas, 2011. jarribas(at)cttc.es
00021  * <li> Luis Esteve, 2012. luis(at)epsilon-formacion.com
00022  * </ul>
00023  *
00024  * -----
00025  *
00026  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00027  * This file is part of GNSS-SDR.
00028  *
00029  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00030  * SPDX-License-Identifier: GPL-3.0-or-later
00031  *
00032  * -----
00033  */
00034
00035 #ifndef GNSS_SDR_LOCK_DETECTORS_H
00036 #define GNSS_SDR_LOCK_DETECTORS_H
00037
00038 #include <gnuradio/gr_complex.h>
00039
00040 /** \addtogroup Tracking
00041  * \{ */
00042 /** \addtogroup Tracking_libs
00043  * \{ */
00044
00045
00046 /*! \brief cn0_svn_estimator is a Carrier-to-Noise (CN0) estimator
00047  * based on the Signal-to-Noise Variance (SNV) estimator
00048  *
00049  * Signal-to-Noise (SNR) ( $\rho$ ) estimator using the Signal-to-Noise Variance (SNV) estimator:
00050  * \f{equation}
00051  * \hat{\rho} = \frac{\hat{P}_s}{\hat{P}_n} = \frac{\hat{P}_s}{\hat{P}_{tot} - \hat{P}_s},
00052  * \f{
00053  * where  $\hat{P}_s = \left( \frac{1}{N} \sum_{i=0}^{N-1} |Re(Pc(i))|^2 \right)$  is the estimation of
the signal power,
00054  *  $\hat{P}_{tot} = \frac{1}{N} \sum_{i=0}^{N-1} |Pc(i)|^2$  is the estimator of the total power,  $| \cdot |$ 
is the absolute value,
00055  *  $Re(\cdot)$  stands for the real part of the value, and  $Pc(i)$  is the prompt correlator
output for the sample index i.
00056  *
00057  * The SNR value is converted to CN0 [dB-Hz], taking into account the coherent integration time, using
the following formula:
00058  * \f{equation}
00059  * CN0_{dB} = 10 \cdot \log(\hat{\rho}) - 10 \cdot \log(T_{int}),
00060  * \f{
00061  * where  $T_{int}$  is the coherent integration time, in seconds.
00062  *
```

```

00063 * Ref: Marco Pini, Emanuela Falletti and Maurizio Fantino, "Performance
00064 * Evaluation of C/N0 Estimators using a Real Time GNSS Software Receiver,"
00065 * IEEE 10th International Symposium on Spread Spectrum Techniques and
00066 * Applications, pp.28-30, August 2008.
00067 */
00068 float cn0_svn_estimator(const gr_complex* Prompt_buffer, int length, float coh_integration_time_s);
00069
00070
00071 /*! \brief cn0_m2m4_estimator is a Carrier-to-Noise (CN0) estimator
00072 * based on the Second- and Fourth-Order Moments Method (M2M4)
00073 *
00074 * Signal-to-Noise (SNR) ( $\rho$ ) estimator using the Moments Method:
00075 * \{equation\}
00076 *  $\hat{\rho} = \frac{\sqrt{2} \hat{M}_2^2 - \hat{M}_4}{\hat{M}_2 - \sqrt{2} \hat{M}_4}$ ,
00077 * \{f\}
00078 * where
00079 *  $\hat{M}_2 = \frac{1}{N} \sum_{k=0}^{K-1} |P[k]|^2$ ,  $\hat{M}_4 = \frac{1}{K} \sum_{k=0}^{K-1} |P[k]|^4$ ,  $|P[k]|$  is the absolute value,
00080 * and  $P[k]$  is the prompt correlator output for the sample index k.
00081 *
00082 * The SNR value is converted to CN0 [dB-Hz] taking into account the coherent integration time, using
the following formula:
00083 * \{equation\}
00084 *  $CN0_{dB} = 10 \log(\rho) - 10 \log(T_{int})$ ,
00085 * \{f\}
00086 * where  $T_{int}$  is the coherent integration time, in seconds.
00087 *
00088 * Ref: D. R. Pauluzzi, N. C. Beaulieu, "A comparison of SNR estimation
00089 * techniques for the AWGN channel," IEEE Trans. on Comm., vol. 48,
00090 * no. 10, pp. 1681-1691, Oct. 2000.
00091 */
00092 float cn0_m2m4_estimator(const gr_complex* Prompt_buffer, int length, float coh_integration_time_s);
00093
00094
00095 /*! \brief A carrier lock detector
00096 *
00097 * The Carrier Phase Lock Detector block uses the estimate of the cosine of twice the carrier phase
error is given by
00098 * \{equation\}
00099 *  $C2\phi = \frac{NBD}{NBP}$ ,
00100 * \{f\}
00101 * where  $NBD = (\sum_{i=0}^{N-1} |\text{Im}(Pc(i))|)^2 + (\sum_{i=0}^{N-1} |\text{Re}(Pc(i))|)^2$ ,
00102 *  $NBP = \sum_{i=0}^{N-1} |\text{Im}(Pc(i))|^2 + \sum_{i=0}^{N-1} |\text{Re}(Pc(i))|^2$ , and
00103 *  $Pc(i)$  is the prompt correlator output for the sample index i.
00104 * Ref: Van Dierendonck, A.J. (1996), Global Positioning System: Theory and
00105 * Applications,
00106 * Volume I, Chapter 8: GPS Receivers, AJ Systems, Los Altos, CA 94024.
00107 * Inc.: 329-407.
00108 */
00109 float carrier_lock_detector(const gr_complex* Prompt_buffer, int length);
00110
00111
00112 /** \} */
00113 /** \} */
00114 #endif // GNSS_SDR_LOCK_DETECTORS_H

```

## 11.572 nonlinear\_tracking.h File Reference

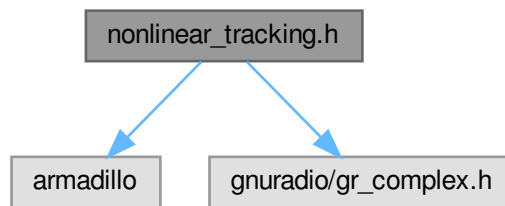
Interface of a library for nonlinear tracking algorithms.

```

#include <armadillo>
#include <gnuradio/gr_complex.h>

```

Include dependency graph for nonlinear\_tracking.h:



## Classes

- class [ModelFunction](#)
- class [CubatureFilter](#)
- class [UnscentedFilter](#)

## 11.572.1 Detailed Description

Interface of a library for nonlinear tracking algorithms.

[CubatureFilter](#) implements the functionality of the Cubature Kalman Filter, which uses multidimensional cubature rules to estimate the time evolution of a nonlinear system. [UnscentedFilter](#) implements an Unscented Kalman Filter which uses Unscented Transform rules to perform a similar estimation.

[1] I Arasaratnam and S Haykin. Cubature kalman filters. IEEE Transactions on Automatic Control, 54(6):1254–1269,2009.

## Authors

- Gerald LaMountain, 2019. gerald(at)ece.neu.edu
- Jordi Vila-Valls 2019. jvila(at)cttc.es

### 11.572.1.1

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [nonlinear\\_tracking.h](#).

## 11.573 nonlinear\_tracking.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file nonlinear_tracking.h
00003  * \brief Interface of a library for nonlinear tracking algorithms
00004  *
00005  * CubatureFilter implements the functionality of the Cubature Kalman
00006  * Filter, which uses multidimensional cubature rules to estimate the
00007  * time evolution of a nonlinear system. UnscentedFilter implements
00008  * an Unscented Kalman Filter which uses Unscented Transform rules to
00009  * perform a similar estimation.
00010  *
00011  * [1] I Arasaratnam and S Haykin. Cubature kalman filters. IEEE
00012  * Transactions on Automatic Control, 54(6):1254-1269,2009.
00013  *
00014  * \authors <ul>
00015  *         <li> Gerald LaMountain, 2019. gerald(at)ece.neu.edu
00016  *         <li> Jordi Vila-Valls 2019. jvila(at)cttc.es
00017  *       </ul>
00018  * -----
00019  *

```

```

00020  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00021  * This file is part of GNSS-SDR.
00022  *
00023  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00024  * SPDX-License-Identifier: GPL-3.0-or-later
00025  *
00026  * -----
00027  */
00028
00029 #ifndef GNSS_SDR_NONLINEAR_TRACKING_H
00030 #define GNSS_SDR_NONLINEAR_TRACKING_H
00031
00032 #if ARMA_NO_BOUND_CHECKING
00033 #define ARMA_NO_DEBUG 1
00034 #endif
00035
00036 #include <armadillo>
00037 #include <gnuradio/gr_complex.h>
00038
00039 /** \addtogroup Tracking
00040  * \{ */
00041 /** \addtogroup Tracking_libs
00042  * \{ */
00043
00044
00045 // Abstract model function
00046 class ModelFunction
00047 {
00048 public:
00049     ModelFunction() {};
00050     virtual arma::vec operator()(const arma::vec& input) = 0;
00051     virtual ~ModelFunction() = default;
00052 };
00053
00054 class CubatureFilter
00055 {
00056 public:
00057     // Constructors and destructors
00058     CubatureFilter();
00059     explicit CubatureFilter(int nx);
00060     CubatureFilter(const arma::vec& x_pred_0, const arma::mat& P_x_pred_0);
00061     ~CubatureFilter() = default;
00062
00063     // Reinitialization function
00064     void initialize(const arma::mat& x_pred_0, const arma::mat& P_x_pred_0);
00065
00066     // Prediction and estimation
00067     void predict_sequential(const arma::vec& x_post, const arma::mat& P_x_post, ModelFunction*
transition_fcn, const arma::mat& noise_covariance);
00068     void update_sequential(const arma::vec& z_upd, const arma::vec& x_pred, const arma::mat& P_x_pred,
ModelFunction* measurement_fcn, const arma::mat& noise_covariance);
00069
00070     // Getters
00071     arma::mat get_x_pred() const;
00072     arma::mat get_P_x_pred() const;
00073     arma::mat get_x_est() const;
00074     arma::mat get_P_x_est() const;
00075
00076 private:
00077     arma::vec x_pred_out;
00078     arma::mat P_x_pred_out;
00079     arma::vec x_est;
00080     arma::mat P_x_est;
00081 };
00082
00083 class UnscentedFilter
00084 {
00085 public:
00086     // Constructors and destructors
00087     UnscentedFilter();
00088     explicit UnscentedFilter(int nx);
00089     UnscentedFilter(const arma::vec& x_pred_0, const arma::mat& P_x_pred_0);
00090     ~UnscentedFilter() = default;
00091
00092     // Reinitialization function
00093     void initialize(const arma::mat& x_pred_0, const arma::mat& P_x_pred_0);
00094
00095     // Prediction and estimation
00096     void predict_sequential(const arma::vec& x_post, const arma::mat& P_x_post, ModelFunction*
transition_fcn, const arma::mat& noise_covariance);
00097     void update_sequential(const arma::vec& z_upd, const arma::vec& x_pred, const arma::mat& P_x_pred,
ModelFunction* measurement_fcn, const arma::mat& noise_covariance);
00098
00099     // Getters
00100     arma::mat get_x_pred() const;
00101     arma::mat get_P_x_pred() const;
00102     arma::mat get_x_est() const;

```



```

00103     arma::mat get_P_x_est() const;
00104
00105 private:
00106     arma::vec x_pred_out;
00107     arma::mat P_x_pred_out;
00108     arma::vec x_est;
00109     arma::mat P_x_est;
00110 };
00111
00112
00113 /** \} */
00114 /** \} */
00115 #endif // GNSS_SDR_NONLINEAR_TRACKING_H

```

## 11.574 tcp\_communication.h File Reference

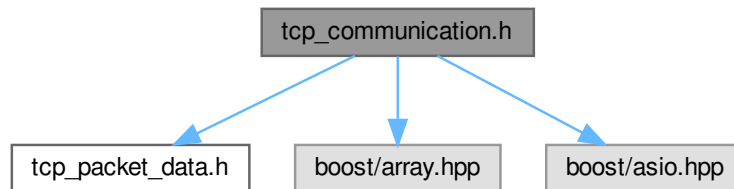
Interface of the TCP communication class.

```
#include "tcp_packet_data.h"
```

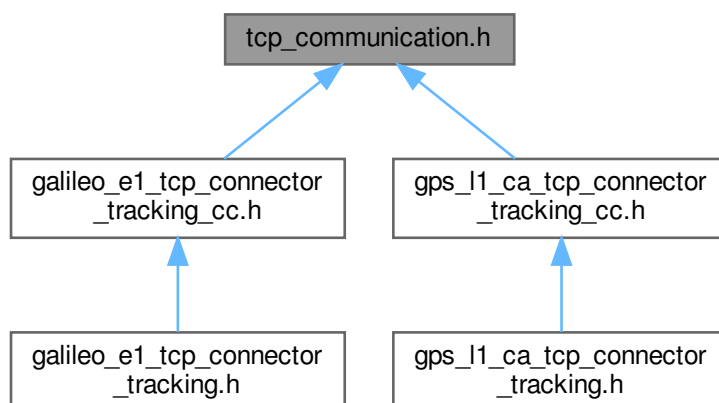
```
#include <boost/array.hpp>
```

```
#include <boost/asio.hpp>
```

Include dependency graph for tcp\_communication.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [Tcp\\_Communication](#)

*TCP communication class.*

## Macros

- `#define NUM_TX_VARIABLES_GALILEO_E1` 13
- `#define NUM_TX_VARIABLES_GPS_L1_CA` 9
- `#define NUM_RX_VARIABLES` 4

## Typedefs

- using `b_io_context` = `boost::asio::io_service`

## 11.574.1 Detailed Description

Interface of the TCP communication class.

### Author

David Pubill, 2011. `dpubill(at)cttc.es`

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [tcp\\_communication.h](#).

---

## 11.575 tcp\_communication.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file tcp_communication.h
00003  * \brief Interface of the TCP communication class
00004  * \author David Pubill, 2011. dpubill(at)cttc.es
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_TCP_COMMUNICATION_H
00019 #define GNSS_SDR_TCP_COMMUNICATION_H
00020
00021 #include "tcp_packet_data.h"
00022 #include <boost/array.hpp>
00023 #include <boost/asio.hpp>
00024
00025 /** \addtogroup Tracking
00026  * \{ */
00027 /** \addtogroup Tracking_libs
00028  * \{ */
00029
00030
00031 #if USE_BOOST_ASIO_IO_CONTEXT
00032 using b_io_context = boost::asio::io_context;
00033 #else
00034 using b_io_context = boost::asio::io_service;
00035 #endif
00036
00037 #define NUM_TX_VARIABLES_GALILEO_E1 13
00038 #define NUM_TX_VARIABLES_GPS_L1_CA 9
00039 #define NUM_RX_VARIABLES 4
00040
00041 /*!
00042  * \brief TCP communication class
00043  */
00044 class Tcp_Communication
00045 {
00046 public:
00047     Tcp_Communication();
00048     ~Tcp_Communication() = default;

```

```

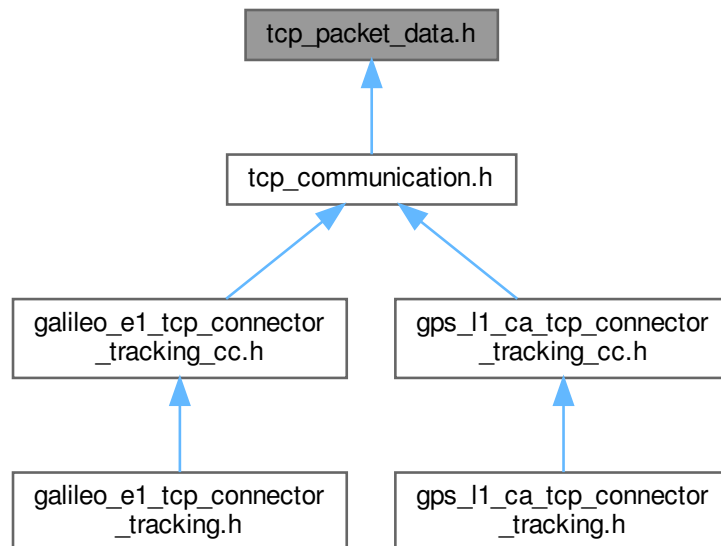
00049
00050     int listen_tcp_connection(size_t d_port_, size_t d_port_ch0_);
00051     void send_receive_tcp_packet_galileo_e1(boost::array<float, NUM_TX_VARIABLES_GALILEO_E1> buf,
00052     Tcp_Packet_Data *tcp_data_);
00052     void send_receive_tcp_packet_gps_l1_ca(boost::array<float, NUM_TX_VARIABLES_GPS_L1_CA> buf,
00053     Tcp_Packet_Data *tcp_data_);
00053     void close_tcp_connection(size_t d_port_);
00054
00055 private:
00056     b_io_context io_context_;
00057     boost::asio::ip::tcp::socket tcp_socket_;
00058 };
00059
00060
00061 /** \} */
00062 /** \} */
00063 #endif

```

## 11.576 tcp\_packet\_data.h File Reference

Interface of the TCP data packet class.

This graph shows which files directly or indirectly include this file:



### Classes

- class [Tcp\\_Packet\\_Data](#)

*Class that implements a TCP data packet.*

### 11.576.1 Detailed Description

Interface of the TCP data packet class.

#### Author

David Pubill, 2011. [dpubill\(at\)cttc.es](mailto:dpubill(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [tcp\\_packet\\_data.h](#).

---

## 11.577 tcp\_packet\_data.h

[Go to the documentation of this file.](#)

```

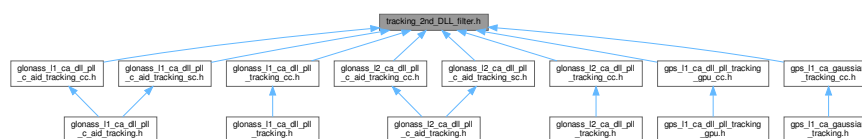
00001  /*!
00002  * \file tcp_packet_data.h
00003  * \brief Interface of the TCP data packet class
00004  * \author David Pubill, 2011. dpubill(at)cttc.es
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_TCP_PACKET_DATA_H
00019 #define GNSS_SDR_TCP_PACKET_DATA_H
00020
00021 /** \addtogroup Tracking
00022  * \{ */
00023 /** \addtogroup Tracking_libs
00024  * \{ */
00025
00026
00027 /*!
00028 * \brief Class that implements a TCP data packet
00029 */
00030 class Tcp_Packet_Data
00031 {
00032 public:
00033     Tcp_Packet_Data() = default;
00034     ~Tcp_Packet_Data() = default;
00035     float proc_pack_code_error = 0.0;
00036     float proc_pack_carr_error = 0.0;
00037     float proc_pack_carrier_doppler_hz = 0.0;
00038 };
00039
00040
00041 /** \} */
00042 /** \} */
00043 #endif

```

## 11.578 tracking\_2nd\_DLL\_filter.h File Reference

Interface of a 2nd order DLL filter for code tracking loop.

This graph shows which files directly or indirectly include this file:



### Classes

- class [Tracking\\_2nd\\_DLL\\_filter](#)

*This class implements a 2nd order DLL filter for code tracking loop.*

### 11.578.1 Detailed Description

Interface of a 2nd order DLL filter for code tracking loop.

## Author

Javier Arribas, 2011. jarribas(at)cttc.es

Class that implements a 2nd order PLL filter for code tracking loop. The algorithm is described in: K.Borre, D.M. Akos, N.Bertelsen, P.Rinder, and S. H. Jensen, A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach, Birkhauser, 2007, Applied and Numerical Harmonic Analysis.

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [tracking\\_2nd\\_DLL\\_filter.h](#).

## 11.579 tracking\_2nd\_DLL\_filter.h

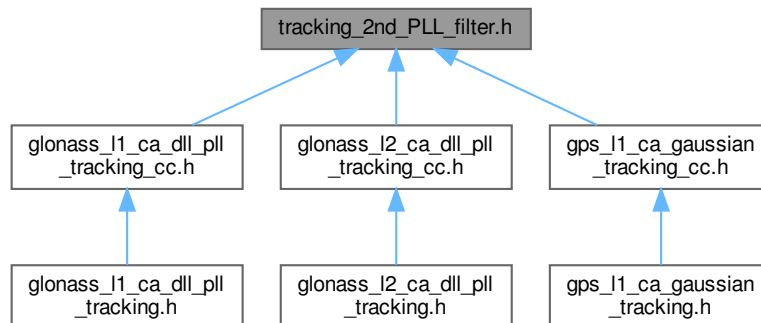
[Go to the documentation of this file.](#)

```
00001  /*!
00002  * \file tracking_2nd_DLL_filter.h
00003  * \brief Interface of a 2nd order DLL filter for code tracking loop.
00004  * \author Javier Arribas, 2011. jarribas(at)cttc.es
00005  *
00006  * Class that implements a 2nd order PLL filter for code tracking loop.
00007  * The algorithm is described in:
00008  * K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S. H. Jensen,
00009  * A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach,
00010  * Birkhauser, 2007, Applied and Numerical Harmonic Analysis.
00011  *
00012  * -----
00013  *
00014  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00015  * This file is part of GNSS-SDR.
00016  *
00017  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00018  * SPDX-License-Identifier: GPL-3.0-or-later
00019  *
00020  * -----
00021  */
00022
00023 #ifndef GNSS_SDR_TRACKING_2ND_DLL_FILTER_H
00024 #define GNSS_SDR_TRACKING_2ND_DLL_FILTER_H
00025
00026 /** \addtogroup Tracking
00027  * \{ */
00028 /** \addtogroup Tracking_libs
00029  * \{ */
00030
00031
00032  /*!
00033  * \brief This class implements a 2nd order DLL filter for code tracking loop.
00034  *
00035  * The algorithm is described in:
00036  * K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S. H. Jensen, A Software-Defined GPS
00037  * and Galileo Receiver. A Single-Frequency Approach,
00038  * Birkhauser, 2007, Applied and Numerical Harmonic Analysis.
00039  */
00040  class Tracking_2nd_DLL_filter
00041  {
00042  public:
00043      Tracking_2nd_DLL_filter();
00044      ~Tracking_2nd_DLL_filter() = default;
00045      explicit Tracking_2nd_DLL_filter(float pdi_code);
00046
00047      void set_DLL_BW(float dll_bw_hz);          //!< Set DLL filter bandwidth [Hz]
00048      void set_pdi(float pdi_code);              //!< Set Summation interval for code [s]
00049      void initialize();                          //!< Start tracking with acquisition information
00050      float get_code_nco(float DLL_discriminator); //!< Numerically controlled oscillator
00051
00052  private:
00053      void calculate_lopp_coef(float* taul, float* tau2, float lbw, float zeta, float k);
00054
00055      // PLL filter parameters
00056      float d_taul_code = 0.0;
00057      float d_tau2_code = 0.0;
00058      float d_pdi_code = 0.0;
00059      float d_dllnoisebandwidth = 0.0;
00060      float d_dlldampingratio = 0.0;
00061      float d_old_code_error = 0.0;
00062      float d_old_code_nco = 0.0;
00063  };
00064
00065
00066  /** \} */
00067  /** \} */
00068 #endif
```

## 11.580 tracking\_2nd\_PLL\_filter.h File Reference

Interface of a 2nd order PLL filter for carrier tracking loop.

This graph shows which files directly or indirectly include this file:



### Classes

- class [Tracking\\_2nd\\_PLL\\_filter](#)

*This class implements a 2nd order PLL filter for carrier tracking loop.*

### 11.580.1 Detailed Description

Interface of a 2nd order PLL filter for carrier tracking loop.

#### Author

Javier Arribas, 2011. jarribas(at)cttc.es

Class that implements 2 order PLL filter for tracking carrier loop. The algorithm is described in K.Borre, D.M. Akos, N.Bertelsen, P.Rinder, and S.H. Jensen, A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach, Birkhauser, 2007, Applied and Numerical Harmonic Analysis.

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [tracking\\_2nd\\_PLL\\_filter.h](#).

## 11.581 tracking\_2nd\_PLL\_filter.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file tracking_2nd_PLL_filter.h
00003  * \brief Interface of a 2nd order PLL filter for carrier tracking loop
00004  * \author Javier Arribas, 2011. jarribas(at)cttc.es
00005  *
00006  * Class that implements 2 order PLL filter for tracking carrier loop.
00007  * The algorithm is described in
00008  * K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S.H. Jensen,
00009  * A Software-Defined GPS and Galileo Receiver. A Single-Frequency Approach,
00010  * Birkhauser, 2007, Applied and Numerical Harmonic Analysis.
00011  *
00012  * -----
00013  *
00014  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00015  * This file is part of GNSS-SDR.
00016  *
00017  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00018  * SPDX-License-Identifier: GPL-3.0-or-later
00019  *
00020  * -----
  
```

```

00021  */
00022
00023 #ifndef GNSS_SDR_TRACKING_2ND_PLL_FILTER_H
00024 #define GNSS_SDR_TRACKING_2ND_PLL_FILTER_H
00025
00026 /** \addtogroup Tracking
00027 * \{ */
00028 /** \addtogroup Tracking_libs
00029 * \{ */
00030
00031
00032 /*!
00033 * \brief This class implements a 2nd order PLL filter for carrier tracking loop.
00034 *
00035 * The algorithm is described in:
00036 * K.Borre, D.M.Akos, N.Bertelsen, P.Rinder, and S. H. Jensen, A Software-Defined GPS and Galileo
Receiver. A Single-Frequency Approach,
00037 * Birkhauser, 2007, Applied and Numerical Harmonic Analysis.
00038 */
00039 class Tracking_2nd_PLL_filter
00040 {
00041 public:
00042     Tracking_2nd_PLL_filter();
00043     ~Tracking_2nd_PLL_filter() = default;
00044     explicit Tracking_2nd_PLL_filter(float pdi_carr);
00045
00046     void set_PLL_BW(float pll_bw_hz);    //!< Set PLL loop bandwidth [Hz]
00047     void set_pdi(float pdi_carr);        //!< Set Summation interval for code [s]
00048     void initialize();
00049     float get_carrier_nco(float PLL_discriminator);
00050
00051 private:
00052     void calculate_lopp_coef(float* tau1, float* tau2, float lbw, float zeta, float k);
00053     // PLL filter parameters
00054     float d_tau1_carr = 0.0;
00055     float d_tau2_carr = 0.0;
00056     float d_pdi_carr = 0.0;
00057     float d_pllnoisebandwidth = 0.0;
00058     float d_plldampingratio = 0.0;
00059     float d_old_carr_error = 0.0;
00060     float d_old_carr_nco = 0.0;
00061 };
00062
00063
00064 /** \} */
00065 /** \} */
00066 #endif

```

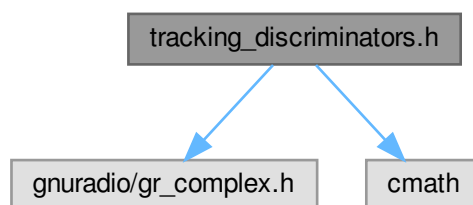
## 11.582 tracking\_discriminators.h File Reference

Interface of a library with a set of code tracking and carrier tracking discriminators.

```
#include <gnuradio/gr_complex.h>
```

```
#include <cmath>
```

Include dependency graph for tracking\_discriminators.h:



### Functions

- [double fill\\_four\\_quadrant\\_atan](#) (gr\_complex prompt\_s1, gr\_complex prompt\_s2, double t1, double t2)

- double **fil\_diff\_atan** (gr\_complex prompt\_s1, gr\_complex prompt\_s2, double t1, double t2)
- double **phase\_unwrap** (double phase\_rad)  
*Phase unwrapping function, input is [rad].*
- double **pll\_four\_quadrant\_atan** (gr\_complex prompt\_s1)  
*PLL four quadrant arctan discriminator.*
- double **pll\_cloop\_two\_quadrant\_atan** (gr\_complex prompt\_s1)  
*PLL Costas loop two quadrant arctan discriminator.*
- double **dll\_nc\_e\_minus\_l\_normalized** (gr\_complex early\_s1, gr\_complex late\_s1, float spc=0.5, float slope=1.0, float y\_intercept=1.0)  
*DLL Noncoherent Early minus Late envelope normalized discriminator.*
- double **dll\_nc\_vemlp\_normalized** (gr\_complex very\_early\_s1, gr\_complex early\_s1, gr\_complex late\_s1, gr\_complex very\_late\_s1)  
*DLL Noncoherent Very Early Minus Late Power (VEMLP) normalized discriminator.*
- template<typename Fun>  
double **CalculateSlope** (Fun &&f, double x)
- template<typename Fun>  
double **CalculateSlopeAbs** (Fun &&f, double x)
- template<typename Fun>  
double **GetYIntercept** (Fun &&f, double x)
- template<typename Fun>  
double **GetYInterceptAbs** (Fun &&f, double x)
- template<int M = 1, int N = M>  
double **SinBocCorrelationFunction** (double offset\_in\_chips)
- template<int M = 1, int N = M>  
double **CosBocCorrelationFunction** (double offset\_in\_chips)

### 11.582.1 Detailed Description

Interface of a library with a set of code tracking and carrier tracking discriminators.

#### Authors

- Javier Arribas, 2011. jarribas(at)cttc.es
- Luis Esteve, 2012. luis(at)epsilon-formacion.com

---

Library with a set of code tracking and carrier tracking discriminators that is used by the tracking algorithms. GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR. Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later Definition in file [tracking\\_discriminators.h](#).

---

## 11.583 tracking\_discriminators.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file tracking_discriminators.h
00003  * \brief Interface of a library with a set of code tracking and carrier
00004  * tracking discriminators.
00005  * \authors <ul>
00006  *         <li> Javier Arribas, 2011. jarribas(at)cttc.es
00007  *         <li> Luis Esteve, 2012. luis(at)epsilon-formacion.com
00008  *         </li>
00009  *
00010  * Library with a set of code tracking and carrier tracking discriminators
00011  * that is used by the tracking algorithms.
00012  *
00013  * -----
00014  *
00015  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00016  * This file is part of GNSS-SDR.
00017  *
00018  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00019  * SPDX-License-Identifier: GPL-3.0-or-later
00020  *
00021  * -----

```



```

00022 */
00023
00024 #ifndef GNSS_SDR_TRACKING_DISCRIMINATORS_H
00025 #define GNSS_SDR_TRACKING_DISCRIMINATORS_H
00026
00027 #include <gnuradio/gr_complex.h>
00028 #include <cmath>
00029
00030 /** \addtogroup Tracking
00031 * \{ */
00032 /** \addtogroup Tracking_libs
00033 * \{ */
00034
00035
00036 /*! brief FLL four quadrant arctan discriminator
00037 *
00038 * FLL four quadrant arctan discriminator:
00039 * \f{equation}
00040 * \frac{\phi_2 - \phi_1}{t_2 - t_1} = \frac{\text{ATAN2}(\text{cross}, \text{dot})}{t_1 - t_2},
00041 * \f{f}
00042 * where \f{cross} = I_{PS1}Q_{PS2} - I_{PS2}Q_{PS1}\f{f} and \f{dot} = I_{PS1}I_{PS2} + Q_{PS1}Q_{PS2}\f{f},
00043 * \f{I}_{PS1}, Q_{PS1}\f{f} are the inphase and quadrature prompt correlator outputs respectively at
00044 * sample time \f{t_1}\f{f}, and
00045 * \f{I}_{PS2}, Q_{PS2}\f{f} are the inphase and quadrature prompt correlator outputs respectively at
00046 * sample time \f{t_2}\f{f}. The output is in [radians/second].
00047 */
00048 double fll_four_quadrant_atan(gr_complex prompt_s1, gr_complex prompt_s2, double t1, double t2);
00049
00050 /*
00051 * FLL differential arctan discriminator:
00052 * \f{equation}
00053 * e_{\text{atan}}(k) = \frac{1}{t_1 - t_2} \text{phase\_unwrap} \left( \tan^{-1} \left( \frac{Q(k)}{I(k)} \right) - \tan^{-1} \left( \frac{Q(k-1)}{I(k-1)} \right) \right)
00054 * \f{f}
00055 * The output is in [radians/second].
00056 */
00057 double fll_diff_atan(gr_complex prompt_s1, gr_complex prompt_s2, double t1, double t2);
00058
00059 /*! \brief Phase unwrapping function, input is [rad]
00060 */
00061 double phase_unwrap(double phase_rad);
00062
00063
00064 /*! \brief PLL four quadrant arctan discriminator
00065 *
00066 * PLL four quadrant arctan discriminator:
00067 * \f{equation}
00068 * \phi = \text{ATAN2}(Q_{PS}, I_{PS}),
00069 * \f{f}
00070 * where \f{I}_{PS1}, Q_{PS1}\f{f} are the inphase and quadrature prompt correlator outputs respectively.
00071 * The output is in [radians].
00072 */
00073 double pll_four_quadrant_atan(gr_complex prompt_s1);
00074
00075 /*! \brief PLL Costas loop two quadrant arctan discriminator
00076 *
00077 * PLL Costas loop two quadrant arctan discriminator:
00078 * \f{equation}
00079 * \phi = \text{ATAN} \left( \frac{Q_{PS}}{I_{PS}} \right),
00080 * \f{f}
00081 * where \f{I}_{PS1}, Q_{PS1}\f{f} are the inphase and quadrature prompt correlator outputs respectively.
00082 * The output is in [radians].
00083 */
00084 double pll_cloop_two_quadrant_atan(gr_complex prompt_s1);
00085
00086 /*! \brief DLL Noncoherent Early minus Late envelope normalized discriminator
00087 *
00088 * DLL Noncoherent Early minus Late envelope normalized discriminator:
00089 * \f{equation}
00090 * \text{error} = \frac{y_{\text{intercept}} - \text{slope} * \epsilon}{\text{slope}} \frac{E-L}{E+L},
00091 * \f{f}
00092 * where \f{E} = \sqrt{I_{ES}^2 + Q_{ES}^2}\f{f} is the Early correlator output absolute value and
00093 * \f{L} = \sqrt{I_{LS}^2 + Q_{LS}^2}\f{f} is the Late correlator output absolute value. The output is in
00094 * [chips].
00095 */
00096 double dll_nc_e_minus_l_normalized(gr_complex early_s1, gr_complex late_s1, float spc = 0.5, float
00097 slope = 1.0, float y_intercept = 1.0);
00098
00099 /*! \brief DLL Noncoherent Very Early Minus Late Power (VEMLP) normalized discriminator
00100 *
00101 * DLL Noncoherent Very Early Minus Late Power (VEMLP) normalized discriminator, using the outputs
00102 * of four correlators, Very Early (VE), Early (E), Late (L) and Very Late (VL):

```

```

00102 * \{equation}
00103 * error=\frac{E-L}{E+L},
00104 * \f}
00105 * where \fE=\sqrt{I_{VE}^2+Q_{VE}^2+I_{E}^2+Q_{E}^2}\f$ and
00106 * \fL=\sqrt{I_{VL}^2+Q_{VL}^2+I_{L}^2+Q_{L}^2}\f$ . The output is in [chips].
00107 */
00108 double dll_nc_vemlp_normalized(gr_complex very_early_s1, gr_complex early_s1, gr_complex late_s1,
    gr_complex very_late_s1);
00109
00110
00111 template <typename Fun>
00112 double CalculateSlope(Fun &&f, double x)
00113 {
00114     static constexpr double dx = 1e-6;
00115
00116     return (f(x + dx / 2.0) - f(x - dx / 2.0)) / dx;
00117 }
00118
00119 template <typename Fun>
00120 double CalculateSlopeAbs(Fun &&f, double x)
00121 {
00122     static constexpr double dx = 1e-6;
00123
00124     return (std::abs(f(x + dx / 2.0)) - std::abs(f(x - dx / 2.0))) / dx;
00125 }
00126
00127 template <typename Fun>
00128 double GetYIntercept(Fun &&f, double x)
00129 {
00130     double slope = CalculateSlope(f, x);
00131     double y1 = f(x);
00132
00133     return y1 - slope * x;
00134 }
00135
00136 template <typename Fun>
00137 double GetYInterceptAbs(Fun &&f, double x)
00138 {
00139     double slope = CalculateSlopeAbs(f, x);
00140     double y1 = std::abs(f(x));
00141     return y1 - slope * x;
00142 }
00143
00144 // SinBocCorrelationFunction and CosBocCorrelationFunction from
00145 // Sousa, F. and Nunes, F., "New Expressions for the Autocorrelation
00146 // Function of BOC GNSS Signals", NAVIGATION - Journal of the Institute
00147 // of Navigation, March 2013.
00148 //
00149 template <int M = 1, int N = M>
00150 double SinBocCorrelationFunction(double offset_in_chips)
00151 {
00152     static constexpr int TWO_P = 2 * M / N;
00153
00154     double abs_tau = std::abs(offset_in_chips);
00155
00156     if (abs_tau > 1.0)
00157     {
00158         return 0.0;
00159     }
00160
00161     int k = static_cast<int>(std::ceil(TWO_P * abs_tau));
00162
00163     double sgn = ((k & 0x01) == 0 ? 1.0 : -1.0); // (-1)^k
00164
00165     return sgn * (2.0 * (k * k - k * TWO_P - k) / TWO_P + 1.0 +
00166         (2 * TWO_P - 2 * k + 1) * abs_tau);
00167 }
00168
00169
00170 template <int M = 1, int N = M>
00171 double CosBocCorrelationFunction(double offset_in_chips)
00172 {
00173     static constexpr int TWO_P = 2 * M / N;
00174
00175     double abs_tau = std::abs(offset_in_chips);
00176
00177     if (abs_tau > 1.0)
00178     {
00179         return 0.0;
00180     }
00181
00182     int k = static_cast<int>(std::floor(2.0 * TWO_P * abs_tau));
00183
00184     if ((k & 0x01) == 0) // k is even
00185     {
00186         double sgn = ((k >> 1) & 0x01 ? -1.0 : 1.0); // (-1)^(k/2)
00187

```

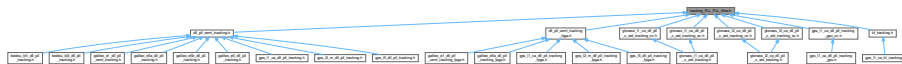
```

00188         return sgn * ((2 * k * TWO_P + 2 * TWO_P - k * k) / (2.0 * TWO_P) + (-2 * TWO_P + k - 1) *
00189             abs_tau);
00190     else
00191     {
00192         double sgn = (((k + 1) >> 1) & 0x01 ? -1.0 : 1.0); // (-1)^((k+1)/2)
00193
00194         return sgn * ((k * k + 2 * k - 2 * k * TWO_P + 1) / (2.0 * TWO_P) + (2 * TWO_P - k - 2) *
00195             abs_tau);
00196     }
00197
00198
00199 /** \} */
00200 /** \} */
00201 #endif // GNSS_SDR_TRACKING_DISCRIMINATORS_H

```

## 11.584 tracking\_FLL\_PLL\_filter.h File Reference

Interface of a hybrid FLL and PLL filter for tracking carrier loop.  
This graph shows which files directly or indirectly include this file:



## Classes

- class `Tracking_FLL_PLL_filter`

*This class implements a hybrid FLL and PLL filter for tracking carrier loop.*

### 11.584.1 Detailed Description

Interface of a hybrid FLL and PLL filter for tracking carrier loop.

**Author**

Javier Arribas, 2011. jarribas(at)cttc.es

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [tracking\\_FLL\\_PLL\\_filter.h](#).

## 11.585 tracking FLL PLL filter.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file tracking_FLL_PLL_filter.h
00003  * \brief Interface of a hybrid FLL and PLL filter for tracking carrier loop
00004  * \author Javier Arribas, 2011. jarribas(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_TRACKING_FLL_PLL_FILTER_H
00018 #define GNSS_SDR_TRACKING_FLL_PLL_FILTER_H
00019
00020 /** \addtogroup Tracking
00021  * \{ */
00022 /** \addtogroup Tracking_libs
00023  * \{ */
00024
00025

```

```

00026 /*!
00027  * \brief This class implements a hybrid FLL and PLL filter for tracking carrier loop
00028  */
00029 class Tracking_FLL_PLL_filter
00030 {
00031 public:
00032     Tracking_FLL_PLL_filter() = default;
00033     ~Tracking_FLL_PLL_filter() = default;
00034     void set_params(float fll_bw_hz, float pll_bw_hz, int order);
00035     void initialize(float d_acq_carrier_doppler_hz);
00036     float get_carrier_error(float FLL_discriminator, float PLL_discriminator, float
        correlation_time_s);
00037
00038 private:
00039     // FLL + PLL filter parameters
00040     float d_pll_w{0.0};
00041     float d_pll_w0p3{0.0};
00042     float d_pll_w0f2{0.0};
00043     float d_pll_x{0.0};
00044     float d_pll_a2{0.0};
00045     float d_pll_w0f{0.0};
00046     float d_pll_a3{0.0};
00047     float d_pll_w0p2{0.0};
00048     float d_pll_b3{0.0};
00049     float d_pll_w0p{0.0};
00050     int d_order{0};
00051 };
00052
00053
00054 /** \} */
00055 /** \} */
00056 #endif

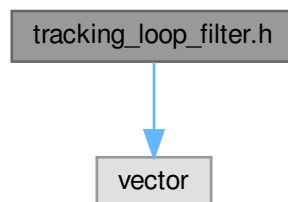
```

## 11.586 tracking\_loop\_filter.h File Reference

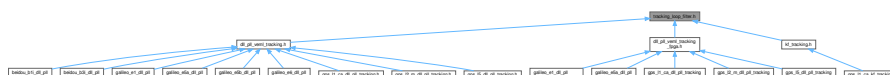
Generic 1st to 3rd order loop filter implementation.

`#include <vector>`

Include dependency graph for tracking\_loop\_filter.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Tracking\\_loop\\_filter](#)

*This class implements a generic 1st, 2nd or 3rd order loop filter.*

## 11.586.1 Detailed Description

Generic 1st to 3rd order loop filter implementation.

### Author

Cillian O'Driscoll, 2015. cillian.odriscoll(at)gmail.com

Class implementing a generic 1st, 2nd or 3rd order loop filter. Based on the bilinear transform of the standard Wiener filter.

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [tracking\\_loop\\_filter.h](#).

## 11.587 tracking\_loop\_filter.h

[Go to the documentation of this file.](#)

```
00001  /*!
00002  * \file tracking_loop_filter.h
00003  * \brief Generic 1st to 3rd order loop filter implementation
00004  * \author Cillian O'Driscoll, 2015. cillian.odriscoll(at)gmail.com
00005  *
00006  * Class implementing a generic 1st, 2nd or 3rd order loop filter. Based
00007  * on the bilinear transform of the standard Wiener filter.
00008  *
00009  * -----
00010  *
00011  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00012  * This file is part of GNSS-SDR.
00013  *
00014  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00015  * SPDX-License-Identifier: GPL-3.0-or-later
00016  *
00017  * -----
00018  */
00019
00020 #ifndef GNSS_SDR_TRACKING_LOOP_FILTER_H
00021 #define GNSS_SDR_TRACKING_LOOP_FILTER_H
00022
00023 #include <vector>
00024
00025 /** \addtogroup Tracking
00026  * \{ */
00027 /** \addtogroup Tracking_libs
00028  * \{ */
00029
00030
00031  /*!
00032  * \brief This class implements a generic 1st, 2nd or 3rd order loop filter
00033  *
00034  */
00035  class Tracking_loop_filter
00036  {
00037  public:
00038      Tracking_loop_filter();
00039      ~Tracking_loop_filter() = default;
00040
00041      Tracking_loop_filter(float update_interval, float noise_bandwidth,
00042                          int loop_order = 2,
00043                          bool include_last_integrator = false);
00044
00045      Tracking_loop_filter(Tracking_loop_filter&&) = default;          //!< Move operator
00046      Tracking_loop_filter& operator=(Tracking_loop_filter&& /*other*/) = default;  //!< Move assignment
00047  private:
00048      float get_noise_bandwidth() const;
00049      float get_update_interval() const;
00050      bool get_include_last_integrator() const;
00051      int get_order() const;
00052
00053      void set_noise_bandwidth(float noise_bandwidth);
00054      void set_update_interval(float update_interval);
00055      void set_include_last_integrator(bool include_last_integrator);
00056      void set_order(int loop_order);
00057
00058      void initialize(float initial_output = 0.0);
00059      float apply(float current_input);
00060
00061  private:
00062      // Compute the filter coefficients:
00063      void update_coefficients();
```

```

00064
00065 // Store the last inputs and outputs:
00066 std::vector<float> d_inputs;
00067 std::vector<float> d_outputs;
00068
00069 // Store the filter coefficients:
00070 std::vector<float> d_input_coefficients;
00071 std::vector<float> d_output_coefficients;
00072
00073 // The noise bandwidth (in Hz)
00074 // Note this is an approximation only valid when the product of this
00075 // number and the update interval (T) is small.
00076 float d_noise_bandwidth;
00077
00078 // Loop update interval
00079 float d_update_interval;
00080
00081 // The loop order:
00082 int d_loop_order;
00083
00084 // The current index in the i/o arrays:
00085 int d_current_index;
00086
00087 // Should the last integrator be included?
00088 bool d_include_last_integrator;
00089 };
00090
00091
00092 /** \} */
00093 /** \} */
00094 #endif // GNSS_SDR_TRACKING_LOOP_FILTER_H

```

## 11.588 acquisition\_interface.h File Reference

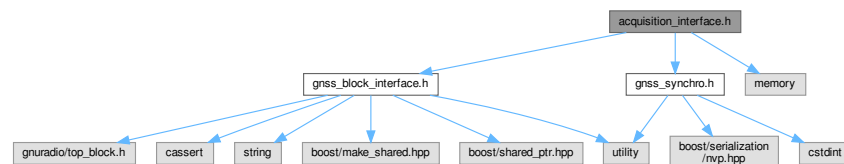
Header file of the interface to an acquisition GNSS block.

```
#include "gnss_block_interface.h"
```

```
#include "gnss_synchro.h"
```

```
#include <memory>
```

Include dependency graph for acquisition\_interface.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [AcquisitionInterface](#)

*This abstract class represents an interface to an acquisition GNSS block.*

### 11.588.1 Detailed Description

Header file of the interface to an acquisition GNSS block.

## Author

Carlos Aviles, 2010. carlos.avilesr(at)gmail.com Luis Esteve, 2011. luis(at)epsilon-formacion.com

This header file contains the interface to an abstract class for acquisition algorithms. Since all its methods are virtual, this class cannot be instantiated directly, and a subclass can only be instantiated directly if all inherited pure virtual methods have been implemented by that class or a parent class.

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [acquisition\\_interface.h](#).

## 11.589 acquisition\_interface.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file acquisition_interface.h
00003  * \brief Header file of the interface to an acquisition GNSS block.
00004  * \author Carlos Aviles, 2010. carlos.avilesr(at)gmail.com
00005  *         Luis Esteve, 2011. luis(at)epsilon-formacion.com
00006  *
00007  * This header file contains the interface to an abstract class
00008  * for acquisition algorithms. Since all its methods are virtual,
00009  * this class cannot be instantiated directly, and a subclass can only be
00010  * instantiated directly if all inherited pure virtual methods have been
00011  * implemented by that class or a parent class.
00012  *
00013  * -----
00014  *
00015  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00016  * This file is part of GNSS-SDR.
00017  *
00018  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00019  * SPDX-License-Identifier: GPL-3.0-or-later
00020  *
00021  * -----
00022  */
00023
00024 #ifndef GNSS_SDR_ACQUISITION_INTERFACE_H
00025 #define GNSS_SDR_ACQUISITION_INTERFACE_H
00026
00027 #include "gnss_block_interface.h"
00028 #include "gnss_synchro.h"
00029 #include <memory>
00030
00031 /** \addtogroup Core
00032  * \{ */
00033 /** \addtogroup GNSS_Block_Interfaces GNSS block interfaces
00034  * GNSS block interfaces.
00035  * \{ */
00036
00037
00038 template <typename Data>
00039 class Concurrent_Queue;
00040
00041 class ChannelFsm;
00042
00043 /*! \brief This abstract class represents an interface to an acquisition GNSS block.
00044 *
00045 * Abstract class for acquisition algorithms. Since all its methods are virtual,
00046 * this class cannot be instantiated directly, and a subclass can only be
00047 * instantiated directly if all inherited pure virtual methods have been
00048 * implemented by that class or a parent class.
00049 */
00050 class AcquisitionInterface : public GNSSBlockInterface
00051 {
00052 public:
00053     virtual void set_gnss_synchro(Gnss_Synchro* gnss_synchro) = 0;
00054     virtual void set_channel(unsigned int channel_id) = 0;
00055     virtual void set_channel_fsm(std::weak_ptr<ChannelFsm> channel_fsm) = 0;
00056     virtual void set_threshold(float threshold) = 0;
00057     virtual void set_doppler_max(unsigned int doppler_max) = 0;
00058     virtual void set_doppler_step(unsigned int doppler_step) = 0;
00059     virtual void set_doppler_center(int doppler_center __attribute__((unused)))
00060     {
00061         return;
00062     }
00063     virtual void init() = 0;
00064     virtual void set_local_code() = 0;
00065     virtual void set_state(int state) = 0;
00066     virtual signed int mag() = 0;
00067     virtual void reset() = 0;
00068     virtual void stop_acquisition() = 0;

```

```

00069     virtual void set_resampler_latency(uint32_t latency_samples) = 0;
00070 };
00071
00072
00073 /** \} */
00074 /** \} */
00075 #endif // GNSS_SDR_ACQUISITION_INTERFACE */

```

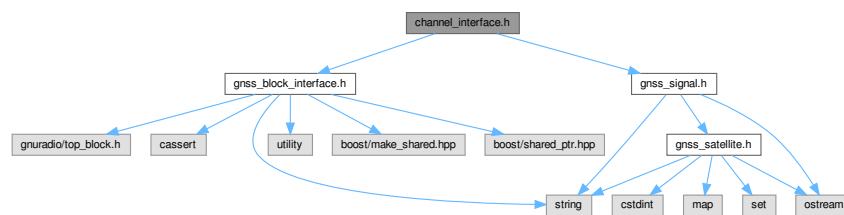
## 11.590 channel\_interface.h File Reference

This class represents an interface to a channel GNSS block.

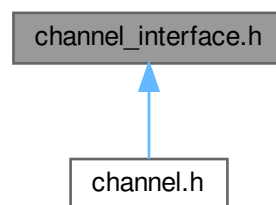
```
#include "gnss_block_interface.h"
```

```
#include "gnss_signal.h"
```

Include dependency graph for channel\_interface.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [ChannelInterface](#)

*This abstract class represents an interface to a channel GNSS block.*

### 11.590.1 Detailed Description

This class represents an interface to a channel GNSS block.

#### Author

Carlos Aviles, 2010. carlos.avilesr(at)gmail.com Luis Esteve, 2011. luis(at)epsilon-formacion.com

Abstract class for channel blocks. Since all its methods are virtual, this class cannot be instantiated directly, and a subclass can only be instantiated directly if all inherited pure virtual methods have been implemented by that class or a parent class.

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [channel\\_interface.h](#).



## 11.591 channel\_interface.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file channel_interface.h
00003  * \brief This class represents an interface to a channel GNSS block.
00004  * \author Carlos Aviles, 2010. carlos.avilesr(at)gmail.com
00005  *         Luis Esteve, 2011. luis(at)epsilon-formacion.com
00006  *
00007  * Abstract class for channel blocks. Since all its methods are virtual,
00008  * this class cannot be instantiated directly, and a subclass can only be
00009  * instantiated directly if all inherited pure virtual methods have been
00010  * implemented by that class or a parent class.
00011  *
00012  * -----
00013  *
00014  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00015  * This file is part of GNSS-SDR.
00016  *
00017  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00018  * SPDX-License-Identifier: GPL-3.0-or-later
00019  *
00020  * -----
00021  */
00022
00023 #ifndef GNSS_SDR_CHANNEL_INTERFACE_H
00024 #define GNSS_SDR_CHANNEL_INTERFACE_H
00025
00026 #include "gnss_block_interface.h"
00027 #include "gnss_signal.h"
00028
00029 /** \addtogroup Core
00030  *  \{ */
00031 /** \addtogroup GNSS_Block_Interfaces
00032  *  \{ */
00033
00034
00035  /*!
00036  * \brief This abstract class represents an interface to a channel GNSS block.
00037  *
00038  * Abstract class for channel blocks. Since all its methods are pure virtual,
00039  * this class cannot be instantiated directly, and a subclass can only be
00040  * instantiated directly if all inherited pure virtual methods have been
00041  * implemented by that class or a parent class.
00042  */
00043  class ChannelInterface : public GNSSBlockInterface
00044  {
00045  public:
00046      virtual gr::basic_block_sptr get_left_block_trk() = 0;
00047      virtual gr::basic_block_sptr get_right_block_trk() = 0;
00048      virtual gr::basic_block_sptr get_left_block_acq() = 0;
00049      virtual gr::basic_block_sptr get_right_block_acq() = 0;
00050      virtual gr::basic_block_sptr get_left_block() = 0;
00051      virtual gr::basic_block_sptr get_right_block() = 0;
00052      virtual Gnss_Signal get_signal() = 0;
00053      virtual void start_acquisition() = 0;
00054      virtual void assist_acquisition_doppler(double Carrier_Doppler_hz) = 0;
00055      virtual void stop_channel() = 0;
00056      virtual void set_signal(const Gnss_Signal&) = 0;
00057  };
00058
00059
00060  /** \} */
00061  /** \} */
00062  #endif // GNSS_SDR_CHANNEL_INTERFACE_H

```

## 11.592 configuration\_interface.h File Reference

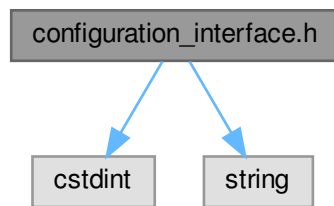
This class represents an interface to configuration parameters.

```

#include <cstdint>
#include <string>

```

Include dependency graph for configuration\_interface.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [ConfigurationInterface](#)

*This abstract class represents an interface to configuration parameters.*

### 11.592.1 Detailed Description

This class represents an interface to configuration parameters.

#### Author

Carlos Aviles, 2010. carlos.avilesr(at)gmail.com

The interface defines an accessor method that gets a parameter name as input and returns the value of this parameter, a string, as output. Property names are defined here. This is an abstract class for interfaces.

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [configuration\\_interface.h](#).

## 11.593 configuration\_interface.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file configuration_interface.h
00003  * \brief This class represents an interface to configuration parameters.
00004  * \author Carlos Aviles, 2010. carlos.avilesr(at)gmail.com
00005  *
00006  * The interface defines an accessor method that gets a parameter name as input
00007  * and returns the value of this parameter, a string, as output.
00008  * Property names are defined here. This is an abstract class for interfaces.
00009  *
00010  * -----
00011  *
00012  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00013  * This file is part of GNSS-SDR.
00014  *
00015  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00016  * SPDX-License-Identifier: GPL-3.0-or-later
00017  *
00018  * -----
00019  */
00020
00021 #ifndef GNSS_SDR_CONFIGURATION_INTERFACE_H

```

```

00022 #define GNSS_SDR_CONFIGURATION_INTERFACE_H
00023
00024 #include <stdint>
00025 #include <string>
00026
00027 /** \addtogroup Core
00028 * \{ */
00029 /** \addtogroup GNSS_Block_Interfaces
00030 * \{ */
00031
00032
00033 /*!
00034 * \brief This abstract class represents an interface to configuration parameters.
00035 *
00036 * The interface defines an accessor method that gets a parameter name as input
00037 * and returns the value of this parameter, a string, as output.
00038 * Property names are defined here. This is an abstract class for interfaces.
00039 * Since all its methods are virtual,
00040 * this class cannot be instantiated directly, and a subclass can only be
00041 * instantiated directly if all inherited pure virtual methods have been
00042 * implemented by that class or a parent class.
00043 */
00044 class ConfigurationInterface
00045 {
00046 public:
00047     virtual ~ConfigurationInterface() = default;
00048     virtual std::string property(std::string property_name, std::string default_value) const = 0;
00049     virtual bool property(std::string property_name, bool default_value) const = 0;
00050     virtual int64_t property(std::string property_name, int64_t default_value) const = 0;
00051     virtual uint64_t property(std::string property_name, uint64_t default_value) const = 0;
00052     virtual int32_t property(std::string property_name, int32_t default_value) const = 0;
00053     virtual uint32_t property(std::string property_name, uint32_t default_value) const = 0;
00054     virtual int16_t property(std::string property_name, int16_t default_value) const = 0;
00055     virtual uint16_t property(std::string property_name, uint16_t default_value) const = 0;
00056     virtual float property(std::string property_name, float default_value) const = 0;
00057     virtual double property(std::string property_name, double default_value) const = 0;
00058     virtual void set_property(std::string property_name, std::string value) = 0;
00059 };
00060
00061
00062 /** \} */
00063 /** \} */
00064 #endif // GNSS_SDR_CONFIGURATION_INTERFACE_H

```

## 11.594 gnss\_block\_interface.h File Reference

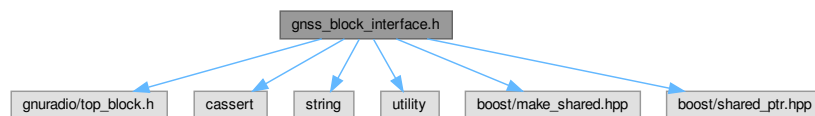
This interface represents a GNSS block.

```

#include <gnuradio/top_block.h>
#include <cassert>
#include <string>
#include <utility>
#include <boost/make_shared.hpp>
#include <boost/shared_ptr.hpp>

```

Include dependency graph for gnss\_block\_interface.h:



### Classes

- class [GNSSBlockInterface](#)

*This abstract class represents an interface to GNSS blocks.*

### Typedefs

- template<typename T>

```
using gnss_shared_ptr = boost::shared_ptr<T>
```

## Functions

- `template<typename C, typename... Args>`  
`gnss_shared_ptr< C > gnss_make_shared (Args &&... args)`

### 11.594.1 Detailed Description

This interface represents a GNSS block.

#### Author

Carlos Aviles, 2010. [carlos.avilesr\(at\)gmail.com](mailto:carlos.avilesr(at)gmail.com)

Abstract class for GNSS block interfaces. Since all its methods are virtual, this class cannot be instantiated directly, and a subclass can only be instantiated directly if all inherited pure virtual methods have been implemented by that class or a parent class.

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

---

Definition in file [gnss\\_block\\_interface.h](#).

## 11.595 gnss\_block\_interface.h

[Go to the documentation of this file.](#)

```
00001  /*!
00002  * \file gnss_block_interface.h
00003  * \brief This interface represents a GNSS block.
00004  * \author Carlos Aviles, 2010. carlos.avilesr\(at\)gmail.com
00005  *
00006  * Abstract class for GNSS block interfaces. Since all its methods are virtual,
00007  * this class cannot be instantiated directly, and a subclass can only be
00008  * instantiated directly if all inherited pure virtual methods have been
00009  * implemented by that class or a parent class.
00010  *
00011  * -----
00012  *
00013  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00014  * This file is part of GNSS-SDR.
00015  *
00016  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00017  * SPDX-License-Identifier: GPL-3.0-or-later
00018  *
00019  * -----
00020  */
00021
00022
00023 #ifndef GNSS_SDR_GNSS_BLOCK_INTERFACE_H
00024 #define GNSS_SDR_GNSS_BLOCK_INTERFACE_H
00025
00026 #include <gnuradio/top_block.h>
00027 #include <cassert>
00028 #include <string>
00029 #include <utility> // for std::forward
00030
00031 /** \addtogroup Core
00032  * \{ */
00033 /** \addtogroup GNSS_Block_Interfaces
00034  * \{ */
00035
00036 // clang-format off
00037 #if GNURADIO_USES_STD_POINTERS
00038 #include <memory>
00039 template <typename T>
00040 using gnss_shared_ptr = std::shared_ptr<T>;
00041 template <typename C, typename... Args>
00042 gnss_shared_ptr<C> gnss_make_shared(Args &&... args)
00043 {
00044     return std::make_shared<C>(std::forward<Args>(args) ...);
00045 }
00046 #else
00047 #include <boost/make_shared.hpp>
00048 #include <boost/shared_ptr.hpp>
00049 template <typename T>
00050 using gnss_shared_ptr = boost::shared_ptr<T>;
00051 template <typename C, typename... Args>
```

```

00052 gnss_shared_ptr<C> gnss_make_shared(Args &&... args)
00053 {
00054     return boost::make_shared<C>(std::forward<Args>(args)...);
00055 }
00056 #endif
00057 // clang-format on
00058
00059
00060 /*!
00061 * \brief This abstract class represents an interface to GNSS blocks.
00062 *
00063 * Abstract class for GNSS block interfaces. Since all its methods are virtual,
00064 * this class cannot be instantiated directly, and a subclass can only be
00065 * instantiated directly if all inherited pure virtual methods have been
00066 * implemented by that class or a parent class.
00067 */
00068 class GNSSBlockInterface
00069 {
00070 public:
00071     virtual ~GNSSBlockInterface() = default;
00072     virtual std::string role() = 0;
00073     virtual std::string implementation() = 0;
00074     virtual size_t item_size() = 0;
00075     virtual void connect(gr::top_block_sptr top_block) = 0;
00076     virtual void disconnect(gr::top_block_sptr top_block) = 0;
00077
00078     virtual gr::basic_block_sptr get_left_block() = 0;
00079     virtual gr::basic_block_sptr get_right_block() = 0;
00080
00081     virtual gr::basic_block_sptr get_left_block(int RF_channel)
00082     {
00083         assert(RF_channel >= 0);
00084         if (RF_channel == 0)
00085         {
00086             }; // avoid unused param warning
00087         return nullptr; // added to support raw array access (non pure virtual to allow left
unimplemented)= 0;
00088     }
00089     virtual gr::basic_block_sptr get_right_block(int RF_channel)
00090     {
00091         assert(RF_channel >= 0);
00092         if (RF_channel == 0)
00093         {
00094             }; // avoid unused param warning
00095         return nullptr; // added to support raw array access (non pure virtual to allow left
unimplemented)= 0;
00096     }
00097
00098     /*!
00099     * \brief Start the flow of samples if needed.
00100     */
00101     virtual void start() {};
00102 };
00103
00104
00105 /** \} */
00106 /** \} */
00107 #endif // GNSS_SDR_GNSS_BLOCK_INTERFACE_H

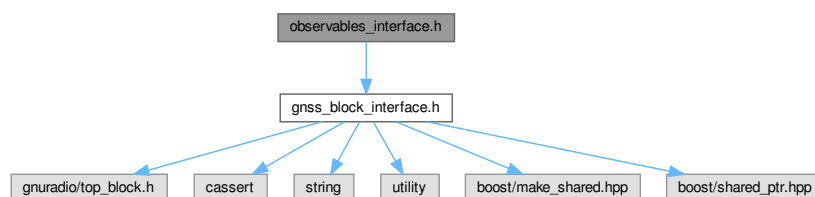
```

## 11.596 observables\_interface.h File Reference

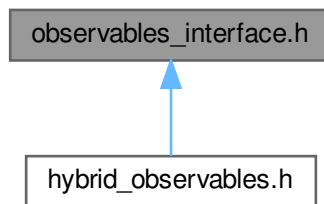
This class represents an interface to an Observables block.

#include "gnss\_block\_interface.h"

Include dependency graph for observables\_interface.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [ObservablesInterface](#)

*This abstract class represents an interface to an observables block.*

## 11.596.1 Detailed Description

This class represents an interface to an Observables block.

### Author

Javier Arribas, 2011. jarribas(at)cttc.es

Abstract class for Observables modules. Since all its methods are virtual, this class cannot be instantiated directly, and a subclass can only be instantiated directly if all inherited pure virtual methods have been implemented by that class or a parent class.

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [observables\\_interface.h](#).

---

## 11.597 observables\_interface.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file observables_interface.h
00003  * \brief This class represents an interface to an Observables block.
00004  * \author Javier Arribas, 2011. jarribas(at)cttc.es
00005  *
00006  * Abstract class for Observables modules. Since all its methods are virtual,
00007  * this class cannot be instantiated directly, and a subclass can only be
00008  * instantiated directly if all inherited pure virtual methods have been
00009  * implemented by that class or a parent class.
00010  *
00011  * -----
00012  *
00013  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00014  * This file is part of GNSS-SDR.
00015  *
00016  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00017  * SPDX-License-Identifier: GPL-3.0-or-later
00018  *
00019  * -----
00020  */
00021
00022
00023 #ifndef GNSS_SDR_OBSERVABLES_INTERFACE_H
00024 #define GNSS_SDR_OBSERVABLES_INTERFACE_H
00025
00026 #include "gnss_block_interface.h"
00027
00028 /** \addtogroup Core
00029  * \{ */
  
```

## 11.598 pvt\_interface.h File Reference

```
#include "galileo_almanac.h"
#include "galileo_ephemeris.h"
#include "gnss_block_interface.h"
#include "gps_almanac.h"
#include "gps_ephemeris.h"
#include <map>
```

```

graph TD
    prf_interface.h --> galileo_almanac.h
    prf_interface.h --> gps_almanac.h
    prf_interface.h --> galileo_ephemeris.h
    prf_interface.h --> gps_ephemeris.h
    prf_interface.h --> gnss_block_interface.h
    galileo_almanac.h --> gnss_almanac.h
    gps_almanac.h --> gnss_almanac.h
    galileo_ephemeris.h --> gnss_ephemeris.h
    gps_ephemeris.h --> gnss_ephemeris.h
    gnss_almanac.h --> boost_serialization_fix_hpp[boost/serialization/fix.hpp]
    gnss_ephemeris.h --> boost_serialization_fix_hpp
    gnss_ephemeris.h --> array
    gnss_ephemeris.h --> cstdint
    gnss_block_interface.h --> map
    gnss_block_interface.h --> string
    gnss_block_interface.h --> boost_shared_ptr_hpp[boost/shared_ptr.hpp]
    gnss_block_interface.h --> gnuradio_top_block_h[gnuradio/top_block.h]
    gnss_block_interface.h --> cassert
    gnss_block_interface.h --> utility
    gnss_block_interface.h --> boost_make_shared_hpp[boost/make_shared.hpp]
  
```

```
graph BT; gnss_flowgraph.h --> pvt_interface.h; rtklib_pvt.h --> pvt_interface.h
```

The diagram illustrates the relationship between three header files. At the top is a grey box labeled `pvt_interface.h`. Below it are two white boxes: `gnss_flowgraph.h` on the left and `rtklib_pvt.h` on the right. Blue arrows point from both `gnss_flowgraph.h` and `rtklib_pvt.h` up to `pvt_interface.h`, indicating that both implement the interface defined in `pvt_interface.h`.

- class PvtInterface

Generated by Doxygen

## 11.598.1 Detailed Description

This class represents an interface to a PVT block.

### Author

Javier Arribas, 2011. jarribas(at)cttc.es

Abstract class for PVT solvers. Since all its methods are virtual, this class cannot be instantiated directly, and a subclass can only be instantiated directly if all inherited pure virtual methods have been implemented by that class or a parent class.

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [pvt\\_interface.h](#).

## 11.599 pvt\_interface.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file pvt_interface.h
00003  * \brief This class represents an interface to a PVT block.
00004  * \author Javier Arribas, 2011. jarribas(at)cttc.es
00005  *
00006  * Abstract class for PVT solvers. Since all its methods are virtual,
00007  * this class cannot be instantiated directly, and a subclass can only be
00008  * instantiated directly if all inherited pure virtual methods have been
00009  * implemented by that class or a parent class.
00010  *
00011  * -----
00012  *
00013  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00014  * This file is part of GNSS-SDR.
00015  *
00016  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00017  * SPDX-License-Identifier: GPL-3.0-or-later
00018  *
00019  * -----
00020  */
00021
00022
00023 #ifndef GNSS_SDR_PVT_INTERFACE_H
00024 #define GNSS_SDR_PVT_INTERFACE_H
00025
00026 #include "galileo_almanac.h"
00027 #include "galileo_ephemeris.h"
00028 #include "gnss_block_interface.h"
00029 #include "gps_almanac.h"
00030 #include "gps_ephemeris.h"
00031 #include <map>
00032
00033 /** \addtogroup Core
00034  * \{ */
00035 /** \addtogroup GNSS_Block_Interfaces
00036  * \{ */
00037
00038
00039 /*!
00040  * \brief This class represents an interface to a PVT block.
00041  *
00042  * Abstract class for PVT interfaces, derived from GNSSBlockInterface.
00043  * Since all its methods are virtual,
00044  * this class cannot be instantiated directly, and a subclass can only be
00045  * instantiated directly if all inherited pure virtual methods have been
00046  * implemented by that class or a parent class.
00047  */
00048 class PvtInterface : public GNSSBlockInterface
00049 {
00050 public:
00051     virtual void reset() = 0;
00052     virtual void clear_ephemeris() = 0;
00053     virtual std::map<int, Gps_Ephemeris> get_gps_ephemeris() const = 0;
00054     virtual std::map<int, Galileo_Ephemeris> get_galileo_ephemeris() const = 0;
00055     virtual std::map<int, Gps_Almanac> get_gps_almanac() const = 0;
00056     virtual std::map<int, Galileo_Almanac> get_galileo_almanac() const = 0;
00057
00058     virtual bool get_latest_PVT(double* longitude_deg,
00059                                double* latitude_deg,
00060                                double* height_m,
00061                                double* ground_speed_kmh,
00062                                double* course_over_ground_deg,
```



```

00063         time_t* UTC_time) = 0;
00064     };
00065
00066
00067     /** \} */
00068     /** \} */
00069 #endif // GNSS_SDR_PVT_INTERFACE_H

```

## 11.600 signal\_source\_interface.h

```

00001  /*!
00002   * \signal_source_interface.h
00003   * \brief Header file of the interface to a signal_source GNSS block.
00004   * \author Jim Melton, 2020. jim.melton(at)sncorp.com
00005   *
00006   * This header file contains the interface to an abstract class for
00007   * signal sources. Since all its methods are virtual, this class
00008   * cannot be instantiated directly, and a subclass can only be
00009   * instantiated directly if all inherited pure virtual methods have
00010   * been implemented by that class or a parent class.
00011   *
00012   * -----
00013   *
00014   * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00015   *
00016   * GNSS-SDR is a software defined Global Navigation
00017   * Satellite Systems receiver
00018   *
00019   * This file is part of GNSS-SDR.
00020   *
00021   * SPDX-License-Identifier: GPL-3.0-or-later
00022   *
00023   * -----
00024   */
00025
00026 #ifndef GNSS_SDR_SIGNAL_SOURCE_INTERFACE_H
00027 #define GNSS_SDR_SIGNAL_SOURCE_INTERFACE_H
00028
00029 #include "gnss_block_interface.h"
00030
00031 #if USE_GLOG_AND_GFLAGS
00032 #include <glog/logging.h>
00033 #else
00034 #include <absl/log/log.h>
00035 #endif
00036
00037 /** \addtogroup Core
00038  * \{ */
00039 /** \addtogroup GNSS_Block_Interfaces GNSS block interfaces
00040  * GNSS block interfaces.
00041  * \{ */
00042
00043  /*! \brief This abstract class represents an interface to signal_source GNSS block.
00044   *
00045   * Abstract class for signal sources. Since all its methods are virtual,
00046   * this class cannot be instantiated directly, and a subclass can only be
00047   * instantiated directly if all inherited pure virtual methods have been
00048   * implemented by that class or a parent class.
00049   */
00050
00051 class SignalSourceInterface : public GNSSBlockInterface
00052 {
00053 public:
00054     virtual size_t getRfChannels() const = 0;
00055
00056 protected:
00057     SignalSourceInterface()
00058     {
00059         VLOG(1) << "SignalSourceInterface: " << this << " ctor";
00060     }
00061
00062 public: // required for polymorphic destruction
00063     ~SignalSourceInterface()
00064     {
00065         VLOG(1) << "SignalSourceInterface: " << this << " dtor";
00066     }
00067 };
00068
00069
00070 #endif

```

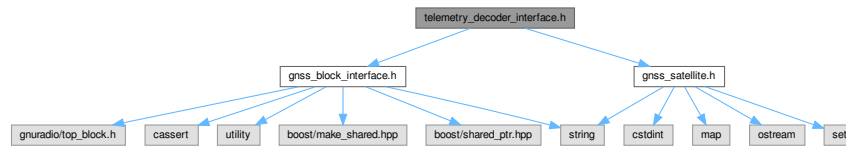
## 11.601 telemetry\_decoder\_interface.h File Reference

This class represents an interface to a telemetry decoder block.

```
#include "gnss_block_interface.h"
```

```
#include "gnss_satellite.h"
```

Include dependency graph for telemetry\_decoder\_interface.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [TelemetryDecoderInterface](#)

*This abstract class represents an interface to a navigation GNSS block.*

### 11.601.1 Detailed Description

This class represents an interface to a telemetry decoder block.

#### Author

Javier Arribas, 2011. jarribas(at)cttc.es

Abstract class for telemetry decoders. Since all its methods are virtual, this class cannot be instantiated directly, and a subclass can only be instantiated directly if all inherited pure virtual methods have been implemented by that class or a parent class.

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [telemetry\\_decoder\\_interface.h](#).

## 11.602 telemetry\_decoder\_interface.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file telemetry_decoder_interface.h
00003  * \brief This class represents an interface to a telemetry decoder block.
00004  * \author Javier Arribas, 2011. jarribas(at)cttc.es
00005  *
00006  * Abstract class for telemetry decoders. Since all its methods are virtual,
00007  * this class cannot be instantiated directly, and a subclass can only be
00008  * instantiated directly if all inherited pure virtual methods have been
00009  * implemented by that class or a parent class.
00010  *
00011  * -----
00012  *
00013  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00014  * This file is part of GNSS-SDR.
00015  *
00016  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00017  * SPDX-License-Identifier: GPL-3.0-or-later
00018  *
00019  * -----
00020  */
00021

```

```

00022
00023 #ifndef GNSS_SDR_TELEMETRY_DECODER_INTERFACE_H
00024 #define GNSS_SDR_TELEMETRY_DECODER_INTERFACE_H
00025
00026 #include "gnss_block_interface.h"
00027 #include "gnss_satellite.h"
00028
00029 /** \addtogroup Core
00030 * \{ */
00031 /** \addtogroup GNSS_Block_Interfaces
00032 * \{ */
00033
00034
00035 /**
00036 * \brief This abstract class represents an interface to a navigation GNSS block.
00037 *
00038 * Abstract class for navigation interfaces. Since all its methods are virtual,
00039 * this class cannot be instantiated directly, and a subclass can only be
00040 * instantiated directly if all inherited pure virtual methods have been
00041 * implemented by that class or a parent class.
00042 */
00043 class TelemetryDecoderInterface : public GNSSBlockInterface
00044 {
00045 public:
00046     virtual void reset() = 0;
00047     virtual void set_satellite(const Gnss_Satellite& sat) = 0;
00048     virtual void set_channel(int channel) = 0;
00049 };
00050
00051
00052 /** \} */
00053 /** \} */
00054 #endif // GNSS_SDR_TELEMETRY_DECODER_INTERFACE_H

```

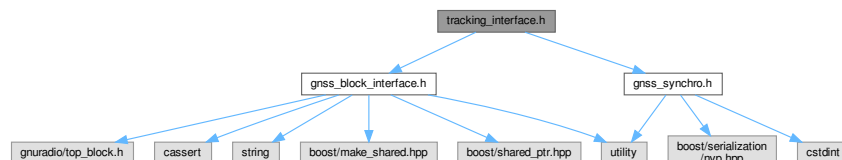
## 11.603 tracking\_interface.h File Reference

This class represents an interface to a tracking block.

```
#include "gnss_block_interface.h"
```

```
#include "gnss_synchro.h"
```

Include dependency graph for tracking\_interface.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [TrackingInterface](#)

*This abstract class represents an interface to a tracking block.*

### 11.603.1 Detailed Description

This class represents an interface to a tracking block.

**Author**

Carlos Aviles, 2010. carlos.avilesr(at)gmail.com

Abstract class for tracking interfaces. Since all its methods are virtual, this class cannot be instantiated directly, and a subclass can only be instantiated directly if all inherited pure virtual methods have been implemented by that class or a parent class.

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [tracking\\_interface.h](#).

## 11.604 tracking\_interface.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file tracking_interface.h
00003  * \brief This class represents an interface to a tracking block.
00004  * \author Carlos Aviles, 2010. carlos.avilesr(at)gmail.com
00005  *
00006  * Abstract class for tracking interfaces. Since all its methods are virtual,
00007  * this class cannot be instantiated directly, and a subclass can only be
00008  * instantiated directly if all inherited pure virtual methods have been
00009  * implemented by that class or a parent class.
00010  *
00011  * -----
00012  *
00013  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00014  * This file is part of GNSS-SDR.
00015  *
00016  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00017  * SPDX-License-Identifier: GPL-3.0-or-later
00018  *
00019  * -----
00020  */
00021
00022
00023 #ifndef GNSS_SDR_TRACKING_INTERFACE_H
00024 #define GNSS_SDR_TRACKING_INTERFACE_H
00025
00026 #include "gnss_block_interface.h"
00027 #include "gnss_synchro.h"
00028
00029 /** \addtogroup Core
00030  * \{ */
00031 /** \addtogroup GNSS_Block_Interfaces
00032  * \{ */
00033
00034
00035 template <typename Data>
00036 class Concurrent_Queue;
00037
00038 /*!
00039 * \brief This abstract class represents an interface to a tracking block.
00040 *
00041 * Abstract class for tracking interfaces. Since all its methods are virtual,
00042 * this class cannot be instantiated directly, and a subclass can only be
00043 * instantiated directly if all inherited pure virtual methods have been
00044 * implemented by that class or a parent class.
00045 *
00046 */
00047 class TrackingInterface : public GNSSBlockInterface
00048 {
00049 public:
00050     virtual void start_tracking() = 0;
00051     virtual void stop_tracking() = 0;
00052     virtual void set_gnss_synchro(Gnss_Synchro* gnss_synchro) = 0;
00053     virtual void set_channel(unsigned int channel) = 0;
00054 };
00055
00056
00057 /** \} */
00058 /** \} */
00059 #endif // GNSS_SDR_TRACKING_INTERFACE_H

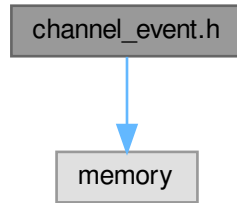
```

## 11.605 channel\_event.h File Reference

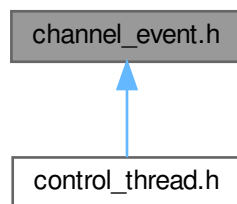
Class that defines a channel event.

```
#include <memory>
```

Include dependency graph for channel\_event.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Channel\\_Event](#)

## Typedefs

- using [channel\\_event\\_sptr](#) = std::shared\_ptr<[Channel\\_Event](#)>

## Functions

- channel\_event\_sptr **channel\_event\_make** (int channel\_id, int event\_type)

## 11.605.1 Detailed Description

Class that defines a channel event.

Author

Javier Arribas, 2019. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [channel\\_event.h](#).

---

## 11.606 channel\_event.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file channel_event.h
00003  * \brief Class that defines a channel event
00004  * \author Javier Arribas, 2019. jarribas(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_CHANNEL_EVENT_H
00018 #define GNSS_SDR_CHANNEL_EVENT_H
00019
00020 #include <memory>
00021
00022 /** \addtogroup Core
00023  * \{ */
00024 /** \addtogroup Core_Receiver_Library core_libs
00025  * \{ */
00026
00027
00028 class Channel_Event;
00029
00030 using channel_event_sptr = std::shared_ptr<Channel_Event>;
00031
00032 channel_event_sptr channel_event_make(int channel_id, int event_type);
00033
00034 class Channel_Event
00035 {
00036 public:
00037     int channel_id;
00038     int event_type;
00039
00040 private:
00041     friend channel_event_sptr channel_event_make(int channel_id, int event_type);
00042     Channel_Event(int channel_id_, int event_type_);
00043 };
00044
00045
00046 /** \} */
00047 /** \} */
00048 #endif // GNSS_SDR_CHANNEL_EVENT_H

```

## 11.607 channel\_status\_msg\_receiver.h File Reference

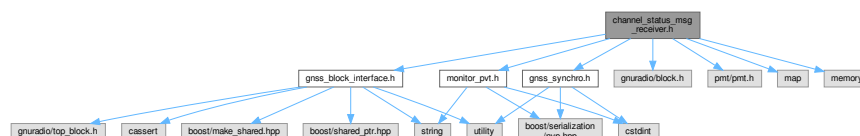
GNU Radio block that receives asynchronous channel messages from acquisition and tracking blocks.

```

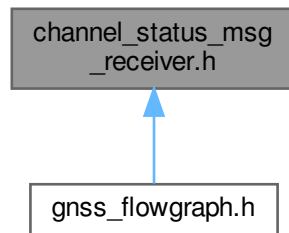
#include "gnss_block_interface.h"
#include "gnss_synchro.h"
#include "monitor_pvt.h"
#include <gnuradio/block.h>
#include <pmt/pmt.h>
#include <map>
#include <memory>

```

Include dependency graph for channel\_status\_msg\_receiver.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [channel\\_status\\_msg\\_receiver](#)  
GNU Radio block that receives asynchronous channel messages from tlm blocks.

## Typedefs

- using [channel\\_status\\_msg\\_receiver\\_sptr](#) = gnss\_shared\_ptr<[channel\\_status\\_msg\\_receiver](#)>

## Functions

- [channel\\_status\\_msg\\_receiver\\_sptr](#) **channel\_status\_msg\_receiver\_make** ()

### 11.607.1 Detailed Description

GNU Radio block that receives asynchronous channel messages from acquisition and tracking blocks.

#### Author

Javier Arribas, 2019. [jarribas\(at\)cttc.es](mailto:jarribas(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [channel\\_status\\_msg\\_receiver.h](#).

---

## 11.608 channel\_status\_msg\_receiver.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file channel_status_msg_receiver.h
00003  * \brief GNU Radio block that receives asynchronous channel messages from
00004  * acquisition and tracking blocks
00005  * \author Javier Arribas, 2019. jarribas(at)cttc.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_CHANNEL_STATUS_MSG_RECEIVER_CC_H
00019 #define GNSS_SDR_CHANNEL_STATUS_MSG_RECEIVER_CC_H
00020

```

```

00021 #include "gnss_block_interface.h"
00022 #include "gnss_synchro.h"
00023 #include "monitor_pvt.h"
00024 #include <gnuradio/block.h>
00025 #include <pmt/pmt.h>
00026 #include <map>
00027 #include <memory>
00028
00029 /** \addtogroup Core
00030  * \{ */
00031 /** \addtogroup Core_Receiver_Library
00032  * \{ */
00033
00034
00035 class channel_status_msg_receiver;
00036
00037 using channel_status_msg_receiver_sptr = gnss_shared_ptr<channel_status_msg_receiver>;
00038
00039 channel_status_msg_receiver_sptr channel_status_msg_receiver_make();
00040
00041 /*!
00042  * \brief GNU Radio block that receives asynchronous channel messages from tlm blocks
00043  */
00044 class channel_status_msg_receiver : public gr::block
00045 {
00046 public:
00047     ~channel_status_msg_receiver() = default; //!< Default destructor
00048
00049     /*!
00050     * \brief return the current status map of all channels with valid telemetry
00051     */
00052     std::map<int, std::shared_ptr<Gnss_Synchro> get_current_status_map();
00053
00054     /*!
00055     * \brief return the current receiver PVT
00056     */
00057     Monitor_Pvt get_current_status_pvt();
00058
00059 private:
00060     friend channel_status_msg_receiver_sptr channel_status_msg_receiver_make();
00061     channel_status_msg_receiver();
00062     void msg_handler_channel_status(const pmt::pmt_t& msg);
00063     Monitor_Pvt d_pvt_status{};
00064     std::map<int, std::shared_ptr<Gnss_Synchro> d_channel_status_map;
00065 };
00066
00067
00068 /** \} */
00069 /** \} */
00070 #endif // GNSS_SDR_CHANNEL_STATUS_MSG_RECEIVER_CC_H

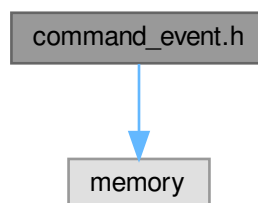
```

## 11.609 command\_event.h File Reference

Class that defines a receiver command event.

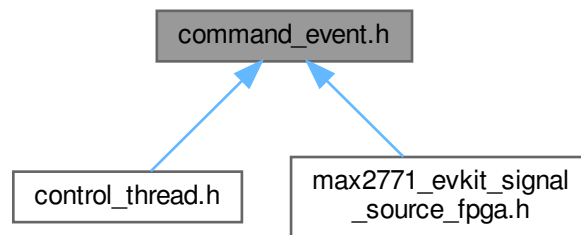
```
#include <memory>
```

Include dependency graph for command\_event.h:





This graph shows which files directly or indirectly include this file:



## Classes

- class [Command\\_Event](#)

## Typedefs

- using [command\\_event\\_sptr](#) = std::shared\_ptr<[Command\\_Event](#)>

## Functions

- [command\\_event\\_sptr](#) **command\_event\_make** (int command\_id, int event\_type)

### 11.609.1 Detailed Description

Class that defines a receiver command event.

#### Author

Javier Arribas, 2019. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [command\\_event.h](#).

---

## 11.610 command\_event.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file command_event.h
00003  * \brief Class that defines a receiver command event
00004  * \author Javier Arribas, 2019. jarribas(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_COMMAND_EVENT_H
00018 #define GNSS_SDR_COMMAND_EVENT_H
00019
00020 #include <memory>
00021
00022 /** \addtogroup Core
  
```

```

00023  * \{ */
00024  /** \addtogroup Core_Receiver_Library
00025  * \{ */
00026
00027
00028  class Command_Event;
00029
00030  using command_event_sptr = std::shared_ptr<Command_Event>;
00031
00032  command_event_sptr command_event_make(int command_id, int event_type);
00033
00034  class Command_Event
00035  {
00036  public:
00037      int command_id;
00038      int event_type;
00039  private:
00040      friend command_event_sptr command_event_make(int command_id, int event_type);
00041      Command_Event(int command_id_, int event_type_);
00042  };
00043
00044
00045
00046  /** \} */
00047  /** \} */
00048  #endif // GNSS_SDR_COMMAND_EVENT_H

```

## 11.611 galileo\_e6\_has\_msg\_receiver.h File Reference

GNU Radio block that processes Galileo HAS message pages received from Galileo E6B telemetry blocks. After successful decoding, sends the content to the PVT block.

```

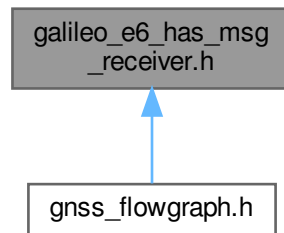
#include "Galileo_CNAV.h"
#include "galileo_has_data.h"
#include "gnss_block_interface.h"
#include "nav_message_packet.h"
#include <gnuradio/block.h>
#include <pmt/pmt.h>
#include <bitset>
#include <cstdint>
#include <map>
#include <memory>
#include <string>
#include <utility>
#include <vector>

```

Include dependency graph for galileo\_e6\_has\_msg\_receiver.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [galileo\\_e6\\_has\\_msg\\_receiver](#)

*GNU Radio block that receives asynchronous Galileo HAS message pages from the telemetry blocks, stores them in memory, and decodes HAS messages when enough data have been received. The decoded HAS message is sent to the PVT block.*

## Typedefs

- using [galileo\\_e6\\_has\\_msg\\_receiver\\_sptr](#) = `gnss_shared_ptr<galileo\_e6\_has\_msg\_receiver>`

## Functions

- `galileo_e6_has_msg_receiver_sptr galileo\_e6\_has\_msg\_receiver\_make ()`

### 11.611.1 Detailed Description

GNU Radio block that processes Galileo HAS message pages received from Galileo E6B telemetry blocks. After successful decoding, sends the content to the PVT block.

#### Author

Carles Fernandez-Prades, 2021. [cfernandez\(at\)cttc.es](mailto:cfernandez@cttc.es)

Javier Arribas, 2021. [jarribas\(at\)cttc.es](mailto:jarribas@cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [galileo\\_e6\\_has\\_msg\\_receiver.h](#).

---

## 11.612 galileo\_e6\_has\_msg\_receiver.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file galileo_e6_has_msg_receiver.h
00003  * \brief GNU Radio block that processes Galileo HAS message pages received from
00004  * Galileo E6B telemetry blocks. After successful decoding, sends the content to
00005  * the PVT block.
00006  * \author Carles Fernandez-Prades, 2021. cfernandez(at)cttc.es
00007  * \author Javier Arribas, 2021. jarribas(at)cttc.es
00008  *
00009  * -----
00010  *
00011  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00012  * This file is part of GNSS-SDR.
00013  *
00014  * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)
  
```

```

00015  * SPDX-License-Identifier: GPL-3.0-or-later
00016  *
00017  * -----
00018  */
00019
00020 #ifndef GNSS_SDR_GALILEO_E6_HAS_MSG_RECEIVER_H
00021 #define GNSS_SDR_GALILEO_E6_HAS_MSG_RECEIVER_H
00022
00023 #include "Galileo_CNAV.h"           // for GALILEO_CNAV_* constants
00024 #include "galileo_has_data.h"       // for Galileo_HAS_data
00025 #include "gnss_block_interface.h"   // for gnss_shared_ptr
00026 #include "nav_message_packet.h"     // for Nav_Message_Packet
00027 #include <gnuradio/block.h>         // for gr::block
00028 #include <pmt/pmt.h>                // for pmt::pmt_t
00029 #include <bitset>
00030 #include <cstdint>
00031 #include <map>
00032 #include <memory> // for std::unique_ptr
00033 #include <string>
00034 #include <utility> // std::pair
00035 #include <vector>
00036
00037 /** \addtogroup Core
00038  * \{ */
00039 /** \addtogroup Core_Receiver_Library
00040  * \{ */
00041
00042 class Galileo_HAS_page;
00043 class ReedSolomon;
00044 class galileo_e6_has_msg_receiver;
00045
00046 using galileo_e6_has_msg_receiver_sptr = gnss_shared_ptr<galileo_e6_has_msg_receiver>;
00047
00048 galileo_e6_has_msg_receiver_sptr galileo_e6_has_msg_receiver_make();
00049
00050 /*!
00051  * \brief GNU Radio block that receives asynchronous Galileo HAS message pages
00052  * from the telemetry blocks, stores them in memory, and decodes HAS messages
00053  * when enough data have been received.
00054  * The decoded HAS message is sent to the PVT block.
00055  */
00056 class galileo_e6_has_msg_receiver : public gr::block
00057 {
00058 public:
00059     ~galileo_e6_has_msg_receiver() = default; //!< Default destructor
00060     void set_enable_navdata_monitor(bool enable);
00061     std::shared_ptr<Galileo_HAS_data> process_test_page(const pmt::pmt_t& msg); //!< For testing
00062     purposes only
00063 private:
00064     friend galileo_e6_has_msg_receiver_sptr galileo_e6_has_msg_receiver_make();
00065     galileo_e6_has_msg_receiver();
00066
00067     void msg_handler_galileo_e6_has(const pmt::pmt_t& msg);
00068     void process_HAS_page(const Galileo_HAS_page& has_page);
00069     void read_MT1_header(const std::string& message_header);
00070     void read_MT1_body(const std::string& message_body);
00071     void delete_outdated_data(const Galileo_HAS_page& has_page);
00072
00073     int decode_message_type1(uint8_t message_id, uint8_t message_size);
00074
00075     uint16_t read_has_message_header_parameter_uint16(const std::bitset<GALILEO_CNAV_MT1_HEADER_BITS>&
00076 bits, const std::pair<int32_t, int32_t>& parameter) const;
00077     uint8_t read_has_message_header_parameter_uint8(const std::bitset<GALILEO_CNAV_MT1_HEADER_BITS>&
00078 bits, const std::pair<int32_t, int32_t>& parameter) const;
00079     bool read_has_message_header_parameter_bool(const std::bitset<GALILEO_CNAV_MT1_HEADER_BITS>& bits,
00080 const std::pair<int32_t, int32_t>& parameter) const;
00081
00082     uint64_t read_has_message_body_uint64(const std::string& bits) const;
00083     uint16_t read_has_message_body_uint16(const std::string& bits) const;
00084     int16_t read_has_message_body_int16(const std::string& bits) const;
00085     uint8_t read_has_message_body_uint8(const std::string& bits) const;
00086
00087     template <class T>
00088     std::string debug_print_vector(const std::string& title, const std::vector<T>& vec) const; //
00089     only for debug purposes
00090
00091     template <class T>
00092     std::string debug_print_matrix(const std::string& title, const std::vector<std::vector<T>& mat)
00093     const; // only for debug purposes
00094
00095     std::unique_ptr<ReedSolomon> d_rs;
00096     Galileo_HAS_data d_HAS_data{};
00097     Nav_Message_Packet d_nav_msg_packet;
00098
00099     // Store decoding matrices and received PIDs
00100     std::vector<std::vector<uint64_t> d_received_timestamps;

```

```

00096     std::vector<std::vector<std::vector<uint8_t>>> d_C_matrix;
00097     std::vector<std::vector<uint8_t> d_M_matrix;
00098     std::vector<std::vector<uint8_t> d_received_pids;
00099     std::vector<uint64_t> d_printed_timestamps;
00100     std::vector<bool> d_printed_mids;
00101
00102     // Store masks
00103     std::vector<int> d_nsat_in_mask_id;
00104     std::vector<std::vector<uint8_t> d_gnss_id_in_mask;
00105     std::vector<std::vector<uint64_t> d_satellite_mask;
00106     std::vector<std::vector<uint16_t> d_signal_mask;
00107     std::vector<std::vector<bool> d_cell_mask_availability_flag;
00108     std::vector<std::vector<std::vector<std::vector<bool>>> d_cell_mask;
00109     std::vector<uint8_t> d_nsys_in_mask;
00110     std::vector<std::vector<uint8_t> d_nav_message_mask;
00111
00112     std::map<std::pair<uint8_t, uint8_t>, std::vector<uint16_t> d_iod_ref_map;
00113
00114     uint8_t d_current_has_status{};
00115     uint8_t d_current_message_id{};
00116     bool d_new_message{};
00117     bool d_enable_navdata_monitor{};
00118 };
00119
00120
00121 /** \} */
00122 /** \} */
00123 #endif // GNSS_SDR_GALILEO_E6_HAS_MSG_RECEIVER_H

```

## 11.613 galileo\_tow\_map.h File Reference

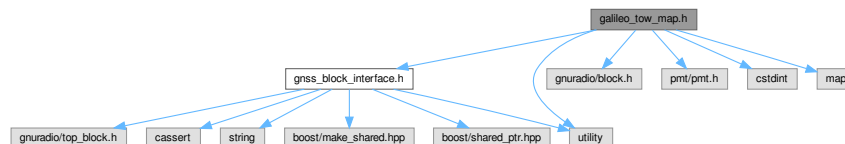
GNU Radio block that stores TOW for Galileo channels.

```

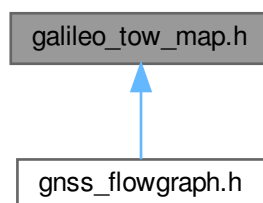
#include "gnss_block_interface.h"
#include <gnuradio/block.h>
#include <pmt/pmt.h>
#include <cstdint>
#include <map>
#include <utility>

```

Include dependency graph for galileo\_tow\_map.h:



This graph shows which files directly or indirectly include this file:



**Classes**

- class [galileo\\_tow\\_map](#)

**Typedefs**

- using [galileo\\_tow\\_map\\_sptr](#) = gnss\_shared\_ptr<[galileo\\_tow\\_map](#)>

**Functions**

- [galileo\\_tow\\_map\\_sptr](#) [galileo\\_tow\\_map\\_make](#) ()

**11.613.1 Detailed Description**

GNU Radio block that stores TOW for Galileo channels.

**Author**

Carles Fernandez-Prades, 2022. [cfernandez\(at\)cttc.es](mailto:cfernandez(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [galileo\\_tow\\_map.h](#).

---

**11.614 galileo\_tow\_map.h**

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file galileo_tow_map.h
00003  * \brief GNU Radio block that stores TOW for Galileo channels
00004  * \author Carles Fernandez-Prades, 2022. cfernandez(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_GALILEO_TOW_MAP_H
00018 #define GNSS_SDR_GALILEO_TOW_MAP_H
00019
00020 #include "gnss_block_interface.h" // for gnss_shared_ptr
00021 #include <gnuradio/block.h>       // for gr::block
00022 #include <pmt/pmt.h>               // for pmt::pmt_t
00023 #include <cstdint>
00024 #include <map>
00025 #include <utility>
00026
00027 /** \addtogroup Core
00028  * \{ */
00029 /** \addtogroup Core_Receiver_Library
00030  * \{ */
00031
00032 class galileo_tow_map;
00033
00034 using galileo_tow_map_sptr = gnss_shared_ptr<galileo_tow_map>;
00035
00036 galileo_tow_map_sptr galileo_tow_map_make();
00037
00038 class galileo_tow_map : public gr::block
00039 {
00040 public:
00041     ~galileo_tow_map() = default; //!< Default destructor
00042
00043 private:
00044     friend galileo_tow_map_sptr galileo_tow_map_make();
00045     galileo_tow_map();
00046
00047     void msg_handler_galileo_tow_map(const pmt::pmt_t& msg);
00048
00049     std::map<uint32_t, std::pair<uint32_t, uint64_t>> d_galileo_tow;

```

```

00050 };
00051
00052 /** \} */
00053 /** \} */
00054 #endif // GNSS_SDR_GALILEO_TOW_MAP_H

```

## 11.615 gnss\_crypto.h File Reference

Class for computing cryptographic functions.

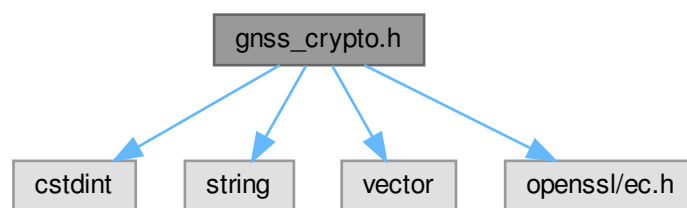
```
#include <stdint>
```

```
#include <string>
```

```
#include <vector>
```

```
#include <openssl/ec.h>
```

Include dependency graph for gnss\_crypto.h:



### Classes

- class [Gnss\\_Crypto](#)

*Class implementing cryptographic functions for Navigation Message Authentication.*

### 11.615.1 Detailed Description

Class for computing cryptographic functions.

#### Author

Carles Fernandez, 2023-2024. [cfernandez@cttc.es](mailto:cfernandez@cttc.es) Cesare Ghionoiu Martinez, 2023-2024. [c.ghionoiu-martinez@tu-braunschweig.de](mailto:c.ghionoiu-martinez@tu-braunschweig.de) [C ↩](#)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [gnss\\_crypto.h](#).

---

## 11.616 gnss\_crypto.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file gnss_crypto.h
00003  * \brief Class for computing cryptographic functions
00004  * \author Carles Fernandez, 2023-2024. cfernandez@cttc.es
00005  *         Cesare Ghionoiu Martinez, 2023-2024. c.ghionoiu-martinez@tu-braunschweig.de
00006  *
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors)

```

```

00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019 #ifndef GNSS_SDR_GNSS_CRYPT_H
00020 #define GNSS_SDR_GNSS_CRYPT_H
00021
00022 #include <stdint>
00023 #include <string>
00024 #include <vector>
00025 #if USE_GNUTLS_FALLBACK
00026 #include <gnutls/abstract.h>
00027 #include <gnutls/gnutls.h>
00028 #else // OpenSSL
00029 #include <openssl/ec.h>
00030 #endif
00031
00032 /** \addtogroup Core
00033  * \{
00034  /** \addtogroup Core_Receiver_Library
00035  * \{
00036
00037  /*!
00038  * \brief Class implementing cryptographic functions
00039  * for Navigation Message Authentication
00040  */
00041 class Gnss_Crypto
00042 {
00043 public:
00044     Gnss_Crypto(); //!< Default constructor
00045
00046     /*!
00047     * Constructor with a .crt or .pem file for the ECDSA Public Key
00048     * and a XML file for the Merkle Tree root.
00049     * Files can be downloaded by registering at https://www.gsc-europa.eu/
00050     */
00051     Gnss_Crypto(const std::string& certFilePath, const std::string& merkleTreePath);
00052     ~Gnss_Crypto(); //!< Default destructor
00053
00054     bool have_public_key() const; //!< Returns true if the ECDSA Public Key is already loaded
00055
00056     /*!
00057     * Stores the ECDSA Public Key in a .pem file, which is read in a following run if the .crt file
00058     * is not found
00059     */
00059     bool store_public_key(const std::string& pubKeyFilePath) const;
00060
00061     bool verify_signature_ecdsa_p256(const std::vector<uint8_t>& message, const std::vector<uint8_t>&
signature) const; //!< Verify ECDSA-P256 signature (message in plain hex, signature in raw format)
00062     bool verify_signature_ecdsa_p521(const std::vector<uint8_t>& message, const std::vector<uint8_t>&
signature) const; //!< Verify ECDSA-P521 signature (message in plain hex, signature in raw format)
00063
00064     std::vector<uint8_t> compute_SHA_256(const std::vector<uint8_t>& input) const;
00065     /*!< Computes SHA-256 hash
00065     std::vector<uint8_t> compute_SHA3_256(const std::vector<uint8_t>& input) const;
00066     /*!< Computes SHA3-256 hash
00066     std::vector<uint8_t> compute_HMAC_SHA_256(const std::vector<uint8_t>& key, const
std::vector<uint8_t>& input) const; /*!< Computes HMAC-SHA-256 message authentication code
00067     std::vector<uint8_t> compute_CMAC_AES(const std::vector<uint8_t>& key, const std::vector<uint8_t>&
input) const; /*!< Computes CMAC-AES message authentication code
00068
00069     std::vector<uint8_t> get_merkle_root() const; /*!< Gets the Merkle Tree root node (\f$ x_{4,0}
\f$)
00070     std::string get_public_key_type() const; /*!< Gets the ECDSA Public Key type (ECDSA P-256 /
ECDSA P-521 / Unknown)
00071
00072     void set_public_key(const std::vector<uint8_t>& publickey); /*!< Sets the ECDSA Public Key
(publickey compressed format)
00073     void set_public_key_type(const std::string& public_key_type); /*!< Sets the ECDSA Public Key type
(ECDSA P-256 / ECDSA P-521)
00074     void set_merkle_root(const std::vector<uint8_t>& v); /*!< Sets the Merkle Tree root node
x(\f$ x_{4,0} \f$)
00075     void read_merkle_xml(const std::string& merkleFilePath); /*!> Reads the XML file provided
from the GSC OSNMA server
00076
00077 private:
00078     void readPublicKeyFromPEM(const std::string& pemFilePath);
00079     bool readPublicKeyFromCRT(const std::string& crtFilePath);
00080     bool convert_raw_to_der_ecdsa(const std::vector<uint8_t>& raw_signature, std::vector<uint8_t>&
der_signature) const;
00081     std::vector<uint8_t> convert_from_hex_str(const std::string& input) const; // TODO - deprecate if
OSNMA helper is to do this operation
00082 #if USE_GNUTLS_FALLBACK
00083     void decompress_public_key_secp256r1(const std::vector<uint8_t>& compressed_key,
std::vector<uint8_t>& x, std::vector<uint8_t>& y) const;
00084     void decompress_public_key_secp521r1(const std::vector<uint8_t>& compressed_key,

```



```

        std::vector<uint8_t>& x, std::vector<uint8_t>& y) const;
00085     bool pubkey_copy(gnutls_pubkey_t src, gnutls_pubkey_t* dest);
00086     gnutls_pubkey_t d_PublicKey{};
00087 #else // OpenSSL
00088 #if USE_OPENSSL_3
00089     bool pubkey_copy(EVP_PKEY* src, EVP_PKEY** dest);
00090     EVP_PKEY* d_PublicKey{};
00091 #else // OpenSSL 1.x
00092     bool pubkey_copy(EC_KEY* src, EC_KEY** dest);
00093     EC_KEY* d_PublicKey = nullptr;
00094 #endif
00095 #endif
00096     std::vector<uint8_t> d_x_4_0;
00097     std::string d_PublicKeyType;
00098 };
00099
00100 /** @} */
00101 /** @} */
00102
00103 #endif // GNSS_SDR_GNSS_CRYPTO_H

```

## 11.617 gnss\_sdr\_fpga\_sample\_counter.h File Reference

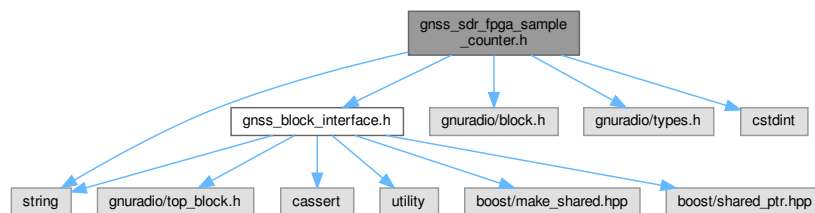
Simple block to report the current receiver time based on the output of the tracking or telemetry blocks.

```

#include "gnss_block_interface.h"
#include <gnuradio/block.h>
#include <gnuradio/types.h>
#include <cstdint>
#include <string>

```

Include dependency graph for gnss\_sdr\_fpga\_sample\_counter.h:



### Classes

- class [gnss\\_sdr\\_fpga\\_sample\\_counter](#)

### Typedefs

- using [gnss\\_sdr\\_fpga\\_sample\\_counter\\_sptr](#) = [gnss\\_shared\\_ptr](#)<[gnss\\_sdr\\_fpga\\_sample\\_counter](#)>

### Functions

- [gnss\\_sdr\\_fpga\\_sample\\_counter\\_sptr](#) **gnss\_sdr\_make\_fpga\_sample\_counter** (double \_fs, int32\_t \_↔ interval\_ms)

### 11.617.1 Detailed Description

Simple block to report the current receiver time based on the output of the tracking or telemetry blocks.

#### Author

Javier Arribas 2018. [jarribas\(at\)cttc.es](mailto:jarribas@cttc.es)

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [gnss\\_sdr\\_fpga\\_sample\\_counter.h](#).

## 11.618 gnss\_sdr\_fpga\_sample\_counter.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file gnss_sdr_fpga_sample_counter.h
00003  * \brief Simple block to report the current receiver time based on the output
00004  * of the tracking or telemetry blocks
00005  * \author Javier Arribas 2018. jarribas(at)cttc.es
00006  *
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019 #ifndef GNSS_SDR_GNSS_SDR_FPGA_SAMPLE_COUNTER_H
00020 #define GNSS_SDR_GNSS_SDR_FPGA_SAMPLE_COUNTER_H
00021
00022 #include "gnss_block_interface.h"
00023 #include <gnuradio/block.h>
00024 #include <gnuradio/types.h> // for gr_vector_const_void_star
00025 #include <stdint.h>
00026 #include <string>
00027
00028 /** \addtogroup Core
00029  * \{ */
00030 /** \addtogroup Core_Receiver_Library
00031  * \{ */
00032
00033
00034 class gnss_sdr_fpga_sample_counter;
00035
00036 using gnss_sdr_fpga_sample_counter_sptr = gnss_shared_ptr<gnss_sdr_fpga_sample_counter>;
00037
00038 gnss_sdr_fpga_sample_counter_sptr gnss_sdr_make_fpga_sample_counter(double _fs, int32_t _interval_ms);
00039
00040 class gnss_sdr_fpga_sample_counter : public gr::block
00041 {
00042 public:
00043     ~gnss_sdr_fpga_sample_counter();
00044     int general_work(int noutput_items,
00045                     gr_vector_int &ninput_items,
00046                     gr_vector_const_void_star &input_items,
00047                     gr_vector_void_star &output_items);
00048 private:
00049     const std::string device_name = "counter"; // UIO device name
00050
00051     static const uint32_t FPGA_PAGE_SIZE = 0x1000; // default page size for the
00052     multicorrelator memory map
00053     static const uint32_t test_reg_sanity_check = 0x55AA; // value to check the presence of the test
00054     register (to detect the hw)
00055     friend gnss_sdr_fpga_sample_counter_sptr gnss_sdr_make_fpga_sample_counter(double _fs, int32_t
00056     _interval_ms);
00057     gnss_sdr_fpga_sample_counter(double _fs, int32_t _interval_ms);
00058     uint32_t test_register(uint32_t writeval);
00059     void configure_samples_per_output(uint32_t interval);
00060     void close_device(void);
00061     void open_device(void);
00062     bool start();
00063     bool stop();
00064     void wait_for_interrupt(void) const;
00065
00066     volatile uint32_t *map_base; // driver memory map
00067
00068     double fs;
00069     uint64_t sample_counter;
00070     uint64_t last_sample_counter;
00071     uint64_t current_T_rx_ms; // Receiver time in ms since the beginning of the run
00072
00073     uint32_t samples_per_output;
00074     uint32_t samples_per_report;
00075     uint32_t interval_ms;
00076     uint32_t current_s; // Receiver time in seconds, modulo 60
00077     uint32_t current_m; // Receiver time in minutes, modulo 60
00078     uint32_t current_h; // Receiver time in hours, modulo 24
00079     uint32_t current_days; // Receiver time in days since the beginning of the run
00080     int32_t report_interval_ms;
00081     int32_t fd; // driver descriptor

```

```

00081
00082     bool flag_enable_send_msg;
00083     bool flag_m;           // True if the receiver has been running for at least 1 minute
00084     bool flag_h;           // True if the receiver has been running for at least 1 hour
00085     bool flag_days;        // True if the receiver has been running for at least 1 day
00086     bool is_open;
00087 };
00088
00089
00090 /** \} */
00091 /** \} */
00092 #endif // GNSS_SDR_GNSS_SDR_FPGA_SAMPLE_COUNTER_H

```

## 11.619 gnss\_sdr\_sample\_counter.h File Reference

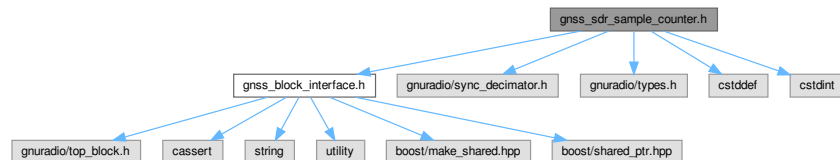
Simple block to report the current receiver time based on the output of the tracking or telemetry blocks.

```

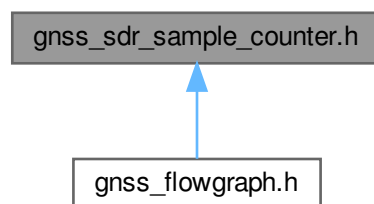
#include "gnss_block_interface.h"
#include <gnuradio/sync_decimator.h>
#include <gnuradio/types.h>
#include <cstdio>
#include <stdint>

```

Include dependency graph for gnss\_sdr\_sample\_counter.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [gnss\\_sdr\\_sample\\_counter](#)

### Typedefs

- using [gnss\\_sdr\\_sample\\_counter\\_sptr](#) = [gnss\\_shared\\_ptr](#)<[gnss\\_sdr\\_sample\\_counter](#)>

### Functions

- [gnss\\_sdr\\_sample\\_counter\\_sptr](#) **gnss\_sdr\_make\_sample\_counter** (double fs, int32\_t interval\_ms, size\_t size)

### 11.619.1 Detailed Description

Simple block to report the current receiver time based on the output of the tracking or telemetry blocks.

#### Author

Javier Arribas 2018. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gnss\\_sdr\\_sample\\_counter.h](#).

---

### 11.620 gnss\_sdr\_sample\_counter.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file gnss_sdr_sample_counter.h
00003  * \brief Simple block to report the current receiver time based on the output of the tracking or
00004  * \author Javier Arribas 2018. jarribas(at)cttc.es
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_GNSS_SDR_SAMPLE_COUNTER_H
00019 #define GNSS_SDR_GNSS_SDR_SAMPLE_COUNTER_H
00020
00021 #include "gnss_block_interface.h"
00022 #include <gnuradio/sync_decimator.h>
00023 #include <gnuradio/types.h> // for gr_vector_const_void_star
00024 #include <cstdlib> // for size_t
00025 #include <cstdint>
00026
00027 /** \addtogroup Core
00028  * \{ */
00029 /** \addtogroup Core_Receiver_Library
00030  * Utilities for the core GNSS receiver.
00031  * \{ */
00032
00033
00034 class gnss_sdr_sample_counter;
00035
00036 using gnss_sdr_sample_counter_sptr = gnss_shared_ptr<gnss_sdr_sample_counter>;
00037
00038 gnss_sdr_sample_counter_sptr gnss_sdr_make_sample_counter(
00039     double _fs,
00040     int32_t _interval_ms,
00041     size_t _size);
00042
00043 class gnss_sdr_sample_counter : public gr::sync_decimator
00044 {
00045 public:
00046     ~gnss_sdr_sample_counter() = default;
00047     int work(int noutput_items,
00048             gr_vector_const_void_star &input_items,
00049             gr_vector_void_star &output_items);
00050
00051 private:
00052     friend gnss_sdr_sample_counter_sptr gnss_sdr_make_sample_counter(
00053         double _fs,
00054         int32_t _interval_ms,
00055         size_t _size);
00056
00057     gnss_sdr_sample_counter(double _fs,
00058                             int32_t _interval_ms,
00059                             size_t _size);
00060
00061     int64_t uint64diff(uint64_t first, uint64_t second);
00062
00063     double fs;
00064     int64_t current_T_rx_ms; // Receiver time in ms since the beginning of the run
00065     uint64_t sample_counter;
00066     int32_t interval_ms;

```

```

00067     int32_t report_interval_ms;
00068     uint32_t samples_per_output;
00069     uint32_t current_s;      // Receiver time in seconds, modulo 60
00070     uint32_t current_m;     // Receiver time in minutes, modulo 60
00071     uint32_t current_h;     // Receiver time in hours, modulo 24
00072     uint32_t current_days;  // Receiver time in days since the beginning of the run
00073     bool flag_m;            // True if the receiver has been running for at least 1 minute
00074     bool flag_h;            // True if the receiver has been running for at least 1 hour
00075     bool flag_days;        // True if the receiver has been running for at least 1 day
00076     bool flag_enable_send_msg;
00077 };
00078
00079
00080 /** \} */
00081 /** \} */
00082 #endif // GNSS_SDR_GNSS_SDR_SAMPLE_COUNTER_H

```

## 11.621 gnss\_sdr\_supl\_client.h File Reference

class that implements a C++ interface to external Secure User Location Protocol (SUPL) client library.

```

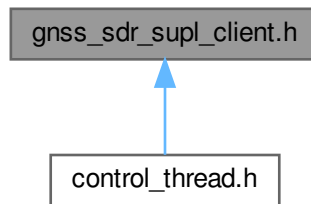
#include "supl.h"
#include "agnss_ref_location.h"
#include "agnss_ref_time.h"
#include "galileo_almanac.h"
#include "galileo_ephemeris.h"
#include "galileo_iono.h"
#include "galileo_utc_model.h"
#include "glonass_gnav_ephemeris.h"
#include "glonass_gnav_utc_model.h"
#include "gps_acq_assist.h"
#include "gps_almanac.h"
#include "gps_cnav_ephemeris.h"
#include "gps_cnav_utc_model.h"
#include "gps_ephemeris.h"
#include "gps_iono.h"
#include "gps_utc_model.h"
#include <fstream>
#include <map>
#include <string>

```

Include dependency graph for gnss\_sdr\_supl\_client.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Gnss\\_Sdr\\_Supl\\_Client](#)

*class that implements a C++ interface to external Secure User Location Protocol (SUPL) client library..*

### 11.621.1 Detailed Description

class that implements a C++ interface to external Secure User Location Protocol (SUPL) client library.

#### Author

Javier Arribas, 2013. jarribas(at)cttc.es

TODO: put here supl.c author info class that implements a C++ interface to external Secure User Location Protocol (SUPL) client library.

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [gnss\\_sdr\\_supl\\_client.h](#).

### 11.622 gnss\_sdr\_supl\_client.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file gnss_sdr_supl_client.h
00003  * \brief class that implements a C++ interface to external Secure User Location Protocol (SUPL)
00004  * \author Javier Arribas, 2013. jarribas(at)cttc.es
00005  *
00006  * TODO: put here supl.c author info
00007  * class that implements a C++ interface to external Secure User Location Protocol (SUPL) client
00008  * library.
00009  * -----
00010  *
00011  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00012  * This file is part of GNSS-SDR.
00013  *
00014  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00015  * SPDX-License-Identifier: GPL-3.0-or-later
00016  * -----
00017  */
00018
00019
00020 #ifndef GNSS_SDR_SUPL_CLIENT_H
00021 #define GNSS_SDR_SUPL_CLIENT_H
00022
00023 extern "C"
00024 {
00025 #include "supl.h"
00026 }
00027 #include "agnss_ref_location.h"
00028 #include "agnss_ref_time.h"
  
```

```

00029 #include "galileo_almanac.h"
00030 #include "galileo_ephemeris.h"
00031 #include "galileo_iono.h"
00032 #include "galileo_utc_model.h"
00033 #include "glonass_gnav_ephemeris.h"
00034 #include "glonass_gnav_utc_model.h"
00035 #include "gps_acq_assist.h"
00036 #include "gps_almanac.h"
00037 #include "gps_cnav_ephemeris.h"
00038 #include "gps_cnav_utc_model.h"
00039 #include "gps_ephemeris.h"
00040 #include "gps_iono.h"
00041 #include "gps_utc_model.h"
00042 #include <fstream>
00043 #include <map>
00044 #include <string>
00045
00046 /** \addtogroup Core
00047  * \{ */
00048 /** \addtogroup Core_Receiver_Library
00049  * \{ */
00050
00051
00052 /*!
00053  * \brief class that implements a C++ interface to external Secure User Location Protocol (SUPL)
00054  * client library..
00055 */
00055 class Gnss_Sdr_Supl_Client
00056 {
00057 public:
00058     Gnss_Sdr_Supl_Client();
00059     ~Gnss_Sdr_Supl_Client() = default;
00060
00061     // SUPL SERVER INFO
00062     std::string server_name;
00063     int server_port;
00064     int request;
00065     // ephemeris map
00066     std::map<int, Gps_Ephemeris> gps_ephemeris_map;
00067     std::map<int, Galileo_Ephemeris> gal_ephemeris_map;
00068     std::map<int, Gps_CNAV_Ephemeris> gps_cnav_ephemeris_map;
00069     std::map<int, Glonass_Gnav_Ephemeris> glonass_gnav_ephemeris_map;
00070
00071     // almanac map
00072     std::map<int, Gps_Almanac> gps_almanac_map;
00073     std::map<int, Galileo_Almanac> gal_almanac_map;
00074
00075     // ionospheric model
00076     Gps_Iono gps_iono;
00077     Galileo_Iono gal_iono;
00078     // reference time
00079     Agnss_Ref_Time gps_time;
00080     // UTC model
00081     Gps_Utc_Model gps_utc;
00082     Galileo_Utc_Model gal_utc;
00083     Gps_CNAV_Utc_Model gps_cnav_utc;
00084     Glonass_Gnav_Utc_Model glo_gnav_utc;
00085     // reference location
00086     Agnss_Ref_Location gps_ref_loc;
00087     // Acquisition Assistance map
00088     std::map<int, Gps_Acq_Assist> gps_acq_map;
00089
00090     /*
00091     * \brief Initiates the TCP SSL SUPL connection to the SUPL server and request assistance data
00092     using the provided GSM Base station parameters
00093     * \param i_mcc Current network MCC (Mobile country code), 3 digits.
00094     * \param i_mns Current network MNC (Mobile Network code), 2 or 3 digits.
00095     * \param i_lac Current network LAC (Location area code), 16 bits, 1-65520 are valid values.
00096     * \param i_ci Cell Identity (16 bits, 0-65535 are valid values).
00097     * \return Error code -> 0 no errors.
00098     */
00098     int get_assistance(int i_mcc, int i_mns, int i_lac, int i_ci);
00099
00100     /*
00101     * \brief Read the received SUPL data and stores it into the corresponding class members
00102     (gps_ephemeris_map, gps_almanac_map, gps_iono, gps_time, gps_utc, gps_acq_map, and gps_ref_loc)
00103     */
00104     void read_supl_data();
00105
00106     /*!
00107     * \brief Read GPS NAV ephemeris map from XML file
00108     */
00109     bool load_ephemeris_xml(const std::string& file_name);
00110
00111     /*!
00112     * \brief Save ephemeris map to XML file.

```

```
00113     */
00114     bool save_ephemeris_map_xml(const std::string& file_name,
00115                                std::map<int, Gps_Ephemeris> eph_map);
00116
00117     /*!
00118     * \brief Read GPS CNAV ephemeris map from XML file
00119     */
00120     bool load_cnav_ephemeris_xml(const std::string& file_name);
00121
00122     /*!
00123     * \brief Save GPS CNAV ephemeris map to XML file.
00124     */
00125     bool save_cnav_ephemeris_map_xml(const std::string& file_name,
00126                                      std::map<int, Gps_CNAV_Ephemeris> eph_map);
00127
00128     /*!
00129     * \brief Read Galileo ephemeris map from XML file
00130     */
00131     bool load_gal_ephemeris_xml(const std::string& file_name);
00132
00133     /*!
00134     * \brief Save Galileo ephemeris map to XML file.
00135     */
00136     bool save_gal_ephemeris_map_xml(const std::string& file_name,
00137                                     std::map<int, Galileo_Ephemeris> eph_map);
00138
00139     /*!
00140     * \brief Read GLONASS GNAV ephemeris map from XML file
00141     */
00142     bool load_gnav_ephemeris_xml(const std::string& file_name);
00143
00144     /*!
00145     * \brief Save GLONASS GNAV ephemeris map to XML file.
00146     */
00147     bool save_gnav_ephemeris_map_xml(const std::string& file_name,
00148                                      std::map<int, Glonass_Gnav_Ephemeris> eph_map);
00149
00150     /*!
00151     * \brief Read GPS utc model from XML file
00152     */
00153     bool load_utc_xml(const std::string& file_name);
00154
00155     /*!
00156     * \brief Save UTC model map to XML file
00157     */
00158     bool save_utc_xml(const std::string& file_name, Gps_Utc_Model& utc);
00159
00160     /*!
00161     * \brief Read CNAV GPS utc model from XML file
00162     */
00163     bool load_cnav_utc_xml(const std::string& file_name);
00164
00165     /*!
00166     * \brief Save CNAV UTC model map to XML file
00167     */
00168     bool save_cnav_utc_xml(const std::string& file_name, Gps_CNAV_Utc_Model& utc);
00169
00170     /*!
00171     * \brief Read Galileo utc model from XML file
00172     */
00173     bool load_gal_utc_xml(const std::string& file_name);
00174
00175     /*!
00176     * \brief Save Galileo UTC model map to XML file
00177     */
00178     bool save_gal_utc_xml(const std::string& file_name, Galileo_Utc_Model& utc);
00179
00180     /*!
00181     * \brief Read Galileo almanac map from XML file
00182     */
00183     bool load_gal_almanac_xml(const std::string& file_name);
00184
00185     /*!
00186     * \brief Save Galileo almanac map to XML file
00187     */
00188     bool save_gal_almanac_xml(const std::string& file_name, std::map<int, Galileo_Almanac>
00189                               galileo_almanac_map_to_save);
00189
00190     /*!
00191     * \brief Read GPS almanac map from XML file
00192     */
00193     bool load_gps_almanac_xml(const std::string& file_name);
00194
00195     /*!
00196     * \brief Save GPS almanac map to XML file
00197     */
00198     bool save_gps_almanac_xml(const std::string& file_name, std::map<int, Gps_Almanac>
```



```

        gps_almanac_map_to_save);
00199
00200     /*!
00201     * \brief Read iono from XML file
00202     */
00203     bool load_iono_xml(const std::string& file_name);
00204
00205     /*!
00206     * \brief Save iono map to XML file
00207     */
00208     bool save_iono_xml(const std::string& file_name, Gps_Iono& iono);
00209
00210     /*!
00211     * \brief Read Galileo iono from XML file
00212     */
00213     bool load_gal_iono_xml(const std::string& file_name);
00214
00215     /*!
00216     * \brief Save Galileo iono map to XML file
00217     */
00218     bool save_gal_iono_xml(const std::string& file_name, Galileo_Iono& iono);
00219
00220     /*!
00221     * \brief Read Glonass utc model from XML file
00222     */
00223     bool load_glo_utc_xml(const std::string& file_name);
00224
00225     /*!
00226     * \brief Save Glonass UTC model map to XML file
00227     */
00228     bool save_glo_utc_xml(const std::string& file_name, Glonass_Gnav_Utc_Model& utc);
00229
00230     /*!
00231     * \brief Read ref time from XML file
00232     */
00233     bool load_ref_time_xml(const std::string& file_name);
00234
00235     /*!
00236     * \brief Save ref time map to XML file
00237     */
00238     bool save_ref_time_xml(const std::string& file_name,
00239                          Agnss_Ref_Time& ref_time_map);
00240
00241     /*!
00242     * \brief Read ref location from XML file
00243     */
00244     bool load_ref_location_xml(const std::string& file_name);
00245
00246     /*!
00247     * \brief Save ref location map to XML file
00248     */
00249     bool save_ref_location_xml(const std::string& file_name,
00250                          Agnss_Ref_Location& ref_location);
00251
00252     /*
00253     * Prints SUPL data to std::cout. Use it for debug purposes only.
00254     */
00255     void print_assistance();
00256
00257 private:
00258     bool read_gal_almanac_from_gsa(const std::string& file_name);
00259     // assistance protocol structure
00260     supl_ctx_t ctx{};
00261     // assistance data
00262     supl_assist_t assist{};
00263     // GSM CELL INFO
00264     int mcc;
00265     int mns;
00266     int lac;
00267     int ci;
00268 };
00269
00270
00271 /** \} */
00272 /** \} */
00273 #endif // GNSS_SDR_SUPL_CLIENT_H

```

## 11.623 gnss\_sdr\_time\_counter.h File Reference

Simple block to report the current receiver time based on the output of the tracking or telemetry blocks.

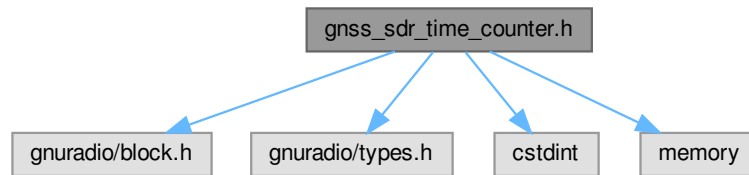
```

#include <gnuradio/block.h>
#include <gnuradio/types.h>
#include <cstdint>

```

```
#include <memory>
```

Include dependency graph for `gnss_sdr_time_counter.h`:



## Classes

- class [gnss\\_sdr\\_time\\_counter](#)

## Typedefs

- using [gnss\\_sdr\\_time\\_counter\\_sptr](#) = `std::shared_ptr<gnss\_sdr\_time\_counter>`

## Functions

- `gnss_sdr_time_counter_sptr gnss_sdr_make_time_counter ()`

### 11.623.1 Detailed Description

Simple block to report the current receiver time based on the output of the tracking or telemetry blocks.

#### Author

Antonio Ramos 2018. [antonio.ramosdet\(at\)gmail.com](mailto:antonio.ramosdet(at)gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gnss\\_sdr\\_time\\_counter.h](#).

---

## 11.624 gnss\_sdr\_time\_counter.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002   * \file gnss_sdr_time_counter.h
00003   * \brief Simple block to report the current receiver time based on the output of the tracking or
00004   *        telemetry blocks
00005   * \author Antonio Ramos 2018. antonio.ramosdet\(at\)gmail.com
00006   *
00007   * -----
00008   *
00009   * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010   * This file is part of GNSS-SDR.
00011   *
00012   * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013   * SPDX-License-Identifier: GPL-3.0-or-later
00014   *
00015   * -----
00016   */
00017
00018 #ifndef GNSS_SDR_GNSS_SDR_TIME_COUNTER_H
00019 #define GNSS_SDR_GNSS_SDR_TIME_COUNTER_H
00020
00021 #include <gnuradio/block.h>
00022 #include <gnuradio/types.h> // for gr_vector_const_void_star
00023 #include <cstdint>
00024 #include <memory>
  
```

```

00025
00026 /** \addtogroup Core
00027 * \{ */
00028 /** \addtogroup Core_Receiver_Library
00029 * \{ */
00030
00031
00032 class gnss_sdr_time_counter;
00033
00034 using gnss_sdr_time_counter_sptr = std::shared_ptr<gnss_sdr_time_counter>;
00035
00036 gnss_sdr_time_counter_sptr gnss_sdr_make_time_counter();
00037
00038 class gnss_sdr_time_counter : public gr::block
00039 {
00040 public:
00041     ~gnss_sdr_time_counter() = default;
00042     int general_work(int noutput_items __attribute__((unused)),
00043         gr_vector_int &ninput_items __attribute__((unused)),
00044         gr_vector_const_void_star &input_items __attribute__((unused)),
00045         gr_vector_void_star &output_items);
00046
00047 private:
00048     gnss_sdr_time_counter();
00049     friend gnss_sdr_time_counter_sptr gnss_sdr_make_time_counter();
00050
00051     int64_t current_T_rx_ms; // Receiver time in ms since the beginning of the run
00052     int32_t report_interval_ms;
00053     uint32_t current_s; // Receiver time in seconds, modulo 60
00054     uint32_t current_m; // Receiver time in minutes, modulo 60
00055     uint32_t current_h; // Receiver time in hours, modulo 24
00056     uint32_t current_days; // Receiver time in days since the beginning of the run
00057     bool flag_m; // True if the receiver has been running for at least 1 minute
00058     bool flag_h; // True if the receiver has been running for at least 1 hour
00059     bool flag_days; // True if the receiver has been running for at least 1 day
00060 };
00061
00062
00063 /** \} */
00064 /** \} */
00065 #endif // GNSS_SDR_GNSS_SDR_SAMPLE_COUNTER_H

```

## 11.625 ini.h File Reference

This function parses an INI file into easy-to-access name/value pairs.

### Macros

- `#define INI_ALLOW_MULTILINE 1`

### Functions

- `int ini_parse (const char *filename, int(*handler)(void *user, const char *section, const char *name, const char *value), void *user)`

*Parse given INI-style file. May have [section]s, name=value pairs (whitespace stripped), and comments starting with ';' (semicolon). Section is "" if name=value pair parsed before any section heading.*

### 11.625.1 Detailed Description

This function parses an INI file into easy-to-access name/value pairs.

#### Author

Brush Technologies, 2009.

inih (INI Not Invented Here) is a simple .INI file parser written in C++. It's only a couple of pages of code, and it was designed to be small and simple, so it's good for embedded systems. To use it, just give `ini_parse()` an INI file, and it will call a callback for every name=value pair parsed, giving you strings for the section, name, and value. It's done this way because it works well on low-memory embedded systems, but also because it makes for a KISS implementation. Parse given INI-style file. May have [section]s, name=value pairs (whitespace stripped), and comments starting with ';' (semicolon). Section is "" if name=value pair parsed before any section heading. For each name=value pair parsed, call handler function with given user pointer as well as section, name, and value

(data only valid for duration of handler call). Handler should return nonzero on success, zero on error. Returns 0 on success, line number of first error on parse error, on -1 on file open error  
 inih and INIReaden are released under the New BSD license:

Copyright (c) 2009, Brush Technology All rights reserved.

SPDX-License-Identifier: BSD-3-Clause

Go to the project home page for more info:

**11.625.1.1** <https://github.com/benhoyt/inih>

Definition in file [ini.h](#).

## 11.626 inih.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file inih.h
00003  * \brief This function parses an INI file into easy-to-access name/value pairs.
00004  * \author Brush Technologies, 2009.
00005  *
00006  * inih (INI Not Invented Here) is a simple .INI file parser written in C++.
00007  * It's only a couple of pages of code, and it was designed to be small
00008  * and simple, so it's good for embedded systems. To use it, just give
00009  * ini_parse() an INI file, and it will call a callback for every
00010  * name=value pair parsed, giving you strings for the section, name,
00011  * and value. It's done this way because it works well on low-memory
00012  * embedded systems, but also because it makes for a KISS implementation.
00013  * Parse given INI-style file. May have [section]s, name=value pairs
00014  * (whitespace stripped), and comments starting with ';' (semicolon).
00015  * Section is "" if name=value pair parsed before any section heading.
00016  * For each name=value pair parsed, call handler function with given user
00017  * pointer as well as section, name, and value (data only valid for duration
00018  * of handler call). Handler should return nonzero on success, zero on error.
00019  * Returns 0 on success, line number of first error on parse error, on -1 on
00020  * file open error
00021  *
00022  * -----
00023  * inih and INIReaden are released under the New BSD license:
00024  *
00025  * Copyright (c) 2009, Brush Technology
00026  * All rights reserved.
00027  *
00028  * SPDX-License-Identifier: BSD-3-Clause
00029  *
00030  * Go to the project home page for more info:
00031  *
00032  * https://github.com/benhoyt/inih
00033  * -----
00034  */
00035
00036 #ifndef GNSS_SDR_INI_H
00037 #define GNSS_SDR_INI_H
00038
00039 /** \addtogroup Core
00040  * \{ */
00041 /** \addtogroup Core_Receiver_Library
00042  * \{ */
00043
00044
00045 /*! \brief Parse given INI-style file. May have [section]s, name=value pairs
00046  * (whitespace stripped), and comments starting with ';' (semicolon). Section
00047  * is "" if name=value pair parsed before any section heading.
00048
00049  * For each name=value pair parsed, call handler function with given user
00050  * pointer as well as section, name, and value (data only valid for duration
00051  * of handler call). Handler should return nonzero on success, zero on error.
00052
00053  * Returns 0 on success, line number of first error on parse error, on -1 on
00054  * file open error.
00055  */
00056 int ini_parse(const char* filename,
00057               int (*handler)(void* user, const char* section,
00058                               const char* name, const char* value),
00059               void* user);
00060
00061 /* Nonzero to allow multi-line value parsing, in the style of Python's
00062  * ConfigParser. If allowed, ini_parse() will call the handler with the same
00063  * name for each subsequent line parsed. */
00064 #ifndef INI_ALLOW_MULTILINE
00065 #define INI_ALLOW_MULTILINE 1
00066 #endif
```

```
00067
00068
00069 /** \} */
00070 /** \} */
00071 #endif // GNSS_SDR_INI_H
```

## 11.627 INIReader.h File Reference

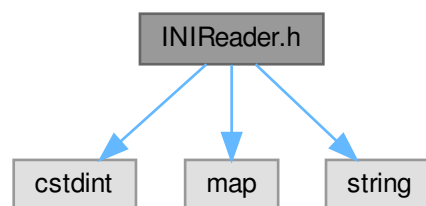
This class reads an INI file into easy-to-access name/value pairs.

```
#include <cstdint>
```

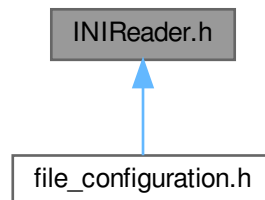
```
#include <map>
```

```
#include <string>
```

Include dependency graph for INIReader.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [INIReader](#)

*Read an INI file into easy-to-access name/value pairs. (Note that I've gone for simplicity here rather than speed, but it should be pretty decent.)*

### 11.627.1 Detailed Description

This class reads an INI file into easy-to-access name/value pairs.

**Author**

Brush Technologies, 2009.

inih (INI Not Invented Here) is a simple .INI file parser written in C++. It's only a couple of pages of code, and it was designed to be small and simple, so it's good for embedded systems. To use it, just give `ini_parse()` an INI file, and it will call a callback for every name=value pair parsed, giving you strings for the section, name, and value. It's done this way because it works well on low-memory embedded systems, but also because it makes for a KISS implementation.

inih and `INIReader` are released under the New BSD license:

Copyright (c) 2009, Brush Technology All rights reserved.

SPDX-License-Identifier: BSD-3-Clause

Go to the project home page for more info:

**11.627.1.1**    <https://github.com/benhoyt/inih>

Definition in file `INIReader.h`.

## 11.628 INIReader.h

Go to the documentation of this file.

```
00001 /*!
00002  * \file INIReader.h
00003  * \brief This class reads an INI file into easy-to-access name/value pairs.
00004  * \author Brush Technologies, 2009.
00005  *
00006  * inih (INI Not Invented Here) is a simple .INI file parser written in C++.
00007  * It's only a couple of pages of code, and it was designed to be small
00008  * and simple, so it's good for embedded systems. To use it, just give
00009  * ini_parse() an INI file, and it will call a callback for every
00010  * name=value pair parsed, giving you strings for the section, name,
00011  * and value. It's done this way because it works well on low-memory
00012  * embedded systems, but also because it makes for a KISS implementation.
00013  *
00014  * -----
00015  * inih and INIReader are released under the New BSD license:
00016  *
00017  * Copyright (c) 2009, Brush Technology
00018  * All rights reserved.
00019  *
00020  * SPDX-License-Identifier: BSD-3-Clause
00021  *
00022  * Go to the project home page for more info:
00023  *
00024  * https://github.com/benhoyt/inih
00025  * -----
00026  */
00027
00028 #ifndef GNSS_SDR_INIREADER_H
00029 #define GNSS_SDR_INIREADER_H
00030
00031 #include <stdint>
00032 #include <map>
00033 #include <string>
00034
00035 /** \addtogroup Core
00036  * \{ */
00037 /** \addtogroup Core_Receiver_Library
00038  * \{ */
00039
00040
00041 /*!
00042  * \brief Read an INI file into easy-to-access name/value pairs. (Note that I've gone
00043  * for simplicity here rather than speed, but it should be pretty decent.)
00044  */
00045 class INIReader
00046 {
00047 public:
00048     /// Construct INIReader and parse given filename. See ini.h for more info about the parsing.
00049     explicit INIReader(const std::string& filename);
00050
00051     /// Return the result of ini_parse(), i.e., 0 on success, line number of first error on parse
    error, or -1 on file open error.
00052     int ParseError() const;
00053
00054     /// Get a string value from INI file, returning default_value if not found.
00055     std::string Get(const std::string& section, const std::string& name,
00056                     const std::string& default_value);
```

```

00057
00058     //!< Get an integer (long) value from INI file, returning default_value if not found.
00059     int64_t GetInteger(const std::string& section, const std::string& name, int64_t default_value);
00060
00061     //!< Return true if the given section exists (section must contain at least one name=value pair).
00062     bool HasSection(const std::string& section) const;
00063
00064     //!< Return true if a value exists with the given section and field names.
00065     bool HasValue(const std::string& section, const std::string& name) const;
00066
00067 private:
00068     static std::string MakeKey(const std::string& section, const std::string& name);
00069     static int ValueHandler(void* user, const char* section, const char* name,
00070                             const char* value);
00071
00072     std::map<std::string, std::string> _values;
00073     int _error;
00074 };
00075
00076
00077 /** \} */
00078 /** \} */
00079 #endif // GNSS_SDR_INIREADER_H

```

## 11.629 nav\_message\_monitor.h File Reference

GNU Radio block that processes [Nav\\_Message\\_Packet](#) received from telemetry blocks and sends them via UDP.

```

#include "gnss_block_interface.h"
#include "nav_message_udp_sink.h"
#include <gnuradio/block.h>
#include <pmt/pmt.h>
#include <cstdint>
#include <memory>
#include <string>
#include <vector>

```

Include dependency graph for nav\_message\_monitor.h:



### Classes

- class [nav\\_message\\_monitor](#)

GNU Radio block that receives asynchronous [Nav\\_Message\\_Packet](#) objects from the telemetry blocks and sends them via UDP.

### Typedefs

- using [nav\\_message\\_monitor\\_sptr](#) = [gnss\\_shared\\_ptr](#)<[nav\\_message\\_monitor](#)>

### Functions

- [nav\\_message\\_monitor\\_sptr nav\\_message\\_monitor\\_make](#) (const std::vector< std::string > &addresses, uint16\_t port)

#### 11.629.1 Detailed Description

GNU Radio block that processes [Nav\\_Message\\_Packet](#) received from telemetry blocks and sends them via UDP.

## Author

Carles Fernandez-Prades, 2021. cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [nav\\_message\\_monitor.h](#).

---

## 11.630 nav\_message\_monitor.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file nav_message_monitor.h
00003  * \brief GNU Radio block that processes Nav_Message_Packet received from
00004  * telemetry blocks and sends them via UDP.
00005  * \author Carles Fernandez-Prades, 2021. cfernandez(at)cttc.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_NAV_MESSAGE_MONITOR_H
00019 #define GNSS_SDR_NAV_MESSAGE_MONITOR_H
00020
00021 #include "gnss_block_interface.h"
00022 #include "nav_message_udp_sink.h"
00023 #include <gnuradio/block.h>
00024 #include <pmt/pmt.h>
00025 #include <stdint>
00026 #include <memory>
00027 #include <string>
00028 #include <vector>
00029
00030 /** \addtogroup Core
00031  * \{ */
00032 /** \addtogroup Core_Receiver_Library
00033  * \{ */
00034
00035 class nav_message_monitor;
00036
00037 using nav_message_monitor_sptr = gnss_shared_ptr<nav_message_monitor>;
00038
00039 nav_message_monitor_sptr nav_message_monitor_make(const std::vector<std::string>& addresses, uint16_t
    port);
00040
00041 /*!
00042  * \brief GNU Radio block that receives asynchronous Nav_Message_Packet objects
00043  * from the telemetry blocks and sends them via UDP
00044  */
00045 class nav_message_monitor : public gr::block
00046 {
00047 public:
00048     ~nav_message_monitor() = default; //!< Default destructor
00049
00050 private:
00051     friend nav_message_monitor_sptr nav_message_monitor_make(const std::vector<std::string>&
        addresses, uint16_t port);
00052     nav_message_monitor(const std::vector<std::string>& addresses, uint16_t port);
00053     void msg_handler_nav_message(const pmt::pmt_t& msg);
00054     std::unique_ptr<Nav_Message_Udp_Sink> nav_message_udp_sink_;
00055 };
00056
00057
00058 /** \} */
00059 /** \} */
00060 #endif // GNSS_SDR_NAV_MESSAGE_MONITOR_H

```

## 11.631 nav\_message\_packet.h File Reference

Class for storage of decoded navigation messages.

```

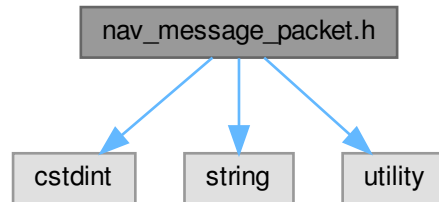
#include <stdint>
#include <string>

```

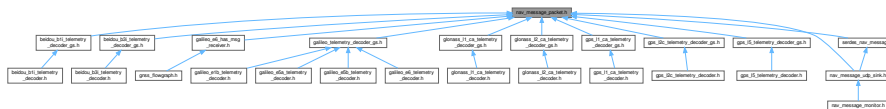


```
#include <utility>
```

Include dependency graph for nav\_message\_packet.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class Nav\_Message\_Packet

### 11.631.1 Detailed Description

Class for storage of decoded navigation messages.

**Author**

Carles Fernandez-Prades, 2021. cfernandez(at)cttc.es

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [nav\\_message\\_packet.h](#).

## 11.632 nav\_message\_packet.h

[Go to the documentation of this file.](#)

```
00001 /*!  
00002  * \file nav_message_packet.h  
00003  * \brief Class for storage of decoded navigation messages  
00004  * \author Carles Fernandez-Prades, 2021. cfernandez(at)cttc.es  
00005  *  
00006  * -----  
00007  *  
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.  
00009  * This file is part of GNSS-SDR.  
00010  *  
00011  * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)  
00012  * SPDX-License-Identifier: GPL-3.0-or-later  
00013  *  
00014  * -----  
00015  */  
00016  
00017 #ifndef GNSS_SDR_NAV_MESSAGE_PACKET_H  
00018 #define GNSS_SDR_NAV_MESSAGE_PACKET_H  
00019  
00020 #include <stdint>  
00021 #include <string>
```

```

00022 #include <utility>
00023
00024 /** \addtogroup Core
00025  * \{ */
00026 /** \addtogroup Core_Receiver_Library
00027  * \{ */
00028
00029 class Nav_Message_Packet
00030 {
00031 public:
00032     Nav_Message_Packet() = default; //!< Default constructor
00033
00034     ~Nav_Message_Packet() = default; //!< Default destructor
00035
00036     std::string system;           //!< GNSS constellation: "G" for GPS, "R" for Glonass, "S" for
SBAS, "E" for Galileo and "C" for Beidou.
00037     std::string signal;           //!< GNSS signal: "1C" for GPS L1 C/A, "1B" for Galileo E1b/c,
"1G" for Glonass L1 C/A, "2S" for GPS L2 L2C(M), "2G" for Glonass L2 C/A, "L5" for GPS L5 and "5X" for
Galileo E5a
00038     int32_t prn;                 //!< SV ID
00039     int32_t tow_at_current_symbol_ms; //!< Time of week of the current symbol, in ms
00040     std::string nav_message;      //!< Content of the navigation page
00041
00042     /// Copy constructor
00043     Nav_Message_Packet(const Nav_Message_Packet& other) noexcept
00044     {
00045         *this = other;
00046     };
00047
00048     /// Copy assignment operator
00049     Nav_Message_Packet& operator=(const Nav_Message_Packet& rhs) noexcept
00050     {
00051         // Only do assignment if RHS is a different object from this.
00052         if (this != &rhs)
00053         {
00054             this->system = rhs.system;
00055             this->signal = rhs.signal;
00056             this->prn = rhs.prn;
00057             this->tow_at_current_symbol_ms = rhs.tow_at_current_symbol_ms;
00058             this->nav_message = rhs.nav_message;
00059         }
00060         return *this;
00061     };
00062
00063     /// Move constructor
00064     Nav_Message_Packet(Nav_Message_Packet&& other) noexcept
00065     {
00066         *this = std::move(other);
00067     };
00068
00069     /// Move assignment operator
00070     Nav_Message_Packet& operator=(Nav_Message_Packet&& other) noexcept
00071     {
00072         if (this != &other)
00073         {
00074             this->system = other.system;
00075             this->signal = other.signal;
00076             this->prn = other.prn;
00077             this->tow_at_current_symbol_ms = other.tow_at_current_symbol_ms;
00078             this->nav_message = other.nav_message;
00079         }
00080         return *this;
00081     };
00082 };
00083
00084 /** \} */
00085 /** \} */
00086 #endif // GNSS_SDR_NAV_MESSAGE_PACKET_H

```

## 11.633 nav\_message\_udp\_sink.h File Reference

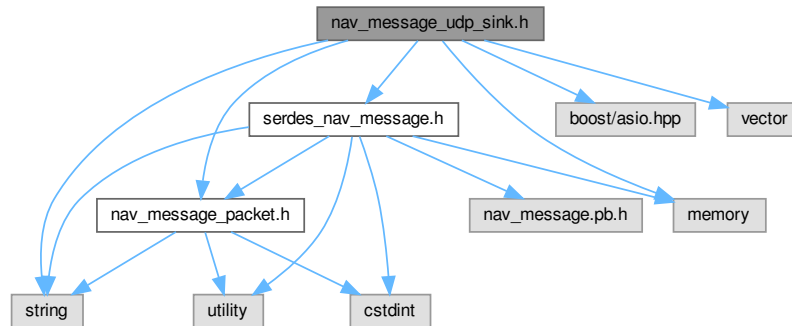
Interface of a class that sends serialized [Nav\\_Message\\_Packet](#) objects over UDP to one or multiple endpoints.

```

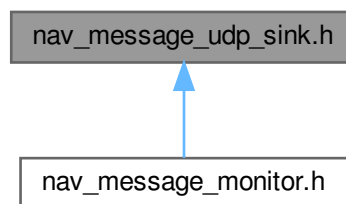
#include "nav_message_packet.h"
#include "serdes_nav_message.h"
#include <boost/asio.hpp>
#include <memory>
#include <string>
#include <vector>

```

Include dependency graph for nav\_message\_udp\_sink.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Nav\\_Message\\_Udp\\_Sink](#)

## Typedefs

- using [b\\_io\\_context](#) = boost::asio::io\_service

### 11.633.1 Detailed Description

Interface of a class that sends serialized [Nav\\_Message\\_Packet](#) objects over UDP to one or multiple endpoints.

#### Author

Carles Fernandez-Prades, 2021. cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [nav\\_message\\_udp\\_sink.h](#).

---

## 11.634 nav\_message\_udp\_sink.h

[Go to the documentation of this file.](#)

00001 /\*!

```

00002  * \file nav_message_udp_sink.h
00003  * \brief Interface of a class that sends serialized Nav_Message_Packet objects
00004  * over UDP to one or multiple endpoints.
00005  * \author Carles Fernandez-Prades, 2021. cfernandez(at)cttc.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_NAV_MESSAGE_UDP_SINK_H
00019 #define GNSS_SDR_NAV_MESSAGE_UDP_SINK_H
00020
00021 #include "nav_message_packet.h"
00022 #include "serdes_nav_message.h"
00023 #include <boost/asio.hpp>
00024 #include <memory>
00025 #include <string>
00026 #include <vector>
00027
00028 /** \addtogroup Core
00029  * \{ */
00030 /** \addtogroup Core_Receiver_Library
00031  * \{ */
00032
00033 #if USE_BOOST_ASIO_IO_CONTEXT
00034 using b_io_context = boost::asio::io_context;
00035 #else
00036 using b_io_context = boost::asio::io_service;
00037 #endif
00038
00039 class Nav_Message_Udp_Sink
00040 {
00041 public:
00042     Nav_Message_Udp_Sink(const std::vector<std::string>& addresses, const uint16_t& port);
00043     bool write_nav_message(const std::shared_ptr<Nav_Message_Packet>& nav_meg_packet);
00044
00045 private:
00046     Serdes_Nav_Message serdes_nav;
00047     b_io_context io_context;
00048     boost::asio::ip::udp::socket socket;
00049     std::vector<boost::asio::ip::udp::endpoint> endpoints;
00050     boost::system::error_code error;
00051 };
00052
00053
00054 /** \} */
00055 /** \} */
00056 #endif // GNSS_SDR_NAV_MESSAGE_UDP_SINK_H

```

## 11.635 osnma\_helper.h File Reference

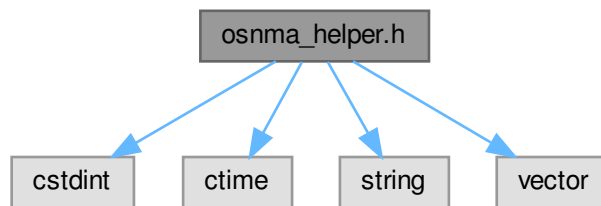
Class for auxiliary osnma functions.

```

#include <cstdint>
#include <ctime>
#include <string>
#include <vector>

```

Include dependency graph for osnma\_helper.h:



## Classes

- class [Osnma\\_Helper](#)

### 11.635.1 Detailed Description

Class for auxiliary osnma functions.

#### Author

Carles Fernandez-Prades, 2024 cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2023 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [osnma\\_helper.h](#).

---

## 11.636 osnma\_helper.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file osnma_helper.h
00003  * \brief Class for auxiliary osnma functions
00004  * \author Carles Fernandez-Prades, 2024 cfernandez(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2023 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_OSNMA_HELPER_H
00018 #define GNSS_SDR_OSNMA_HELPER_H
00019
00020
00021 #include <cstdint>
00022 #include <ctime>
00023 #include <string>
00024 #include <vector>
00025
00026 /** \addtogroup Core
00027  * \{ */
00028 /** \addtogroup Core_Receiver_Library
00029  * \{ */
00030
00031 class Osnma_Helper
00032 {
00033 public:
00034     Osnma_Helper();
00035     ~Osnma_Helper() = default;
  
```

```

00036     uint32_t compute_gst(uint32_t WN, uint32_t TOW) const;
00037     uint32_t compute_gst(std::tm& input);
00038     uint32_t compute_gst_now();
00039     uint32_t get_WN(uint32_t GST) const;
00040     uint32_t get_TOW(uint32_t GST) const;
00041     std::vector<uint8_t> gst_to_uint8(uint32_t GST) const;
00042     std::vector<uint8_t> bytes(const std::string& binaryString) const;
00043     std::string verification_status_str(int status) const;
00044     std::string convert_to_hex_string(const std::vector<uint8_t>& vector) const;
00045     std::vector<uint8_t> convert_from_hex_string(const std::string& hex_string) const; // TODO remove
    similar function in gnss_crypto
00046     std::tm GST_START_EPOCH{};
00047 };
00048
00049 /** \} */
00050 /** \} */
00051 #endif // GNSS_SDR_OSNMA_HELPER_H

```

## 11.637 osnma\_msg\_receiver.h File Reference

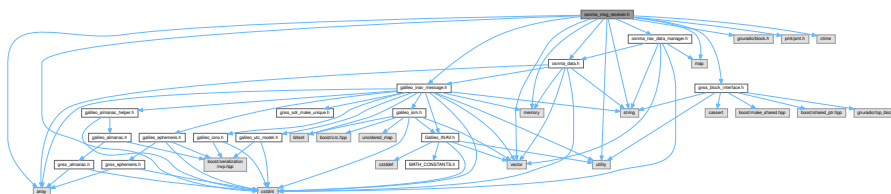
GNU Radio block that processes Galileo OSNMA data received from Galileo E1B telemetry blocks. After successful decoding, sends the content to the PVT block.

```

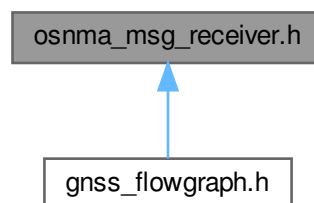
#include "galileo_inav_message.h"
#include "gnss_block_interface.h"
#include "osnma_data.h"
#include "osnma_nav_data_manager.h"
#include <gnuradio/block.h>
#include <pmt/pmt.h>
#include <array>
#include <stdint>
#include <ctime>
#include <map>
#include <memory>
#include <string>
#include <utility>
#include <vector>

```

Include dependency graph for osnma\_msg\_receiver.h:



This graph shows which files directly or indirectly include this file:



**Classes**

- class [osnma\\_msg\\_receiver](#)

*GNU Radio block that receives asynchronous OSNMA messages from the telemetry blocks, stores them in memory, and decodes OSNMA info when enough data have been received. The decoded OSNMA data is sent to the PVT block.*

**Macros**

- `#define FRIEND\_TEST(test_case_name, test_name)`

**Typedefs**

- using [osnma\\_msg\\_receiver\\_sptr](#) = `gnss_shared_ptr<osnma\_msg\_receiver>`

**Functions**

- `osnma_msg_receiver_sptr osnma\_msg\_receiver\_make (const std::string &pemFilePath, const std::string &merkleFilePath, bool strict_mode=false)`

**11.637.1 Detailed Description**

GNU Radio block that processes Galileo OSNMA data received from Galileo E1B telemetry blocks. After successful decoding, sends the content to the PVT block.

**Author**

Carles Fernandez-Prades, 2023-2024. [cfernandez\(at\)cttc.es](mailto:cfernandez(at)cttc.es) Cesare Ghionoiu Martinez, 2023-2024. [c.ghionoiu-martinez@tu-braunschweig.de](mailto:c.ghionoiu-martinez@tu-braunschweig.de)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [osnma\\_msg\\_receiver.h](#).

---

**11.637.2 Macro Definition Documentation****11.637.2.1 FRIEND\_TEST**

```
#define FRIEND_TEST(
    test_case_name,
    test_name)
```

**Value:**

```
friend class test_case_name##_Test
```

Definition at line 23 of file [osnma\\_msg\\_receiver.h](#).

**11.638 osnma\_msg\_receiver.h**

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file osnma_msg_receiver.h
00003  * \brief GNU Radio block that processes Galileo OSNMA data received from
00004  * Galileo E1B telemetry blocks. After successful decoding, sends the content to
00005  * the PVT block.
00006  * \author Carles Fernandez-Prades, 2023-2024. cfernandez(at)cttc.es
00007  * Cesare Ghionoiu Martinez, 2023-2024. c.ghionoiu-martinez@tu-braunschweig.de
00008  *
00009  * -----
00010  *
00011  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00012  * This file is part of GNSS-SDR.
00013  *
00014  * Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors)
00015  * SPDX-License-Identifier: GPL-3.0-or-later
00016  *
00017  * -----
00018  */
```

```

00019
00020 #ifndef GNSS_SDR_OSNMA_MSG_RECEIVER_H
00021 #define GNSS_SDR_OSNMA_MSG_RECEIVER_H
00022
00023 #define FRIEND_TEST(test_case_name, test_name) \
00024     friend class test_case_name##_Test
00025
00026 #include "galileo_inav_message.h" // for OSNMA_msg
00027 #include "gnss_block_interface.h" // for gnss_shared_ptr
00028 #include "osnma_data.h" // for OSNMA_data structures
00029 #include "osnma_nav_data_manager.h" // for OSNMA_NavDataManager
00030 #include <gnuradio/block.h> // for gr::block
00031 #include <pmt/pmt.h> // for pmt::pmt_t
00032 #include <array> // for std::array
00033 #include <cstdint> // for uint8_t
00034 #include <ctime> // for std::time_t
00035 #include <map> // for std::map, std::multimap
00036 #include <memory> // for std::shared_ptr
00037 #include <string> // for std::string
00038 #include <utility> // for std::pair
00039 #include <vector> // for std::vector
00040
00041 /** \addtogroup Core
00042 * \{ */
00043 /** \addtogroup Core_Receiver_Library
00044 * \{ */
00045
00046 class OSNMA_DSM_Reader;
00047 class Gnss_Crypto;
00048 class Osnma_Helper;
00049 class osnma_msg_receiver;
00050
00051 using osnma_msg_receiver_sptr = gnss_shared_ptr<osnma_msg_receiver>;
00052
00053 osnma_msg_receiver_sptr osnma_msg_receiver_make(const std::string& pemFilePath, const std::string&
merkleFilePath, bool strict_mode = false);
00054
00055 /*!
00056 * \brief GNU Radio block that receives asynchronous OSNMA messages
00057 * from the telemetry blocks, stores them in memory, and decodes OSNMA info
00058 * when enough data have been received.
00059 * The decoded OSNMA data is sent to the PVT block.
00060 */
00061 class osnma_msg_receiver : public gr::block
00062 {
00063 public:
00064     ~osnma_msg_receiver() = default; //!< Default destructor
00065     bool verify_dsm_pkr(const DSM_PKR_message& message) const; //!< Public for benchmarking purposes
00066     void msg_handler_osnma(const pmt::pmt_t& msg); //!< For testing purposes
00067     void read_merkle_xml(const std::string& merklepath); //!< Public for testing purposes
00068     void set_merkle_root(const std::vector<uint8_t>& v); //!< Public for benchmarking purposes
00069
00070 private:
00071     friend osnma_msg_receiver_sptr osnma_msg_receiver_make(const std::string& pemFilePath, const
std::string& merkleFilePath, bool strict_mode);
00072     osnma_msg_receiver(const std::string& crtFilePath, const std::string& merkleFilePath, bool
strict_mode);
00073
00074     void process_osnma_message(const std::shared_ptr<OSNMA_msg>& osnma_msg);
00075     void read_nma_header(uint8_t nma_header);
00076     void read_dsm_header(uint8_t dsm_header);
00077     void read_dsm_block(const std::shared_ptr<OSNMA_msg>& osnma_msg);
00078     void process_dsm_block(const std::shared_ptr<OSNMA_msg>& osnma_msg);
00079     void process_dsm_message(const std::vector<uint8_t>& dsm_msg, const uint8_t& nma_header);
00080     void read_and_process_mack_block(const std::shared_ptr<OSNMA_msg>& osnma_msg);
00081     void read_mack_header();
00082     void read_mack_body();
00083     void process_mack_message();
00084     void remove_verified_tags();
00085     void control_tags_awaiting_verify_size();
00086     void display_data();
00087     void send_data_to_pvt(const std::vector<OSNMA_NavData>& data);
00088
00089     bool verify_tesla_key(std::vector<uint8_t>& key, uint32_t TOW);
00090     bool verify_tag(Tag& tag) const;
00091     bool tag_has_nav_data_available(const Tag& t) const;
00092     bool tag_has_key_available(const Tag& t) const;
00093     bool verify_macseq(const MACK_message& mack);
00094
00095     bool store_dsm_kroot(const std::vector<uint8_t>& dsm, const uint8_t nma_header) const;
00096
00097     std::pair<std::vector<uint8_t>, uint8_t> parse_dsm_kroot() const;
00098     std::vector<uint8_t> get_merkle_tree_leaves(const DSM_PKR_message& dsm_pkr_message) const;
00099     std::vector<uint8_t> compute_merkle_root(const DSM_PKR_message& dsm_pkr_message, const
std::vector<uint8_t>& m_i) const;
00100     std::vector<uint8_t> build_message(Tag& tag) const;
00101     std::vector<uint8_t> hash_chain(uint32_t num_of_hashes_needed, const std::vector<uint8_t>& key,

```



```

uint32_t GST_SfI, const uint8_t lk_bytes) const;
00102     std::vector<MACK_tag_and_info> verify_macseq_new(const MACK_message& mack);
00103
00104     std::map<uint32_t, std::map<uint32_t, OSNMA_NavData> d_satellite_nav_data; // map holding
OSNMA_NavData sorted by SVID (first key) and TOW (second key).
00105     std::map<uint32_t, std::vector<uint8_t> d_tesla_keys; // tesla keys over
time, sorted by TOW
00106     std::multimap<uint32_t, Tag> d_tags_awaiting_verify; // container with
tags to verify from arbitrary SVIDs, sorted by TOW
00107
00108     std::vector<uint8_t> d_new_public_key;
00109     std::vector<uint8_t> d_tags_to_verify{0, 4, 12};
00110     std::vector<MACK_message> d_macks_awaiting_MACSEQ_verification;
00111
00112     std::array<std::array<uint8_t, 256>, 16> d_dsm_message{}; // structure for recording DSM blocks,
when filled it sends them to parse and resets itself.
00113     std::array<std::array<uint8_t, 16>, 16> d_dsm_id_received{};
00114     std::array<uint16_t, 16> d_number_of_blocks{};
00115     std::array<uint8_t, 60> d_mack_message{}; // C: 480 b
00116
00117     std::unique_ptr<Gnss_Crypto> d_crypto; // class for cryptographic functions
00118     std::unique_ptr<OSNMA_DSM_Reader> d_dsm_reader; // osnma parameters parser
00119     std::unique_ptr<Osnma_Helper> d_helper; // helper class with auxiliary
functions
00120     std::unique_ptr<OSNMA_NavDataManager> d_nav_data_manager; // refactor for holding and processing
navigation data
00121
00122     OSNMA_data d_osnma_data{};
00123
00124     uint32_t d_last_received_GST{0}; // latest GST received
00125     uint32_t d_GST_Sf{}; // Scaled GST time for cryptographic computations
00126     uint32_t d_GST_Rx{0}; // local GST receiver time
00127     uint32_t d_last_verified_key_GST{0}; // GST for the latest verified TESLA key
00128     uint32_t d_GST_0{}; // Time of applicability GST (KROOT + 30 s)
00129     uint32_t d_GST_SIS{}; // GST coming from W6 and W5 of SIS
00130     uint32_t d_GST_PKR_PKREV_start{};
00131     uint32_t d_GST_PKR_AM_start{};
00132     uint32_t d_GST_chain_renewal_start{};
00133     uint32_t d_GST_chain_revocation_start{};
00134
00135     uint32_t d_count_successful_tags{0};
00136     uint32_t d_count_failed_tags{0};
00137     uint32_t d_count_failed_Kroot{0};
00138     uint32_t d_count_failed_pubKey{0}; // failed public key verifications against Merkle root
00139     uint32_t d_count_failed_macseq{0};
00140
00141     uint8_t const d_T_L{30}; // s RG Section 2.1
00142     uint8_t d_new_public_key_id{};
00143
00144     bool d_new_data{false};
00145     bool d_public_key_verified{false};
00146     bool d_kroot_verified{false};
00147     bool d_tesla_key_verified{false};
00148     bool d_strict_mode{false};
00149     bool d_flag_hot_start{false};
00150     bool d_flag_PK_renewal{false};
00151     bool d_flag_PK_revocation{false};
00152     bool d_flag_NPK_set{false};
00153     bool d_flag_alert_message{false};
00154     bool d_flag_chain_renewal{false};
00155     bool d_flag_chain_revocation{false};
00156
00157     // Provide access to inner functions to Gtest
00158     FRIEND_TEST(OsnmaMsgReceiverTest, TeslaKeyVerification);
00159     FRIEND_TEST(OsnmaMsgReceiverTest, TagVerification);
00160     FRIEND_TEST(OsnmaMsgReceiverTest, BuildTagMessageM0);
00161     FRIEND_TEST(OsnmaMsgReceiverTest, VerifyPublicKey);
00162     FRIEND_TEST(OsnmaMsgReceiverTest, ComputeBaseLeaf);
00163     FRIEND_TEST(OsnmaMsgReceiverTest, ComputeMerkleRoot);
00164     FRIEND_TEST(OsnmaTestVectors, NominalTestConf1);
00165     FRIEND_TEST(OsnmaTestVectors, NominalTestConf2);
00166     FRIEND_TEST(OsnmaTestVectors, PublicKeyRenewal);
00167     FRIEND_TEST(OsnmaTestVectors, PublicKeyRevocation);
00168     FRIEND_TEST(OsnmaTestVectors, ChainRenewal);
00169     FRIEND_TEST(OsnmaTestVectors, ChainRevocation);
00170     FRIEND_TEST(OsnmaTestVectors, AlertMessage);
00171 };
00172
00173
00174 /** \} */
00175 /** \} */
00176 #endif // GNSS_SDR_OSNMA_MSG_RECEIVER_H

```



## 11.640 osnma\_nav\_data\_manager.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file osnma_nav_data_manager.h
00003  * \brief Class for Galileo OSNMA navigation data management
00004  * \author Cesare Ghioniu-Martinez, 2020-2023 cesare.martinez(at)proton.me
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2023 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_OSNMA_NAV_DATA_MANAGER_H
00018 #define GNSS_SDR_OSNMA_NAV_DATA_MANAGER_H
00019
00020 #include "osnma_data.h" // for OSNMA_NavData, Tag
00021 #include <stdint>        // for uint32_t
00022 #include <map>
00023 #include <string>
00024 #include <vector>
00025
00026 /** \addtogroup Core
00027  * \{ */
00028 /** \addtogroup Core_Receiver_Library
00029  * \{ */
00030
00031 /**
00032  * @class OSNMA_NavDataManager
00033  * @brief Class for managing OSNMA navigation data
00034  */
00035 class OSNMA_NavDataManager
00036 {
00037 public:
00038     OSNMA_NavDataManager() = default;
00039
00040     void log_status() const;
00041     bool have_nav_data(const Tag& t) const;
00042     bool have_nav_data(uint32_t PRNd, uint32_t TOW, uint8_t ADKD) const;
00043     std::string get_navigation_data(const Tag& t) const;
00044
00045     void add_navigation_data(const std::string& nav_bits, uint32_t PRNd, uint32_t TOW);
00046     void update_nav_data(const std::multimap<uint32_t, Tag>& tags_verified, uint8_t tag_size);
00047     bool have_nav_data(const std::string& nav_bits, uint32_t PRNd, uint32_t TOW);
00048     std::vector<OSNMA_NavData> get_verified_data();
00049
00050 private:
00051     std::map<uint32_t, std::map<uint32_t, OSNMA_NavData> > d_satellite_nav_data{}; // NavData sorted by
00052     [PRNd][TOW_start]
00053     const uint32_t L_t_min{40};
00054     const uint16_t EPH_SIZE{549};
00055     const uint16_t UTC_SIZE{141};
00056 };
00057 /** \} */
00058 /** \} */
00059 #endif // GNSS_SDR_OSNMA_NAV_DATA_MANAGER_H

```

## 11.641 serdes\_nav\_message.h File Reference

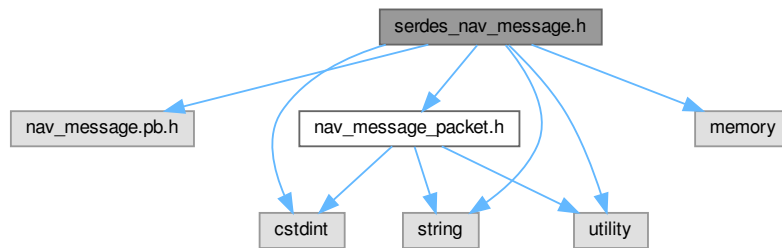
Serialization / Deserialization of [Nav\\_Message\\_Packet](#) objects using Protocol Buffers.

```

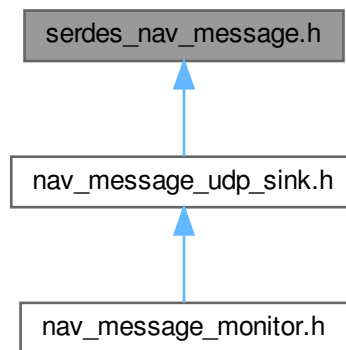
#include "nav_message.pb.h"
#include "nav_message_packet.h"
#include <stdint>
#include <memory>
#include <string>
#include <utility>

```

Include dependency graph for `serdes_nav_message.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Serdes\\_Nav\\_Message](#)

*This class implements serialization and deserialization of [Nav\\_Message\\_Packet](#) objects using Protocol Buffers.*

### 11.641.1 Detailed Description

Serialization / Deserialization of [Nav\\_Message\\_Packet](#) objects using Protocol Buffers.

#### Author

Carles Fernandez-Prades, 2021. cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [serdes\\_nav\\_message.h](#).

---

## 11.642 serdes\_nav\_message.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file serdes_nav_message.h
  
```

```

00003  * \brief Serialization / Deserialization of Nav_Message_Packet objects using
00004  * Protocol Buffers
00005  * \author Carles Fernandez-Prades, 2021. cfernandez(at)cttc.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_SERDES_NAV_MESSAGE_H
00019 #define GNSS_SDR_SERDES_NAV_MESSAGE_H
00020
00021 #include "nav_message.pb.h" // file created by Protocol Buffers at compile time
00022 #include "nav_message_packet.h"
00023 #include <stdint>
00024 #include <memory>
00025 #include <string>
00026 #include <utility>
00027
00028 /** \addtogroup Telemetry_Decoder
00029  * \{ */
00030 /** \addtogroup Telemetry_Decoder_libs
00031  * \{ */
00032
00033
00034 /*!
00035  * \brief This class implements serialization and deserialization of
00036  * Nav_Message_Packet objects using Protocol Buffers.
00037  */
00038 class Serdes_Nav_Message
00039 {
00040 public:
00041     Serdes_Nav_Message()
00042     {
00043         // Verify that the version of the library that we linked against is
00044         // compatible with the version of the headers we compiled against.
00045         GOOGLE_PROTOBUF_VERIFY_VERSION;
00046     }
00047
00048     ~Serdes_Nav_Message()
00049     {
00050         // google::protobuf::ShutdownProtobufLibrary();
00051     }
00052
00053     inline Serdes_Nav_Message(const Serdes_Nav_Message& other) noexcept : navmsg_(other.navmsg_) //!<
Copy constructor
00054     {
00055     }
00056
00057     inline Serdes_Nav_Message& operator=(const Serdes_Nav_Message& rhs) noexcept //!< Copy assignment
operator
00058     {
00059         if (this != &rhs)
00060         {
00061             this->navmsg_.CopyFrom(rhs.navmsg_);
00062         }
00063         return *this;
00064     }
00065
00066     inline Serdes_Nav_Message(Serdes_Nav_Message&& other) noexcept : navmsg_(std::move(other.navmsg_))
//!< Move constructor
00067     {
00068         // Set the other object's navmsg_ to a default-constructed state
00069         other.navmsg_ = gnss_sdr::navMsg{};
00070     }
00071
00072     inline Serdes_Nav_Message& operator=(Serdes_Nav_Message&& other) noexcept //!< Move assignment
operator
00073     {
00074         if (this != &other)
00075         {
00076             navmsg_ = std::move(other.navmsg_);
00077             other.navmsg_ = gnss_sdr::navMsg{};
00078         }
00079         return *this;
00080     }
00081
00082     inline std::string createProtobuffer(const std::shared_ptr<Nav_Message_Packet> nav_msg_packet)
//!< Serialization into a string
00083     {
00084         navmsg_.Clear();

```

```

00085         std::string data;
00086
00087         navmsg_.set_system(nav_msg_packet->system);
00088         navmsg_.set_signal(nav_msg_packet->signal);
00089         navmsg_.set_prn(nav_msg_packet->prn);
00090         navmsg_.set_tow_at_current_symbol_ms(nav_msg_packet->tow_at_current_symbol_ms);
00091         navmsg_.set_nav_message(nav_msg_packet->nav_message);
00092
00093         navmsg_.SerializeToString(&data);
00094
00095         return data;
00096     }
00097
00098     inline Nav_Message_Packet readProtobuffer(const gnss_sdr::navMsg& msg) const //!< Deserialization
00099     {
00100         Nav_Message_Packet navmsg;
00101
00102         navmsg.system = msg.system();
00103         navmsg.signal = msg.signal();
00104         navmsg.prn = msg.prn();
00105         navmsg.tow_at_current_symbol_ms = msg.tow_at_current_symbol_ms();
00106         navmsg.nav_message = msg.nav_message();
00107
00108         return navmsg;
00109     }
00110
00111 private:
00112     gnss_sdr::navMsg navmsg_{};
00113 };
00114
00115
00116 /** @} */
00117 /** @} */
00118 #endif // GNSS_SDR_SERDES_NAV_MESSAGE_H

```

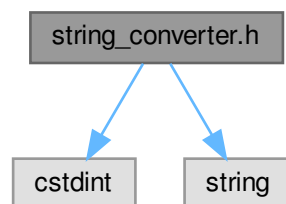
## 11.643 string\_converter.h File Reference

Interface of a class that interprets the contents of a string and converts it into different types.

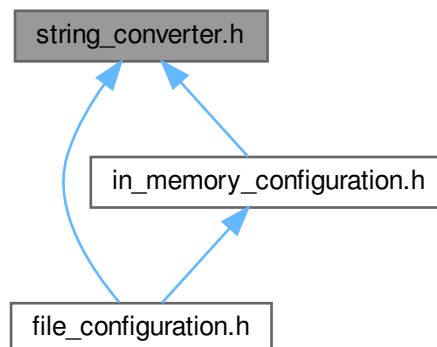
```
#include <cstdint>
```

```
#include <string>
```

Include dependency graph for string\_converter.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [StringConverter](#)

*Class that interprets the contents of a string and converts it into different types.*

### 11.643.1 Detailed Description

Interface of a class that interprets the contents of a string and converts it into different types.

#### Author

Carlos Aviles, 2010. carlos.avilesr(at)gmail.com

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [string\\_converter.h](#).

---

## 11.644 string\_converter.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002   * \file string_converter.h
00003   * \brief Interface of a class that interprets the contents of a string
00004   * and converts it into different types.
00005   * \author Carlos Aviles, 2010. carlos.avilesr(at)gmail.com
00006   *
00007   * -----
00008   *
00009   * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010   * This file is part of GNSS-SDR.
00011   *
00012   * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013   * SPDX-License-Identifier: GPL-3.0-or-later
00014   *
00015   * -----
00016   */
00017
00018
00019 #ifndef GNSS_SDR_STRING_CONVERTER_H
00020 #define GNSS_SDR_STRING_CONVERTER_H
00021
00022 #include <stdint>
00023 #include <string>
00024
00025 /** \addtogroup Core
00026   * \{ */
  
```

```

00027 /** \addtogroup Core_Receiver_Library
00028  * \{ */
00029
00030
00031 /*!
00032  * \brief Class that interprets the contents of a string
00033  * and converts it into different types.
00034  */
00035 class StringConverter
00036 {
00037 public:
00038     StringConverter() = default;
00039     ~StringConverter() = default;
00040
00041     bool convert(const std::string& value, bool default_value);
00042     int64_t convert(const std::string& value, int64_t default_value);
00043     uint64_t convert(const std::string& value, uint64_t default_value);
00044     int32_t convert(const std::string& value, int32_t default_value);
00045     uint32_t convert(const std::string& value, uint32_t default_value);
00046     int16_t convert(const std::string& value, int16_t default_value);
00047     uint16_t convert(const std::string& value, uint16_t default_value);
00048     float convert(const std::string& value, float default_value);
00049     double convert(const std::string& value, double default_value);
00050 };
00051
00052
00053 /** \} */
00054 /** \} */
00055 #endif // GNSS_SDR_STRING_CONVERTER_H

```

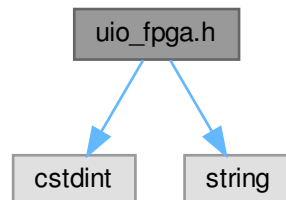
## 11.645 uio\_fpga.h File Reference

This library contains functions to determine the uio device driver file that corresponds to a hardware accelerator device name in the FPGA.

```
#include <cstdint>
```

```
#include <string>
```

Include dependency graph for uio\_fpga.h:



### Functions

- `const std::string uio_dir ("/sys/class/uio/")`
- `const std::string uio_filename ("uio")`
- `const std::string uio_subdir_name ("/name")`
- `int32_t find_uio_dev_file_name (std::string &device_file_name, const std::string &device_name, uint32_t device_num)`

*This function finds the uio device driver device file name out of the device name and the device number.*

### 11.645.1 Detailed Description

This library contains functions to determine the uio device driver file that corresponds to a hardware accelerator device name in the FPGA.



## Author

Marc Majoral, 2020. mmajoral(at)cttc.es

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [uio\\_fpga.h](#).

## 11.646 uio\_fpga.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file uio_fpga.h
00003  * \brief This library contains functions to determine the uio device driver
00004  * \file that corresponds to a hardware accelerator device name in the FPGA.
00005  * \author Marc Majoral, 2020. mmajoral(at)cttc.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_UIO_FPGA_H
00019 #define GNSS_SDR_UIO_FPGA_H
00020
00021 #include <stdint>
00022 #include <string>
00023
00024 /** \addtogroup Core
00025  * \{ */
00026 /** \addtogroup Core_Receiver_Library
00027  * \{ */
00028
00029 const std::string uio_dir("/sys/class/uio/");
00030 const std::string uio_filename("uio");
00031 const std::string uio_subdir_name("/name");
00032
00033 /*!
00034  * \brief This function finds the uio device driver device file name out of the
00035  * \device name and the device number.
00036  */
00037 int32_t find_uio_dev_file_name(std::string &device_file_name,
00038                               const std::string &device_name,
00039                               uint32_t device_num);
00040
00041
00042 /** \} */
00043 /** \} */
00044 #endif // GNSS_SDR_UIO_FPGA_H
```

## 11.647 gnss\_synchro\_monitor.h File Reference

Interface of a receiver monitoring block which allows sending a data stream with the receiver internal parameters ([Gnss\\_Synchro](#) objects) to local or remote clients over UDP.

```
#include "gnss_block_interface.h"
#include "gnss_synchro_udp_sink.h"
#include <gnuradio/block.h>
#include <gnuradio/runtime_types.h>
#include <memory>
#include <string>
#include <vector>
```



```

00025 #include <gnuradio/block.h>
00026 #include <gnuradio/runtime_types.h> // for gr_vector_void_star
00027 #include <memory>
00028 #include <string>
00029 #include <vector>
00030
00031 /** \addtogroup Core
00032  * \{ */
00033 /** \addtogroup Gnss_Synchro_Monitor core_monitor
00034  * Classes for the Gnss_Synchro monitor.
00035  * \{ */
00036
00037
00038 class gnss_synchro_monitor;
00039
00040 using gnss_synchro_monitor_sptr = gnss_shared_ptr<gnss_synchro_monitor>;
00041
00042 gnss_synchro_monitor_sptr gnss_synchro_make_monitor(int n_channels,
00043     int decimation_factor,
00044     const std::vector<std::string>& udp_ports,
00045     const std::vector<std::string>& udp_addresses,
00046     bool enable_protobuf);
00047
00048 /*!
00049  * \brief This class implements a monitoring block which allows sending
00050  * a data stream with the receiver internal parameters (Gnss_Synchro objects)
00051  * to local or remote clients over UDP.
00052  */
00053 class gnss_synchro_monitor : public gr::block
00054 {
00055 public:
00056     ~gnss_synchro_monitor() = default; //!< Default destructor
00057     void forecast(int noutput_items, gr_vector_int& ninput_items_required);
00058     int general_work(int noutput_items, gr_vector_int& ninput_items,
00059         gr_vector_const_void_star& input_items, gr_vector_void_star& output_items);
00060
00061 private:
00062     friend gnss_synchro_monitor_sptr gnss_synchro_make_monitor(int n_channels,
00063         int decimation_factor,
00064         const std::vector<std::string>& udp_ports,
00065         const std::vector<std::string>& udp_addresses,
00066         bool enable_protobuf);
00067
00068     gnss_synchro_monitor(int n_channels,
00069         int decimation_factor,
00070         const std::vector<std::string>& udp_ports,
00071         const std::vector<std::string>& udp_addresses,
00072         bool enable_protobuf);
00073
00074     std::unique_ptr<Gnss_Synchro_Udp_Sink> udp_sink_ptr;
00075     int count;
00076     int d_nchannels;
00077     int d_decimation_factor;
00078 };
00079
00080
00081 /** \} */
00082 /** \} */
00083 #endif // GNSS_SDR_GNSS_SYNCHRO_MONITOR_H

```

## 11.649 gnss\_synchro\_udp\_sink.h File Reference

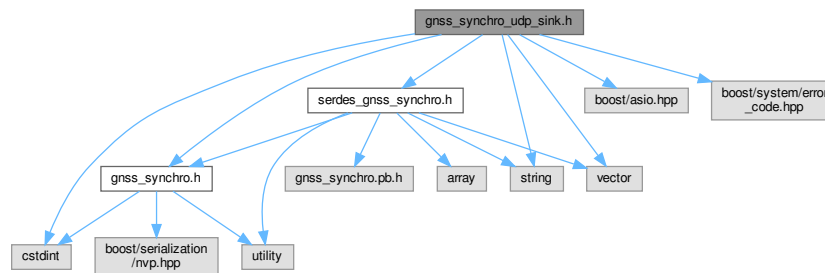
Interface of a class that sends serialized [Gnss\\_Synchro](#) objects over udp to one or multiple endpoints.

```

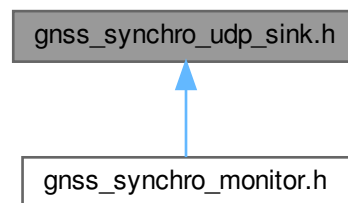
#include "gnss_synchro.h"
#include "serdes_gnss_synchro.h"
#include <boost/asio.hpp>
#include <boost/system/error_code.hpp>
#include <cstdint>
#include <string>
#include <vector>

```

Include dependency graph for `gnss_synchro_udp_sink.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Gnss\\_Synchro\\_Udp\\_Sink](#)

*This class sends serialized [Gnss\\_Synchro](#) objects over UDP to one or multiple endpoints.*

## Typedefs

- using [b\\_io\\_context](#) = `boost::asio::io_service`

## 11.649.1 Detailed Description

Interface of a class that sends serialized [Gnss\\_Synchro](#) objects over udp to one or multiple endpoints.

### Author

Álvaro Cebrián Juan, 2018. [acebrianjuan\(at\)gmail.com](mailto:acebrianjuan(at)gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gnss\\_synchro\\_udp\\_sink.h](#).

---

## 11.650 gnss\_synchro\_udp\_sink.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002   * \file gnss_synchro_udp_sink.h
00003   * \brief Interface of a class that sends serialized Gnss_Synchro objects
00004   * over udp to one or multiple endpoints

```

```

00005  * \author Álvaro Cebrián Juan, 2018. acebrianjuan(at)gmail.com
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_GNSS_SYNCHRO_UDP_SINK_H
00019 #define GNSS_SDR_GNSS_SYNCHRO_UDP_SINK_H
00020
00021 #include "gnss_synchro.h"
00022 #include "serdes_gnss_synchro.h"
00023 #include <boost/asio.hpp>
00024 #include <boost/system/error_code.hpp>
00025 #include <stdint>
00026 #include <string>
00027 #include <vector>
00028
00029 /** \addtogroup Core
00030  * \{ */
00031 /** \addtogroup Gnss_Synchro_Monitor
00032  * \{ */
00033
00034
00035 #if USE_BOOST_ASIO_IO_CONTEXT
00036 using b_io_context = boost::asio::io_context;
00037 #else
00038 using b_io_context = boost::asio::io_service;
00039 #endif
00040
00041 /*!
00042  * \brief This class sends serialized Gnss_Synchro objects
00043  * over UDP to one or multiple endpoints.
00044  */
00045 class Gnss_Synchro_Udp_Sink
00046 {
00047 public:
00048     Gnss_Synchro_Udp_Sink(const std::vector<std::string>& addresses, const std::vector<std::string>&
ports, bool enable_protobuf);
00049     bool write_gnss_synchro(const std::vector<Gnss_Synchro>& stocks);
00050
00051 private:
00052     b_io_context io_context;
00053     boost::asio::ip::udp::socket socket;
00054     boost::system::error_code error;
00055     std::vector<boost::asio::ip::udp::endpoint> endpoints;
00056     Serdes_Gnss_Synchro serdes;
00057     bool use_protobuf;
00058 };
00059
00060
00061 /** \} */
00062 /** \} */
00063 #endif // GNSS_SDR_GNSS_SYNCHRO_UDP_SINK_H

```

## 11.651 serdes\_gnss\_synchro.h File Reference

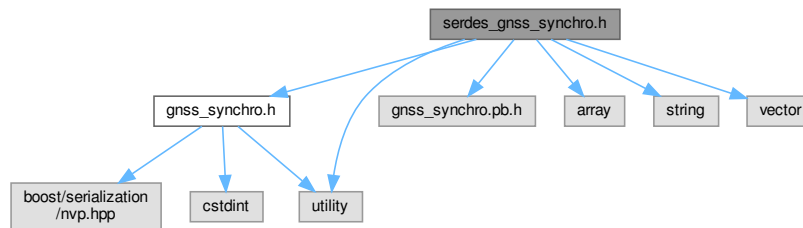
Serialization / Deserialization of [Gnss\\_Synchro](#) objects using Protocol Buffers.

```

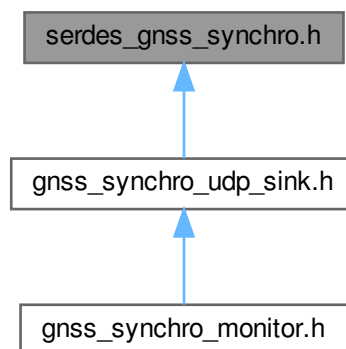
#include "gnss_synchro.h"
#include "gnss_synchro.pb.h"
#include <array>
#include <string>
#include <utility>
#include <vector>

```

Include dependency graph for `serdes_gnss_synchro.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Serdes\\_Gnss\\_Synchro](#)

*This class implements serialization and deserialization of [Gnss\\_Synchro](#) objects using Protocol Buffers.*

### 11.651.1 Detailed Description

Serialization / Deserialization of [Gnss\\_Synchro](#) objects using Protocol Buffers.

#### Author

Carles Fernandez-Prades, 2019. cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [serdes\\_gnss\\_synchro.h](#).

---

## 11.652 serdes\_gnss\_synchro.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file serdes_gnss_synchro.h
00003  * \brief Serialization / Deserialization of Gnss_Synchro objects using
00004  * Protocol Buffers

```

```

00005  * \author Carles Fernandez-Prades, 2019. cfernandez(at)cttc.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_SERDES_GNSS_SYNCHRO_H
00019 #define GNSS_SDR_SERDES_GNSS_SYNCHRO_H
00020
00021 #include "gnss_synchro.h"
00022 #include "gnss_synchro.pb.h" // file created by Protocol Buffers at compile time
00023 #include <array>
00024 #include <string>
00025 #include <utility>
00026 #include <vector>
00027
00028 /*!
00029  * \brief This class implements serialization and deserialization of
00030  * Gnss_Synchro objects using Protocol Buffers.
00031  */
00032 class Serdes_Gnss_Synchro
00033 {
00034 public:
00035     Serdes_Gnss_Synchro()
00036     {
00037         // Verify that the version of the library that we linked against is
00038         // compatible with the version of the headers we compiled against.
00039         GOOGLE_PROTOBUF_VERIFY_VERSION;
00040     }
00041
00042     ~Serdes_Gnss_Synchro()
00043     {
00044         google::protobuf::ShutdownProtobufLibrary();
00045     }
00046
00047     inline Serdes_Gnss_Synchro(const Serdes_Gnss_Synchro& other) noexcept //!< Copy constructor
00048     {
00049         this->observables = other.observables;
00050     }
00051
00052     inline Serdes_Gnss_Synchro& operator=(const Serdes_Gnss_Synchro& rhs) noexcept //!< Copy
00053     assignment operator
00054     {
00055         this->observables = rhs.observables;
00056         return *this;
00057     }
00058
00059     inline Serdes_Gnss_Synchro(Serdes_Gnss_Synchro&& other) noexcept //!< Move constructor
00060     {
00061         this->observables = std::move(other.observables);
00062     }
00063
00064     inline Serdes_Gnss_Synchro& operator=(Serdes_Gnss_Synchro&& other) noexcept //!< Move assignment
00065     operator
00066     {
00067         if (this != &other)
00068         {
00069             this->observables = std::move(other.observables);
00070         }
00071         return *this;
00072     }
00073
00074     inline std::string createProtobuffer(const std::vector<Gnss_Synchro>& vgs) //!< Serialization
00075     into a string
00076     {
00077         observables.Clear();
00078         std::string data;
00079         for (const auto& gs : vgs)
00080         {
00081             gnss_sdr::GnssSynchro* obs = observables.add_observable();
00082             char c = gs.System;
00083             const std::string sys(1, c);
00084
00085             std::array<char, 2> cc;
00086             cc[0] = gs.Signal[0];
00087             cc[1] = gs.Signal[1];
00088             const std::string sig(cc.cbegin(), cc.cend());
00089
00090             obs->set_system(sys);
00091             obs->set_signal(sig);
00092         }
00093     }

```

```

00089         obs->set_prn(gs.PRN);
00090         obs->set_channel_id(gs.Channel_ID);
00091
00092         obs->set_acq_delay_samples(gs.Acq_delay_samples);
00093         obs->set_acq_doppler_hz(gs.Acq_doppler_hz);
00094         obs->set_acq_samplestamp_samples(gs.Acq_samplestamp_samples);
00095         obs->set_acq_doppler_step(gs.Acq_doppler_step);
00096         obs->set_flag_valid_acquisition(gs.Flag_valid_acquisition);
00097
00098         obs->set_fs(gs.fs);
00099         obs->set_prompt_i(gs.Prompt_I);
00100         obs->set_prompt_q(gs.Prompt_Q);
00101         obs->set_cn0_db_hz(gs.CN0_db_hz);
00102         obs->set_carrier_doppler_hz(gs.Carrier_Doppler_hz);
00103         obs->set_carrier_phase_rads(gs.Carrier_phase_rads);
00104         obs->set_code_phase_samples(gs.Code_phase_samples);
00105         obs->set_tracking_sample_counter(gs.Tracking_sample_counter);
00106         obs->set_flag_valid_symbol_output(gs.Flag_valid_symbol_output);
00107         obs->set_correlation_length_ms(gs.correlation_length_ms);
00108
00109         obs->set_flag_valid_word(gs.Flag_valid_word);
00110         obs->set_tow_at_current_symbol_ms(gs.TOW_at_current_symbol_ms);
00111
00112         obs->set_pseudorange_m(gs.Pseudorange_m);
00113         obs->set_rx_time(gs.RX_time);
00114         obs->set_flag_valid_pseudorange(gs.Flag_valid_pseudorange);
00115         obs->set_flag_pll_180_deg_phase_locked(gs.Flag_PLL_180_deg_phase_locked);
00116         obs->set_interp_tow_ms(gs.interp_TOW_ms);
00117     }
00118     observables.SerializeToString(&data);
00119     return data;
00120 }
00121
00122 inline std::vector<Gnss_Synchro> readProtobuffer(const gnss_sdr::Observables& obs) const //!  

Deserialization
00123 {
00124     std::vector<Gnss_Synchro> vgs;
00125     vgs.reserve(obs.observable_size());
00126     for (int i = 0; i < obs.observable_size(); ++i)
00127     {
00128         const gnss_sdr::GnssSynchro& gs_read = obs.observable(i);
00129         Gnss_Synchro gs = Gnss_Synchro();
00130         gs.System = gs_read.system()[0];
00131         gs.Signal[0] = gs_read.signal()[0];
00132         gs.Signal[1] = gs_read.signal()[1];
00133         gs.Signal[2] = '\0';
00134         gs.PRN = gs_read.prn();
00135         gs.Channel_ID = gs_read.channel_id();
00136
00137         gs.Acq_delay_samples = gs_read.acq_delay_samples();
00138         gs.Acq_doppler_hz = gs_read.acq_doppler_hz();
00139         gs.Acq_samplestamp_samples = gs_read.acq_samplestamp_samples();
00140         gs.Acq_doppler_step = gs_read.acq_doppler_step();
00141         gs.Flag_valid_acquisition = gs_read.flag_valid_acquisition();
00142
00143         gs.fs = gs_read.fs();
00144         gs.Prompt_I = gs_read.prompt_i();
00145         gs.Prompt_Q = gs_read.prompt_q();
00146         gs.CN0_db_hz = gs_read.cn0_db_hz();
00147         gs.Carrier_Doppler_hz = gs_read.carrier_doppler_hz();
00148         gs.Carrier_phase_rads = gs_read.carrier_phase_rads();
00149         gs.Code_phase_samples = gs_read.code_phase_samples();
00150         gs.Tracking_sample_counter = gs_read.tracking_sample_counter();
00151         gs.Flag_valid_symbol_output = gs_read.flag_valid_symbol_output();
00152         gs.correlation_length_ms = gs_read.correlation_length_ms();
00153
00154         gs.Flag_valid_word = gs_read.flag_valid_word();
00155         gs.TOW_at_current_symbol_ms = gs_read.tow_at_current_symbol_ms();
00156
00157         gs.Pseudorange_m = gs_read.pseudorange_m();
00158         gs.RX_time = gs_read.rx_time();
00159         gs.Flag_valid_pseudorange = gs_read.flag_valid_pseudorange();
00160         gs.Flag_PLL_180_deg_phase_locked = gs_read.flag_pll_180_deg_phase_locked();
00161         gs.interp_TOW_ms = gs_read.interp_tow_ms();
00162
00163         vgs.push_back(gs);
00164     }
00165     return vgs;
00166 }
00167
00168 private:
00169     gnss_sdr::Observables observables{};
00170 };
00171
00172 #endif // GNSS_SDR_SERDES_GNSS_SYNCHRO_H

```



## 11.653 concurrent\_map.h File Reference

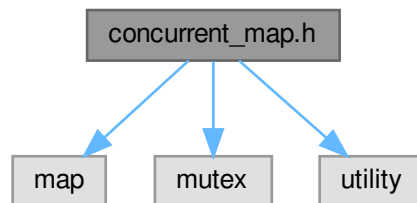
Interface of a thread-safe std::map.

```
#include <map>
```

```
#include <mutex>
```

```
#include <utility>
```

Include dependency graph for concurrent\_map.h:



### Classes

- class [Concurrent\\_Map< Data >](#)

*This class implements a thread-safe std::map.*

### 11.653.1 Detailed Description

Interface of a thread-safe std::map.

#### Author

Javier Arribas, 2011. [jarribas\(at\)cttc.es](mailto:jarribas(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [concurrent\\_map.h](#).

---

## 11.654 concurrent\_map.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file concurrent_map.h
00003  * \brief Interface of a thread-safe std::map
00004  * \author Javier Arribas, 2011. jarribas\(at\)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_CONCURRENT_MAP_H
00018 #define GNSS_SDR_CONCURRENT_MAP_H
00019
00020 #include <map>
00021 #include <mutex>
00022 #include <utility>
00023
00024 /** \addtogroup Core

```

```

00025  * \{ */
00026  /** \addtogroup Core_Receiver core_receiver
00027  * \{ */
00028
00029
00030  template <typename Data>
00031
00032
00033  /*!
00034  * \brief This class implements a thread-safe std::map
00035  *
00036  */
00037  class Concurrent_Map
00038  {
00039  public:
00040      void write(int key, Data const& data)
00041      {
00042          std::lock_guard<std::mutex> lock(the_mutex);
00043          auto data_iter = the_map.find(key);
00044          if (data_iter != the_map.end())
00045          {
00046              data_iter->second = data; // update
00047          }
00048          else
00049          {
00050              the_map.insert(std::pair<int, Data>(key, data)); // insert does not overwrite if the
item already exists in the map!
00051          }
00052      }
00053
00054      std::map<int, Data> get_map_copy() const&
00055      {
00056          std::lock_guard<std::mutex> lock(the_mutex);
00057          return the_map; // This implicitly creates a copy
00058      }
00059
00060      std::map<int, Data> get_map_copy() &&
00061      {
00062          std::lock_guard<std::mutex> lock(the_mutex);
00063          return std::move(the_map);
00064      }
00065
00066      size_t size() const
00067      {
00068          std::lock_guard<std::mutex> lock(the_mutex);
00069          return the_map.size();
00070      }
00071
00072      bool read(int key, Data& p_data) const
00073      {
00074          std::lock_guard<std::mutex> lock(the_mutex);
00075          auto data_iter = the_map.find(key);
00076          if (data_iter != the_map.end())
00077          {
00078              p_data = data_iter->second;
00079              return true;
00080          }
00081          return false;
00082      }
00083
00084  private:
00085      std::map<int, Data> the_map;
00086      mutable std::mutex the_mutex;
00087  };
00088
00089
00090  /** \} */
00091  /** \} */
00092  #endif // GNSS_SDR_CONCURRENT_MAP_H

```

## 11.655 concurrent\_queue.h File Reference

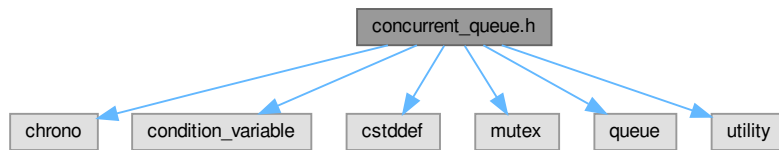
Interface of a thread-safe std::queue.

```

#include <chrono>
#include <condition_variable>
#include <cstdint>
#include <mutex>
#include <queue>
#include <utility>

```

Include dependency graph for concurrent\_queue.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Concurrent\\_Queue< Data >](#)  
*This class implements a thread-safe std::queue.*

## 11.655.1 Detailed Description

Interface of a thread-safe std::queue.

### Author

Javier Arribas, 2011. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [concurrent\\_queue.h](#).

---

## 11.656 concurrent\_queue.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file concurrent_queue.h
00003  * \brief Interface of a thread-safe std::queue
00004  * \author Javier Arribas, 2011. jarribas(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_CONCURRENT_QUEUE_H
00018 #define GNSS_SDR_CONCURRENT_QUEUE_H
00019
00020 #include <chrono>
00021 #include <condition_variable>
00022 #include <cstdint>
00023 #include <mutex>
00024 #include <queue>
00025 #include <utility>
00026
00027 /** \addtogroup Core
00028  * \{ */
00029 /** \addtogroup Core_Receiver
00030  * \{ */
  
```

```

00031
00032
00033 template <typename Data>
00034
00035 /*!
00036  * \brief This class implements a thread-safe std::queue
00037  */
00038 class Concurrent_Queue
00039 {
00040 public:
00041     void push(const Data& data)
00042     {
00043     {
00044         std::lock_guard<std::mutex> lock(the_mutex);
00045         the_queue.push(data);
00046     }
00047     the_condition_variable.notify_one();
00048 }
00049
00050     void push(Data&& data)
00051     {
00052     {
00053         std::lock_guard<std::mutex> lock(the_mutex);
00054         the_queue.push(std::move(data));
00055     }
00056     the_condition_variable.notify_one();
00057 }
00058
00059     bool empty() const noexcept
00060     {
00061         return size() == 0;
00062     }
00063
00064     size_t size() const noexcept
00065     {
00066         std::lock_guard<std::mutex> lock(the_mutex);
00067         return the_queue.size();
00068     }
00069
00070     void clear()
00071     {
00072         std::lock_guard<std::mutex> lock(the_mutex);
00073         std::queue<Data>().swap(the_queue);
00074     }
00075
00076     bool try_pop(Data& popped_value)
00077     {
00078         std::lock_guard<std::mutex> lock(the_mutex);
00079         if (the_queue.empty())
00080         {
00081             return false;
00082         }
00083         popped_value = std::move(the_queue.front());
00084         the_queue.pop();
00085         return true;
00086     }
00087
00088     void wait_and_pop(Data& popped_value)
00089     {
00090         std::unique_lock<std::mutex> lock(the_mutex);
00091         the_condition_variable.wait(lock, [this] { return !the_queue.empty(); });
00092         popped_value = std::move(the_queue.front());
00093         the_queue.pop();
00094     }
00095
00096     bool timed_wait_and_pop(Data& popped_value, int wait_ms)
00097     {
00098         std::unique_lock<std::mutex> lock(the_mutex);
00099         if (!the_condition_variable.wait_for(lock,
00100             std::chrono::milliseconds(wait_ms),
00101             [this] { return !the_queue.empty(); }))
00102         {
00103             return false;
00104         }
00105         popped_value = std::move(the_queue.front());
00106         the_queue.pop();
00107         return true;
00108     }
00109 private:
00110     std::queue<Data> the_queue;
00111     mutable std::mutex the_mutex;
00112     std::condition_variable the_condition_variable;
00113 };
00114
00115
00116
00117 /** \} */

```

```
00118 /** \} */
00119 #endif // GNSS_SDR_CONCURRENT_QUEUE_H
```

## 11.657 control\_thread.h File Reference

Interface of the receiver control plane.

```
#include "agnss_ref_location.h"
#include "agnss_ref_time.h"
#include "channel_event.h"
#include "command_event.h"
#include "concurrent_queue.h"
#include "gnss_sdr_supl_client.h"
#include "tcp_cmd_interface.h"
#include <pmt/pmt.h>
#include <array>
#include <stddef>
#include <memory>
#include <string>
#include <thread>
#include <typeinfo>
#include <utility>
#include <vector>
```

Include dependency graph for control\_thread.h:



### Classes

- class [ControlThread](#)

*This class represents the main thread of the application, so the name is [ControlThread](#). This is the GNSS Receiver Control Plane: it connects the flowgraph, starts running it, and while it does not stop, reads the control messages generated by the blocks, processes them, and applies the corresponding actions.*

### 11.657.1 Detailed Description

Interface of the receiver control plane.

#### Author

Carlos Aviles, 2010. carlos.avilesr(at)gmail.com

GNSS Receiver Control Plane: connects the flowgraph, starts running it, and while it does not stop, reads the control messages generated by the blocks, processes them, and applies the corresponding actions.

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [control\\_thread.h](#).

## 11.658 control\_thread.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002 * \file control_thread.h
00003 * \brief Interface of the receiver control plane
00004 * \author Carlos Aviles, 2010. carlos.avilesr(at)gmail.com
00005 *
00006 * GNSS Receiver Control Plane: connects the flowgraph, starts running it,
00007 * and while it does not stop, reads the control messages generated by the blocks,
```

```

00008  * processes them, and applies the corresponding actions.
00009  *
00010  * -----
00011  *
00012  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00013  * This file is part of GNSS-SDR.
00014  *
00015  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00016  * SPDX-License-Identifier: GPL-3.0-or-later
00017  *
00018  * -----
00019  */
00020
00021 #ifndef GNSS_SDR_CONTROL_THREAD_H
00022 #define GNSS_SDR_CONTROL_THREAD_H
00023
00024 #include "agnss_ref_location.h" // for Agnss_Ref_Location
00025 #include "agnss_ref_time.h" // for Agnss_Ref_Time
00026 #include "channel_event.h" // for channel_event_sptr
00027 #include "command_event.h" // for command_event_sptr
00028 #include "concurrent_queue.h" // for Concurrent_Queue
00029 #include "gnss_sdr_supl_client.h" // for Gnss_Sdr_Supl_Client
00030 #include "tcp_cmd_interface.h" // for TcpCmdInterface
00031 #include <pmt/pmt.h>
00032 #include <array> // for array
00033 #include <cstdint> // for size_t
00034 #include <memory> // for shared_ptr
00035 #include <string> // for string
00036 #include <thread> // for std::thread
00037 #include <typeinfo> // for std::type_info, typeid
00038 #include <utility> // for pair
00039 #include <vector> // for vector
00040
00041 #ifdef ENABLE_FPGA
00042 #include <boost/thread.hpp> // for boost::thread
00043 #endif
00044
00045 /** \addtogroup Core Core GNSS Receiver
00046  * Core GNSS Receiver.
00047  * \{ */
00048 /** \addtogroup Core_Receiver
00049  * Classes for the core GNSS receiver.
00050  * \{ */
00051
00052
00053 class ConfigurationInterface;
00054 class GNSSFlowgraph;
00055 class Gnss_Satellite;
00056
00057
00058 /*!
00059  * \brief This class represents the main thread of the application, so the name is ControlThread.
00060  * This is the GNSS Receiver Control Plane: it connects the flowgraph, starts running it,
00061  * and while it does not stop, reads the control messages generated by the blocks,
00062  * processes them, and applies the corresponding actions.
00063  */
00064 class ControlThread
00065 {
00066 public:
00067     static ControlThread *me;
00068     /*!
00069      * \brief Default constructor
00070      */
00071     ControlThread();
00072
00073     /*!
00074      * \brief Constructor that initializes the class with parameters
00075      *
00076      * \param[in] configuration Pointer to a ConfigurationInterface
00077      */
00078     explicit ControlThread(std::shared_ptr<ConfigurationInterface> configuration);
00079
00080     /*!
00081      * \brief Destructor
00082      */
00083     ~ControlThread();
00084
00085     /*! \brief Runs the control thread
00086      *
00087      * This is the main loop that reads and process the control messages:
00088      *
00089      * - Connect the GNSS receiver flowgraph;
00090      *
00091      * - Start the GNSS receiver flowgraph;
00092      *
00093      * while (flowgraph_>running() && !stop_){
00094      *

```

```

00095     * - Read control messages and process them; }
00096     */
00097     int run();
00098
00099     /*!
00100     * \brief Sets the control_queue
00101     *
00102     * \param[in] std::shared_ptr<Concurrent_Queue<pmt::pmt_t> control_queue
00103     */
00104     void set_control_queue(std::shared_ptr<Concurrent_Queue<pmt::pmt_t>> control_queue);
00105
00106     unsigned int processed_control_messages() const
00107     {
00108         return processed_control_messages_;
00109     }
00110
00111     unsigned int applied_actions() const
00112     {
00113         return applied_actions_;
00114     }
00115
00116     /*!
00117     * \brief Instantiates a flowgraph
00118     *
00119     * \return Returns a smart pointer to a flowgraph object
00120     */
00121     std::shared_ptr<GNSSFlowgraph> flowgraph()
00122     {
00123         return flowgraph_;
00124     }
00125
00126 private:
00127     /*
00128     * Callback function for handling signals.
00129     * sig identifier of signal
00130     */
00131     static void handle_signal(int sig);
00132
00133     void init();
00134
00135     void apply_action(unsigned int what);
00136
00137     /*
00138     * New receiver event dispatcher
00139     */
00140     void event_dispatcher(bool &valid_event, pmt::pmt_t &msg);
00141
00142     // Read {ephemeris, iono, utc, ref loc, ref time} assistance from a local XML file previously
    recorded
00143     bool read_assistance_from_XML();
00144
00145     /*
00146     * Blocking function that reads the GPS assistance queue
00147     */
00148     void gps_acq_assist_data_collector() const;
00149
00150     /*
00151     * Compute elevations for the specified time and position for all the available satellites in
    ephemeris and almanac queues
00152     * returns a vector filled with the available satellites ordered from high elevation to low
    elevation angle.
00153     */
00154     std::vector<std::pair<int, Gnss_Satellite> get_visible_sats(time_t rx_utc_time, const
    std::array<float, 3> &LLH);
00155
00156     /*
00157     * Read initial GNSS assistance from SUPL server or local XML files
00158     */
00159     void assist_GNSS();
00160
00161     void telecommand_listener();
00162     void keyboard_listener();
00163     void message_queue_listener();
00164     void print_help_at_exit() const;
00165
00166     // default filename for assistance data
00167     const std::string eph_default_xml_filename_ = "./gps_ephemeris.xml";
00168     const std::string utc_default_xml_filename_ = "./gps_utc_model.xml";
00169     const std::string iono_default_xml_filename_ = "./gps_iono.xml";
00170     const std::string ref_time_default_xml_filename_ = "./gps_ref_time.xml";
00171     const std::string ref_location_default_xml_filename_ = "./gps_ref_location.xml";
00172     const std::string eph_gal_default_xml_filename_ = "./gal_ephemeris.xml";
00173     const std::string eph_cnav_default_xml_filename_ = "./gps_cnav_ephemeris.xml";
00174     const std::string gal_iono_default_xml_filename_ = "./gal_iono.xml";
00175     const std::string gal_utc_default_xml_filename_ = "./gal_utc_model.xml";
00176     const std::string cnav_utc_default_xml_filename_ = "./gps_cnav_utc_model.xml";
00177     const std::string eph_glo_gnav_default_xml_filename_ = "./glo_gnav_ephemeris.xml";

```

```

00178     const std::string glo_utc_default_xml_filename_ = "./glo_utc_model.xml";
00179     const std::string gal_almanac_default_xml_filename_ = "./gal_almanac.xml";
00180     const std::string gps_almanac_default_xml_filename_ = "./gps_almanac.xml";
00181
00182     const std::string control_message_queue_name_ = "receiver_control_queue";
00183
00184     const size_t channel_event_type_hash_code_ = typeid(channel_event_sptr).hash_code();
00185     const size_t command_event_type_hash_code_ = typeid(command_event_sptr).hash_code();
00186
00187     std::shared_ptr<ConfigurationInterface> configuration_;
00188     std::shared_ptr<Concurrent_Queue<pmt::pmt_t>> control_queue_;
00189     std::shared_ptr<GNSSFlowgraph> flowgraph_;
00190
00191     std::thread cmd_interface_thread_;
00192     std::thread keyboard_thread_;
00193     std::thread message_queue_thread_;
00194     std::thread gps_acq_assist_data_collector_thread_;
00195
00196 #ifdef ENABLE_FPGA
00197     boost::thread fpga_helper_thread_;
00198 #endif
00199
00200     TcpCmdInterface cmd_interface_;
00201
00202     // SUPL assistance classes
00203     Gnss_Sdr_Supl_Client supl_client_acquisition_;
00204     Gnss_Sdr_Supl_Client supl_client_ephemeris_;
00205     int supl_mcc_; // Current network MCC (Mobile country code), 3 digits.
00206     int supl_mns_; // Current network MNC (Mobile Network code), 2 or 3 digits.
00207     int supl_lac_; // Current network LAC (Location area code), 16 bits, 1-65520 are valid values.
00208     int supl_ci_; // Cell Identity (16 bits, 0-65535 are valid values).
00209
00210     Agnss_Ref_Location agnss_ref_location_;
00211     Agnss_Ref_Time agnss_ref_time_;
00212
00213     unsigned int processed_control_messages_;
00214     unsigned int applied_actions_;
00215
00216     bool well_formatted_configuration_;
00217     bool conf_file_has_section_;
00218     bool conf_file_has_mandatory_globals_;
00219     bool conf_has_signal_sources_;
00220     bool conf_has_observables_;
00221     bool conf_has_pvt_;
00222     bool receiver_on_standby_;
00223     bool stop_;
00224     bool restart_;
00225     bool telecommand_enabled_;
00226     bool pre_2009_file_; // to override the system time to postprocess old gnss records and avoid
    wrong week rollover
00227 };
00228
00229 /** */
00230 /** */
00231 /** */
00232 #endif // GNSS_SDR_CONTROL_THREAD_H

```

## 11.659 file\_configuration.h File Reference

A [ConfigurationInterface](#) that reads the configuration from a file.

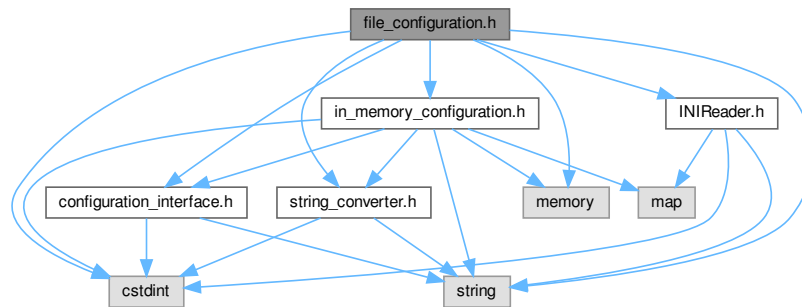
```

#include "INIReader.h"
#include "configuration_interface.h"
#include "in_memory_configuration.h"
#include "string_converter.h"
#include <stdint>
#include <memory>
#include <string>

```



Include dependency graph for file\_configuration.h:



## Classes

- class [FileConfiguration](#)

*This class is an implementation of the interface [ConfigurationInterface](#).*

## 11.659.1 Detailed Description

A [ConfigurationInterface](#) that reads the configuration from a file.

### Author

Carlos Aviles, 2010. carlos.avilesr(at)gmail.com

This implementation has a text file as the source for the values of the parameters. The file is in the INI format, containing sections and pairs of names and values. For more information about the INI format, see [https://en.wikipedia.org/wiki/INI\\_file](https://en.wikipedia.org/wiki/INI_file)

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [file\\_configuration.h](#).

## 11.660 file\_configuration.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file file_configuration.h
00003  * \brief A ConfigurationInterface that reads the configuration from a file.
00004  * \author Carlos Aviles, 2010. carlos.avilesr(at)gmail.com
00005  *
00006  * This implementation has a text file as the source for the values of the parameters.
00007  * The file is in the INI format, containing sections and pairs of names and values.
00008  * For more information about the INI format, see https://en.wikipedia.org/wiki/INI_file
00009  *
00010  * -----
00011  *
00012  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00013  * This file is part of GNSS-SDR.
00014  *
00015  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00016  * SPDX-License-Identifier: GPL-3.0-or-later
00017  *
00018  * -----
00019  */
00020
00021
00022 #ifndef GNSS_SDR_FILE_CONFIGURATION_H
00023 #define GNSS_SDR_FILE_CONFIGURATION_H
00024
00025 #include "INIReader.h"
00026 #include "configuration_interface.h"
00027 #include "in_memory_configuration.h"
00028 #include "string_converter.h"

```

```

00029 #include <cstdint>
00030 #include <memory>
00031 #include <string>
00032
00033 /** \addtogroup Core
00034  * \{ */
00035 /** \addtogroup Core_Receiver
00036  * \{ */
00037
00038
00039 /*!
00040 * \brief This class is an implementation of the interface ConfigurationInterface
00041 *
00042 * Derived from ConfigurationInterface, this class implements an interface
00043 * to a configuration file. This implementation has a text file as the source
00044 * for the values of the parameters.
00045 * The file is in the INI format, containing sections and pairs of names and values.
00046 * For more information about the INI format, see https://en.wikipedia.org/wiki/INI\_file
00047 */
00048 class FileConfiguration : public ConfigurationInterface
00049 {
00050 public:
00051     explicit FileConfiguration(std::string filename);
00052     FileConfiguration();
00053     ~FileConfiguration() = default;
00054     std::string property(std::string property_name, std::string default_value) const override;
00055     bool property(std::string property_name, bool default_value) const override;
00056     int64_t property(std::string property_name, int64_t default_value) const override;
00057     uint64_t property(std::string property_name, uint64_t default_value) const override;
00058     int32_t property(std::string property_name, int32_t default_value) const override;
00059     uint32_t property(std::string property_name, uint32_t default_value) const override;
00060     int16_t property(std::string property_name, int16_t default_value) const override;
00061     uint16_t property(std::string property_name, uint16_t default_value) const override;
00062     float property(std::string property_name, float default_value) const override;
00063     double property(std::string property_name, double default_value) const override;
00064     void set_property(std::string property_name, std::string value) override;
00065     bool is_present(const std::string& property_name) const;
00066     bool has_section() const;
00067
00068 private:
00069     void init();
00070     std::string filename_;
00071     std::unique_ptr<INIReader> ini_reader_;
00072     std::unique_ptr<InMemoryConfiguration> overrided_;
00073     std::unique_ptr<StringConverter> converter_;
00074     int error_{};
00075 };
00076
00077
00078 /** \} */
00079 /** \} */
00080 #endif // GNSS_SDR_FILE_CONFIGURATION_H

```

## 11.661 gnss\_block\_factory.h File Reference

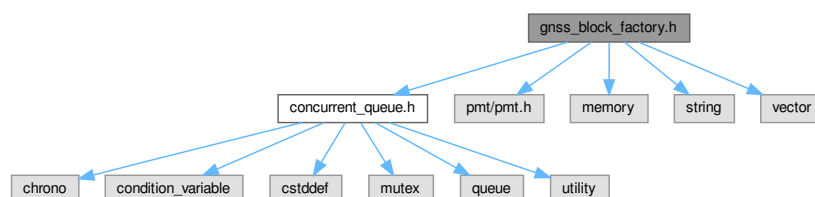
Interface of a factory that returns smart pointers to GNSS blocks.

```

#include "concurrent_queue.h"
#include <pmt/pmt.h>
#include <memory>
#include <string>
#include <vector>

```

Include dependency graph for gnss\_block\_factory.h:



**Classes**

- class [GNSSBlockFactory](#)

*Class that produces all kinds of GNSS blocks.*

**11.661.1 Detailed Description**

Interface of a factory that returns smart pointers to GNSS blocks.

**Author**

Carlos Aviles, 2010. carlos.avilesr(at)gmail.com Luis Esteve, 2011. luis(at)epsilon-formacion.com Javier Arribas, 2011. jarribas(at)cttc.es Carles Fernandez-Prades, 2014-2020. cfernandez(at)cttc.es

This class encapsulates the complexity behind the instantiation of GNSS blocks.

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [gnss\\_block\\_factory.h](#).

**11.662 gnss\_block\_factory.h**

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file gnss_block_factory.h
00003  * \brief Interface of a factory that returns smart pointers to GNSS blocks.
00004  * \author Carlos Aviles, 2010. carlos.avilesr(at)gmail.com
00005  *         Luis Esteve, 2011. luis(at)epsilon-formacion.com
00006  *         Javier Arribas, 2011. jarribas(at)cttc.es
00007  *         Carles Fernandez-Prades, 2014-2020. cfernandez(at)cttc.es
00008  *
00009  * This class encapsulates the complexity behind the instantiation
00010  * of GNSS blocks.
00011  *
00012  *
00013  * -----
00014  *
00015  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00016  * This file is part of GNSS-SDR.
00017  *
00018  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00019  * SPDX-License-Identifier: GPL-3.0-or-later
00020  *
00021  * -----
00022  */
00023
00024 #ifndef GNSS_SDR_BLOCK_FACTORY_H
00025 #define GNSS_SDR_BLOCK_FACTORY_H
00026
00027 #include "concurrent_queue.h"
00028 #include <pmt/pmt.h>
00029 #include <memory> // for unique_ptr
00030 #include <string> // for string
00031 #include <vector> // for vector
00032
00033 /** \addtogroup Core
00034  * \{ */
00035 /** \addtogroup Core_Receiver
00036  * \{ */
00037
00038
00039 class ConfigurationInterface;
00040 class GNSSBlockInterface;
00041 class SignalSourceInterface;
00042 class AcquisitionInterface;
00043 class TrackingInterface;
00044 class TelemetryDecoderInterface;
00045
00046 /*!
00047 * \brief Class that produces all kinds of GNSS blocks
00048 */
00049 class GNSSBlockFactory
00050 {
00051 public:
00052     GNSSBlockFactory() = default;
00053     ~GNSSBlockFactory() = default;
00054
00055     std::unique_ptr<SignalSourceInterface> GetSignalSource(const ConfigurationInterface*
configuration,
```

```

00056         Concurrent_Queue<pmt::pmt_t>* queue, int ID = -1);
00057
00058     std::unique_ptr<GNSSBlockInterface> GetSignalConditioner(const ConfigurationInterface*
configuration, int ID = -1);
00059
00060     std::unique_ptr<std::vector<std::unique_ptr<GNSSBlockInterface>>> GetChannels(const
ConfigurationInterface* configuration,
00061         Concurrent_Queue<pmt::pmt_t>* queue);
00062
00063     std::unique_ptr<GNSSBlockInterface> GetObservables(const ConfigurationInterface* configuration);
00064
00065     std::unique_ptr<GNSSBlockInterface> GetPVT(const ConfigurationInterface* configuration);
00066
00067     /*!
00068     * \brief Returns the block with the required role implementation and its configuration parameters
00069     */
00070     std::unique_ptr<GNSSBlockInterface> GetBlock(const ConfigurationInterface* configuration,
00071         const std::string& role,
00072         unsigned int in_streams,
00073         unsigned int out_streams,
00074         Concurrent_Queue<pmt::pmt_t>* queue = nullptr);
00075
00076 private:
00077     std::unique_ptr<GNSSBlockInterface> GetChannel(
00078         const ConfigurationInterface* configuration,
00079         const std::string& signal,
00080         int channel,
00081         Concurrent_Queue<pmt::pmt_t>* queue);
00082
00083     std::unique_ptr<AcquisitionInterface> GetAcqBlock(
00084         const ConfigurationInterface* configuration,
00085         const std::string& role,
00086         unsigned int in_streams,
00087         unsigned int out_streams);
00088
00089     std::unique_ptr<TrackingInterface> GetTrkBlock(
00090         const ConfigurationInterface* configuration,
00091         const std::string& role,
00092         unsigned int in_streams,
00093         unsigned int out_streams);
00094
00095     std::unique_ptr<TelemetryDecoderInterface> GetTlmBlock(
00096         const ConfigurationInterface* configuration,
00097         const std::string& role,
00098         unsigned int in_streams,
00099         unsigned int out_streams);
00100 };
00101
00102
00103 /** @} */
00104 /** @} */
00105 #endif // GNSS_SDR_BLOCK_FACTORY_H

```

## 11.663 gnss\_flowgraph.h File Reference

Interface of a GNSS receiver flow graph.

```

#include "channel_status_msg_receiver.h"
#include "concurrent_queue.h"
#include "galileo_e6_has_msg_receiver.h"
#include "galileo_tow_map.h"
#include "gnss_sdr_sample_counter.h"
#include "gnss_signal.h"
#include "osnma_msg_receiver.h"
#include "pvt_interface.h"
#include <gnuradio/blocks/null_sink.h>
#include <gnuradio/runtime_types.h>
#include <pmt/pmt.h>
#include <list>
#include <map>
#include <memory>
#include <mutex>
#include <string>
#include <utility>
#include <vector>

```

Include dependency graph for gnss\_flowgraph.h:



## Classes

- class [GNSSFlowgraph](#)

*This class represents a GNSS flow graph.*

## 11.663.1 Detailed Description

Interface of a GNSS receiver flow graph.

### Author

Carlos Aviles, 2010. [carlos.avilesr\(at\)gmail.com](mailto:carlos.avilesr(at)gmail.com) Luis Esteve, 2011. [luis\(at\)epsilon-formacion.com](mailto:luis(at)epsilon-formacion.com) Carles Fernandez-Prades, 2014-2020. [cfernandez\(at\)cttc.es](mailto:cfernandez(at)cttc.es) Álvaro Cebrián Juan, 2018. [acebrianjuan\(at\)gmail.com](mailto:acebrianjuan(at)gmail.com)

It contains a signal source, a signal conditioner, a set of channels, an observables block and a pvt.

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [gnss\\_flowgraph.h](#).

## 11.664 gnss\_flowgraph.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file gnss_flowgraph.h
00003  * \brief Interface of a GNSS receiver flow graph.
00004  * \author Carlos Aviles, 2010. carlos.avilesr\(at\)gmail.com
00005  *         Luis Esteve, 2011. luis\(at\)epsilon-formacion.com
00006  *         Carles Fernandez-Prades, 2014-2020. cfernandez\(at\)cttc.es
00007  *         Álvaro Cebrián Juan, 2018. acebrianjuan\(at\)gmail.com
00008  *
00009  * It contains a signal source,
00010  * a signal conditioner, a set of channels, an observables block and a pvt.
00011  *
00012  *
00013  * -----
00014  *
00015  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00016  * This file is part of GNSS-SDR.
00017  *
00018  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00019  * SPDX-License-Identifier: GPL-3.0-or-later
00020  *
00021  * -----
00022  */
00023
00024 #ifndef GNSS_SDR_GNSS_FLOWGRAPH_H
00025 #define GNSS_SDR_GNSS_FLOWGRAPH_H
00026
00027 #include "channel_status_msg_receiver.h"
00028 #include "concurrent_queue.h"
00029 #include "galileo_e6_has_msg_receiver.h"
00030 #include "galileo_tow_map.h"
00031 #include "gnss_sdr_sample_counter.h"
00032 #include "gnss_signal.h"
00033 #include "osnma_msg_receiver.h"
00034 #include "pvt_interface.h"
00035 #include <gnuradio/blocks/null_sink.h> // for null_sink
00036 #include <gnuradio/runtime_types.h> // for basic_block_sptr, top_block_sptr
00037 #include <pmt/pmt.h> // for pmt_t
00038 #include <list> // for list
00039 #include <map> // for map
00040 #include <memory> // for for shared_ptr, dynamic_pointer_cast
00041 #include <mutex> // for mutex
00042 #include <string> // for string
```

```

00043 #include <utility> // for pair
00044 #include <vector> // for vector
00045 #if ENABLE_FPGA
00046 #include "gnss_sdr_fpga_sample_counter.h"
00047 #endif
00048
00049 /** \addtogroup Core
00050 * \{ */
00051 /** \addtogroup Core_Receiver
00052 * \{ */
00053
00054
00055 class ChannelInterface;
00056 class ConfigurationInterface;
00057 class GNSSBlockInterface;
00058 class Gnss_Satellite;
00059 class SignalSourceInterface;
00060
00061 /*! \brief This class represents a GNSS flow graph.
00062 *
00063 * It contains a signal source,
00064 * a signal conditioner, a set of channels, a PVT and an output filter.
00065 */
00066 class GNSSFlowgraph
00067 {
00068 public:
00069     /*!
00070      * \brief Constructor that initializes the receiver flow graph
00071      */
00072     GNSSFlowgraph(std::shared_ptr<ConfigurationInterface> configuration,
00073                   std::shared_ptr<Concurrent_Queue<pmt::pmt_t>> queue);
00074
00075     /*!
00076      * \brief Destructor
00077      */
00078     ~GNSSFlowgraph();
00079
00080     /*!
00081      * \brief Start the flow graph
00082      */
00083     void start();
00084
00085     /*!
00086      * \brief Stop the flow graph
00087      */
00088     void stop();
00089
00090     /*!
00091      * \brief Connects the defined blocks in the flow graph
00092      * Signal Source > Signal conditioner > Channels » Observables » PVT > Output filter
00093      */
00094     void connect();
00095
00096     /*!
00097      * \brief Disconnect the blocks in the flow graph
00098      */
00099     void disconnect();
00100
00101     /*!
00102      * \brief Wait for a flowgraph to complete.
00103      *
00104      * Flowgraphs complete when either
00105      * (1) all blocks indicate that they are done, or
00106      * (2) after stop() has been called to request shutdown.
00107      */
00108     void wait();
00109
00110     /*!
00111      * \brief Manage satellite acquisition
00112      *
00113      * \param[in] who Channel ID
00114      */
00115     void acquisition_manager(unsigned int who);
00116
00117     /*!
00118      * \brief Applies an action to the flow graph
00119      *
00120      * \param[in] who Who generated the action
00121      * \param[in] what What is the action. 0: acquisition failed; 1: acquisition success; 2: tracking
00122      * lost
00123      */
00124     void apply_action(unsigned int who, unsigned int what);
00125
00126     /*!
00127      * \brief Set flow graph configuratiob
00128      */

```

```

00128     void set_configuration(const std::shared_ptr<ConfigurationInterface>& configuration);
00129
00130     bool connected() const
00131     {
00132         return connected_;
00133     }
00134
00135     bool running() const
00136     {
00137         return running_;
00138     }
00139
00140     /*!
00141      * \brief Sends a GNU Radio asynchronous message from telemetry to PVT
00142      *
00143      * It is used to assist the receiver with external ephemeris data
00144      */
00145     bool send_telemetry_msg(const pmt::pmt_t& msg);
00146
00147     /*!
00148      * \brief Returns a smart pointer to the PVT object
00149      */
00150     std::shared_ptr<PvtInterface> get_pvt()
00151     {
00152         return std::dynamic_pointer_cast<PvtInterface>(pvt_);
00153     }
00154
00155     /*!
00156      * \brief Prioritize visible satellites in the specified vector
00157      */
00158     void prioritize_satellites(const std::vector<std::pair<int, Gnss_Satellite>& visible_satellites);
00159
00160     #if ENABLE_FPGA
00161         void start_acquisition_helper();
00162
00163         void perform_hw_reset();
00164     #endif
00165
00166     private:
00167         void init(); // Populates the SV PRN list available for acquisition and tracking
00168         int connect_desktop_flowgraph();
00169
00170         int connect_signal_sources();
00171         int connect_signal_conditioners();
00172         int connect_channels();
00173         int connect_observables();
00174         int connect_pvt();
00175         int connect_sample_counter();
00176         int connect_galileo_tow_map();
00177
00178         int connect_signal_sources_to_signal_conditioners();
00179         int connect_signal_conditioners_to_channels();
00180         int connect_channels_to_observables();
00181         int connect_observables_to_pvt();
00182         int connect_monitors();
00183         int connect_osnma();
00184         int connect_gal_e6_has();
00185         int connect_gnss_synchro_monitor();
00186         int connect_acquisition_monitor();
00187         int connect_tracking_monitor();
00188         int connect_navdata_monitor();
00189
00190     #if ENABLE_FPGA
00191         int connect_fpga_flowgraph();
00192         int connect_fpga_sample_counter();
00193     #endif
00194
00195         int assign_channels();
00196         void check_signal_conditioners();
00197
00198         void set_signals_list();
00199         void set_channels_state(); // Initializes the channels state (start acquisition or keep standby)
00200                                   // using the configuration parameters (number of channels and max
00201     channels in acquisition)
00202     Gnss_Signal search_next_signal(const std::string& searched_signal,
00203         bool& is_primary_frequency,
00204         bool& assistance_available,
00205         float& estimated_doppler,
00206         double& RX_time);
00207
00208     void push_back_signal(const Gnss_Signal& gs);
00209     void remove_signal(const Gnss_Signal& gs);
00210     void print_help();
00211     void check_desktop_conf_in_fpga_env();
00212
00213     double project_doppler(const std::string& searched_signal, double primary_freq_doppler_hz);
00214     bool is_multiband() const;

```

```

00214
00215     std::vector<std::string> split_string(const std::string& s, char delim);
00216     std::vector<bool> signal_conditioner_connected_;
00217
00218     gr::top_block_sptr top_block_;
00219
00220     std::shared_ptr<ConfigurationInterface> configuration_;
00221     std::shared_ptr<Concurrent_Queue<pmt::pmt_t> queue_;
00222
00223     std::vector<std::shared_ptr<SignalSourceInterface> sig_source_;
00224     std::vector<std::shared_ptr<GNSSBlockInterface> sig_conditioner_;
00225     std::vector<std::shared_ptr<ChannelInterface> channels_;
00226     std::shared_ptr<GNSSBlockInterface> observables_;
00227     std::shared_ptr<GNSSBlockInterface> pvt_;
00228
00229     std::map<std::string, gr::basic_block_sptr> acq_resamplers_;
00230     std::vector<gr::blocks::null_sink::sptr> null_sinks_;
00231
00232     gr::basic_block_sptr GnssSynchroMonitor_;
00233     gr::basic_block_sptr GnssSynchroAcquisitionMonitor_;
00234     gr::basic_block_sptr GnssSynchroTrackingMonitor_;
00235     gr::basic_block_sptr NavDataMonitor_;
00236     channel_status_msg_receiver_sptr channels_status_; // class that receives and stores the current
status of the receiver channels
00237     galileo_e6_has_msg_receiver_sptr gal_e6_has_rx_;
00238     galileo_tow_map_sptr galileo_tow_map_;
00239     osnma_msg_receiver_sptr osnma_rx_;
00240
00241     gnss_sdr_sample_counter_sptr ch_out_sample_counter_;
00242     #if ENABLE_FPGA
00243     gnss_sdr_fpga_sample_counter_sptr ch_out_fpga_sample_counter_;
00244     #endif
00245
00246     std::vector<unsigned int> channels_state_;
00247
00248     std::list<Gnss_Signal> available_GPS_1C_signals_;
00249     std::list<Gnss_Signal> available_GPS_2S_signals_;
00250     std::list<Gnss_Signal> available_GPS_L5_signals_;
00251     std::list<Gnss_Signal> available_SBAS_1C_signals_;
00252     std::list<Gnss_Signal> available_GAL_1B_signals_;
00253     std::list<Gnss_Signal> available_GAL_5X_signals_;
00254     std::list<Gnss_Signal> available_GAL_7X_signals_;
00255     std::list<Gnss_Signal> available_GAL_E6_signals_;
00256     std::list<Gnss_Signal> available_GLO_1G_signals_;
00257     std::list<Gnss_Signal> available_GLO_2G_signals_;
00258     std::list<Gnss_Signal> available_BDS_B1_signals_;
00259     std::list<Gnss_Signal> available_BDS_B3_signals_;
00260
00261     enum StringValue
00262     {
00263         evGPS_1C,
00264         evGPS_2S,
00265         evGPS_L5,
00266         evSBAS_1C,
00267         evGAL_1B,
00268         evGAL_5X,
00269         evGAL_7X,
00270         evGAL_E6,
00271         evGLO_1G,
00272         evGLO_2G,
00273         evBDS_B1,
00274         evBDS_B3
00275     };
00276     std::map<std::string, StringValue> mapStringValue_;
00277
00278     std::string config_file_;
00279     std::string help_hint_;
00280
00281     std::mutex signal_list_mutex_;
00282
00283     int sources_count_;
00284     int channels_count_;
00285     int acq_channels_count_;
00286     int max_acq_channels_;
00287
00288     bool connected_;
00289     bool running_;
00290     bool multiband_;
00291     bool enable_monitor_;
00292     bool enable_acquisition_monitor_;
00293     bool enable_tracking_monitor_;
00294     bool enable_navdata_monitor_;
00295     bool enable_fpga_offloading_;
00296     bool enable_osnma_rx_;
00297     bool enable_e6_has_rx_;
00298 };
00299

```



```

00300
00301 /** \} */
00302 /** \} */
00303 #endif // GNSS_SDR_GNSS_FLOWGRAPH_H

```

## 11.665 in\_memory\_configuration.h File Reference

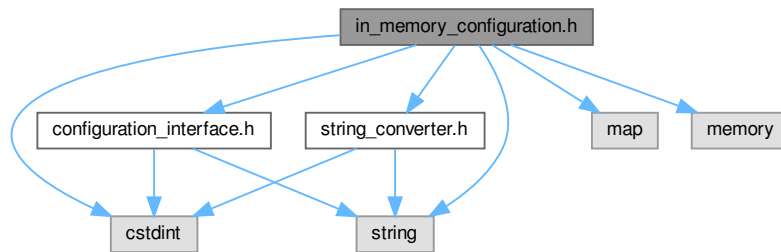
A [ConfigurationInterface](#) for testing purposes.

```

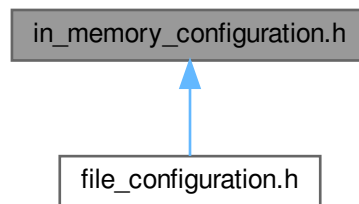
#include "configuration_interface.h"
#include "string_converter.h"
#include <cstdint>
#include <map>
#include <memory>
#include <string>

```

Include dependency graph for in\_memory\_configuration.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [InMemoryConfiguration](#)

*This class is an implementation of the interface [ConfigurationInterface](#).*

### 11.665.1 Detailed Description

A [ConfigurationInterface](#) for testing purposes.

**Author**

Carlos Aviles, 2010. carlos.avilesr(at)gmail.com

This implementation accepts configuration parameters upon instantiation and it is intended to be used in unit testing. GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [in\\_memory\\_configuration.h](#).

**11.666 in\_memory\_configuration.h**

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file in_memory_configuration.h
00003  * \brief A ConfigurationInterface for testing purposes.
00004  * \author Carlos Aviles, 2010. carlos.avilesr(at)gmail.com
00005  *
00006  * This implementation accepts configuration parameters upon instantiation and
00007  * it is intended to be used in unit testing.
00008  *
00009  * -----
00010  *
00011  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00012  * This file is part of GNSS-SDR.
00013  *
00014  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00015  * SPDX-License-Identifier: GPL-3.0-or-later
00016  *
00017  * -----
00018  */
00019
00020
00021 #ifndef GNSS_SDR_IN_MEMORY_CONFIGURATION_H
00022 #define GNSS_SDR_IN_MEMORY_CONFIGURATION_H
00023
00024 #include "configuration_interface.h"
00025 #include "string_converter.h"
00026 #include <stdint>
00027 #include <map>
00028 #include <memory>
00029 #include <string>
00030
00031 /** \addtogroup Core
00032  * \{ */
00033 /** \addtogroup Core_Receiver
00034  * \{ */
00035
00036
00037 /*!
00038  * \brief This class is an implementation of the interface ConfigurationInterface.
00039  *
00040  * This implementation accepts configuration parameters upon instantiation and
00041  * it is intended to be used in unit testing.
00042  */
00043 class InMemoryConfiguration : public ConfigurationInterface
00044 {
00045 public:
00046     InMemoryConfiguration();
00047     ~InMemoryConfiguration();
00048     std::string property(std::string property_name, std::string default_value) const override;
00049     bool property(std::string property_name, bool default_value) const override;
00050     int64_t property(std::string property_name, int64_t default_value) const override;
00051     uint64_t property(std::string property_name, uint64_t default_value) const override;
00052     int32_t property(std::string property_name, int32_t default_value) const override;
00053     uint32_t property(std::string property_name, uint32_t default_value) const override;
00054     int16_t property(std::string property_name, int16_t default_value) const override;
00055     uint16_t property(std::string property_name, uint16_t default_value) const override;
00056     float property(std::string property_name, float default_value) const override;
00057     double property(std::string property_name, double default_value) const override;
00058     void set_property(std::string property_name, std::string value) override;
00059     void supersede_property(const std::string& property_name, const std::string& value);
00060     bool is_present(const std::string& property_name) const;
00061
00062 private:
00063     std::map<std::string, std::string> properties_;
00064     std::unique_ptr<StringConverter> converter_;
00065 };
00066
00067
00068 /** \} */
00069 /** \} */
00070 #endif // GNSS_SDR_IN_MEMORY_CONFIGURATION_H

```

## 11.667 tcp\_cmd\_interface.h File Reference

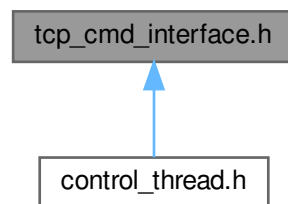
Class that implements a TCP/IP telecommand command line interface for GNSS-SDR.

```
#include "concurrent_queue.h"
#include <pmt/pmt.h>
#include <array>
#include <cstdint>
#include <ctime>
#include <functional>
#include <memory>
#include <string>
#include <unordered_map>
#include <vector>
```

Include dependency graph for tcp\_cmd\_interface.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [TcpCmdInterface](#)

### 11.667.1 Detailed Description

Class that implements a TCP/IP telecommand command line interface for GNSS-SDR.

#### Author

Javier Arribas jarribas (at) ctte.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [tcp\\_cmd\\_interface.h](#).

---

## 11.668 tcp\_cmd\_interface.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file tcp_cmd_interface.h
00003  *
00004  * \brief Class that implements a TCP/IP telecommand command line interface
00005  * for GNSS-SDR
00006  * \author Javier Arribas jarribas (at) cttc.es
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018  #ifndef GNSS_SDR_TCP_CMD_INTERFACE_H
00019  #define GNSS_SDR_TCP_CMD_INTERFACE_H
00020
00021
00022  #include "concurrent_queue.h"
00023  #include <pmt/pmt.h>
00024  #include <array>
00025  #include <cstdint>
00026  #include <ctime>
00027  #include <functional>
00028  #include <memory>
00029  #include <string>
00030  #include <unordered_map>
00031  #include <vector>
00032
00033  /** \addtogroup Core
00034   * \{ */
00035  /** \addtogroup Core_Receiver
00036   * \{ */
00037
00038
00039  class PvtInterface;
00040
00041  class TcpCmdInterface
00042  {
00043  public:
00044      TcpCmdInterface();
00045      ~TcpCmdInterface() = default;
00046      void run_cmd_server(int tcp_port);
00047      void set_msg_queue(std::shared_ptr<Concurrent_Queue<pmt::pmt_t>> control_queue);
00048
00049      /*!
00050       * \brief gets the UTC time parsed from the last TC command issued
00051       */
00052      time_t get_utc_time() const;
00053
00054      /*!
00055       * \brief gets the Latitude, Longitude and Altitude vector from the last TC command issued
00056       */
00057      std::array<float, 3> get_LLH() const;
00058
00059      void set_pvt(std::shared_ptr<PvtInterface> PVT_sptr);
00060
00061  private:
00062      std::unordered_map<std::string, std::function<std::string(const std::vector<std::string> &)>>
00063          functions_;
00064      std::string status(const std::vector<std::string> &commandLine);
00065      std::string reset(const std::vector<std::string> &commandLine);
00066      std::string standby(const std::vector<std::string> &commandLine);
00067      std::string hotstart(const std::vector<std::string> &commandLine);
00068      std::string warmstart(const std::vector<std::string> &commandLine);
00069      std::string coldstart(const std::vector<std::string> &commandLine);
00070      std::string set_ch_satellite(const std::vector<std::string> &commandLine);
00071
00072      void register_functions();
00073
00074      std::shared_ptr<Concurrent_Queue<pmt::pmt_t>> control_queue_;
00075      std::shared_ptr<PvtInterface> PVT_sptr_;
00076
00077      float rx_latitude_;
00078      float rx_longitude_;
00079      float rx_altitude_;
00080
00081      time_t receiver_utc_time_;
00082
00083      bool keep_running_;
00084  };
00085
00086
00087  /** \} */

```

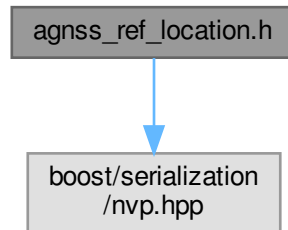
```
00088 /** \} */
00089 #endif // GNSS_SDR_TCP_CMD_INTERFACE_H
```

## 11.669 agnss\_ref\_location.h File Reference

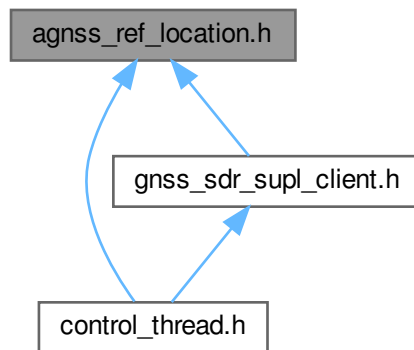
Interface of an Assisted GNSS REFERENCE LOCATION storage.

```
#include <boost/serialization/nvp.hpp>
```

Include dependency graph for agnss\_ref\_location.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [Agnss\\_Ref\\_Location](#)

*Interface of an Assisted GNSS REFERENCE LOCATION storage.*

### 11.669.1 Detailed Description

Interface of an Assisted GNSS REFERENCE LOCATION storage.

## Author

Javier Arribas, 2013. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [agnss\\_ref\\_location.h](#).

---

## 11.670 agnss\_ref\_location.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file agnss_ref_location.h
00003  * \brief Interface of an Assisted GNSS REFERENCE LOCATION storage
00004  * \author Javier Arribas, 2013. jarribas(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_AGNSS_REF_LOCATION_H
00019 #define GNSS_SDR_AGNSS_REF_LOCATION_H
00020
00021 #include <boost/serialization/nvp.hpp>
00022
00023 /** \addtogroup Core
00024  * \{ */
00025 /** \addtogroup System_Parameters
00026  * Classes containing info about system parameters for the different GNSS.
00027  * \{ */
00028
00029 /*!
00030 * \brief Interface of an Assisted GNSS REFERENCE LOCATION storage
00031 *
00032 */
00033 class Agnss_Ref_Location
00034 {
00035 public:
00036     /*!
00037     * Default constructor
00038     */
00039     Agnss_Ref_Location() = default;
00040
00041     double lat{};
00042     double lon{};
00043     double uncertainty{};
00044     bool valid{};
00045
00046     template <class Archive>
00047
00048     /*!
00049     * \brief Serialize is a boost standard method to be called by the boost XML
00050     * serialization. Here is used to save the Ref location on disk file.
00051     */
00052     inline void serialize(Archive& archive, const unsigned int version)
00053     {
00054         using boost::serialization::make_nvp;
00055         if (version)
00056         {
00057         };
00058         archive& BOOST_SERIALIZATION_NVP(lat);
00059         archive& BOOST_SERIALIZATION_NVP(lon);
00060         archive& BOOST_SERIALIZATION_NVP(uncertainty);
00061         archive& BOOST_SERIALIZATION_NVP(valid);
00062     }
00063 };
00064
00065
00066 /** \} */
00067 /** \} */
00068 #endif // GNSS_SDR_AGNSS_REF_LOCATION_H

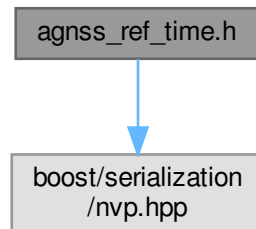
```

## 11.671 agnss\_ref\_time.h File Reference

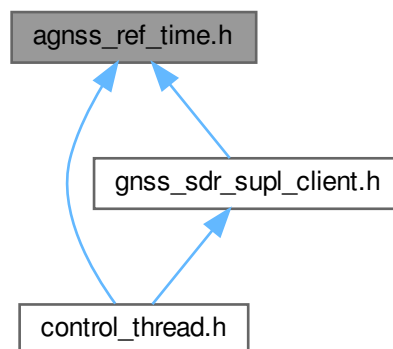
Interface of an Assisted GNSS REFERENCE TIME storage.

```
#include <boost/serialization/nvp.hpp>
```

Include dependency graph for agnss\_ref\_time.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [Agnss\\_Ref\\_Time](#)

*Interface of an Assisted GNSS REFERENCE TIME storage.*

### 11.671.1 Detailed Description

Interface of an Assisted GNSS REFERENCE TIME storage.

Author

Javier Arribas, 2013. [jarribas\(at\)cttc.es](mailto:jarribas(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [agnss\\_ref\\_time.h](#).

---

## 11.672 agnss\_ref\_time.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file agnss_ref_time.h
00003  * \brief Interface of an Assisted GNSS REFERENCE TIME storage
00004  * \author Javier Arribas, 2013. jarribas(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_AGNSS_REF_TIME_H
00019 #define GNSS_SDR_AGNSS_REF_TIME_H
00020
00021 #include <boost/serialization/nvp.hpp>
00022
00023 /** \addtogroup Core
00024  * \{ */
00025 /** \addtogroup System_Parameters
00026  * \{ */
00027
00028
00029 /*!
00030 * \brief Interface of an Assisted GNSS REFERENCE TIME storage
00031 *
00032 */
00033 class Agnss_Ref_Time
00034 {
00035 public:
00036     /*!
00037      * Default constructor
00038      */
00039     Agnss_Ref_Time() = default;
00040
00041     double tow{};
00042     double week{};
00043     double seconds{};
00044     double microseconds{};
00045     bool valid{};
00046
00047     template <class Archive>
00048
00049     /*!
00050      * \brief Serialize is a boost standard method to be called by the boost XML
00051      * serialization. Here is used to save the ref time data on disk file.
00052      */
00053     inline void serialize(Archive& archive, const unsigned int version)
00054     {
00055         if (version)
00056         {
00057         };
00058         archive& BOOST_SERIALIZATION_NVP(tow);
00059         archive& BOOST_SERIALIZATION_NVP(week);
00060         archive& BOOST_SERIALIZATION_NVP(seconds);
00061         archive& BOOST_SERIALIZATION_NVP(microseconds);
00062         archive& BOOST_SERIALIZATION_NVP(valid);
00063     }
00064 };
00065
00066
00067 /** \} */
00068 /** \} */
00069 #endif // GNSS_SDR_AGNSS_REF_TIME_H

```

## 11.673 Beidou\_B1I.h File Reference

Defines system parameters for BeiDou B1I signal and DNAV data.

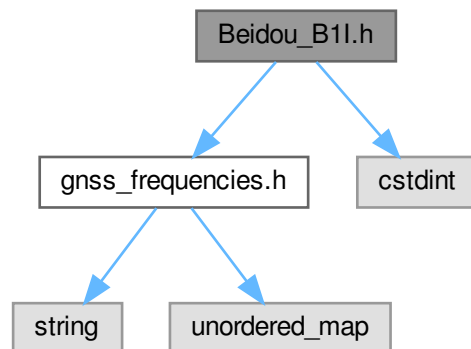
```

#include "gnss_frequencies.h"
#include <cstdint>

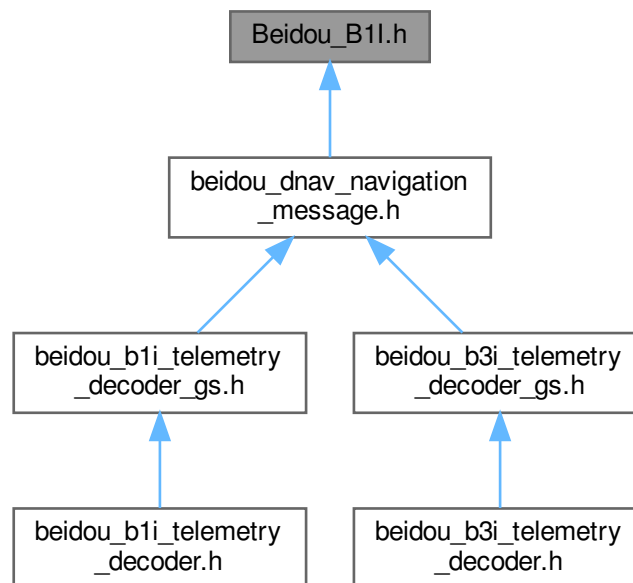
```



Include dependency graph for Beidou\_B1I.h:



This graph shows which files directly or indirectly include this file:



## Variables

- constexpr double `BEIDOU_B1I_FREQ_HZ` = `FREQ1_BDS`  
*B1I [Hz].*
- constexpr double `BEIDOU_B1I_CODE_RATE_CPS` = 2.046e6  
*Beidou B1I code rate [chips/s].*
- constexpr double `BEIDOU_B1I_CODE_LENGTH_CHIPS` = 2046.0  
*Beidou B1I code length [chips].*

- constexpr double `BEIDOU_B1I_CODE_PERIOD_S` = 0.001  
*Beidou B1I code period [seconds].*
- constexpr double `BEIDOU_B1I_PREAMBLE_DURATION_S` = 0.220
- constexpr uint32\_t `BEIDOU_B1I_CODE_PERIOD_MS` = 1  
*Beidou B1I code period [ms].*
- constexpr uint32\_t `BEIDOU_B1I_PREAMBLE_LENGTH_BITS` = 11
- constexpr uint32\_t `BEIDOU_B1I_PREAMBLE_LENGTH_SYMBOLS` = 220
- constexpr int32\_t `BEIDOU_B1I_SECONDARY_CODE_LENGTH` = 20
- constexpr int32\_t `BEIDOU_B1I_GEO_PREAMBLE_LENGTH_SYMBOLS` = 22
- constexpr int32\_t `BEIDOU_B1I_PREAMBLE_DURATION_MS` = 220
- constexpr int32\_t `BEIDOU_B1I_TELEMETRY_RATE_BITS_SECOND` = 50
- constexpr int32\_t `BEIDOU_B1I_TELEMETRY_SYMBOLS_PER_BIT` = 20
- constexpr int32\_t `BEIDOU_B1I_GEO_TELEMETRY_SYMBOLS_PER_BIT` = 2
- constexpr int32\_t `BEIDOU_B1I_TELEMETRY_SYMBOL_PERIOD_MS` = static\_cast<int32\_t>(static\_cast<uint32\_t>(BEIDOU\_B1I\_TELEMETRY\_SYMBOLS\_PER\_BIT) \* `BEIDOU_B1I_CODE_PERIOD_MS`)
- constexpr int32\_t `BEIDOU_B1I_TELEMETRY_RATE_SYMBOLS_SECOND` = BEIDOU\_B1I\_TELEMETRY\_RATE\_BITS\_SECOND \* BEIDOU\_B1I\_TELEMETRY\_SYMBOLS\_PER\_BIT
- constexpr char `BEIDOU_B1I_SECONDARY_CODE_STR` [21] = "00000100110101001110"
- constexpr char `BEIDOU_B1I_GEO_PREAMBLE_SYMBOLS_STR` [23] = "1111110000001100001100"
- constexpr char `BEIDOU_B1I_D2_SECONDARY_CODE_STR` [3] = "00"

### 11.673.1 Detailed Description

Defines system parameters for BeiDou B1I signal and DNAV data.

#### Author

Sergi Segura, 2018. [sergi.segura.munoz\(at\)gmail.com](mailto:sergi.segura.munoz(at)gmail.com)

Damian Miralles, 2018. [dmiralles2009@gmail.com](mailto:dmiralles2009@gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [Beidou\\_B1I.h](#).

---

## 11.674 Beidou\_B1I.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file Beidou_B1I.h
00003  * \brief Defines system parameters for BeiDou B1I signal and DNAV data
00004  * \author Sergi Segura, 2018. sergi.segura.munoz\(at\)gmail.com
00005  * \author Damian Miralles, 2018. dmiralles2009@gmail.com
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018
00019 #ifndef GNSS_SDR_BEIDOU_B1I_H
00020 #define GNSS_SDR_BEIDOU_B1I_H
00021
00022 #include "gnss_frequencies.h"
00023 #include <cstdint>
00024
00025 /** \addtogroup Core
00026  * \{ */
00027 /** \addtogroup System_Parameters
00028  * \{ */
00029
00030

```

```

00031 // carrier and code frequencies
00032 constexpr double BEIDOU_B1I_FREQ_HZ = FREQ1_BDS;          //!< B1I [Hz]
00033 constexpr double BEIDOU_B1I_CODE_RATE_CPS = 2.046e6;      //!< Beidou B1I code rate [chips/s]
00034 constexpr double BEIDOU_B1I_CODE_LENGTH_CHIPS = 2046.0;   //!< Beidou B1I code length [chips]
00035 constexpr double BEIDOU_B1I_CODE_PERIOD_S = 0.001;        //!< Beidou B1I code period [seconds]
00036 constexpr double BEIDOU_B1I_PREAMBLE_DURATION_S = 0.220;
00037 constexpr uint32_t BEIDOU_B1I_CODE_PERIOD_MS = 1;         //!< Beidou B1I code period [ms]
00038 constexpr uint32_t BEIDOU_B1I_PREAMBLE_LENGTH_BITS = 11;
00039 constexpr uint32_t BEIDOU_B1I_PREAMBLE_LENGTH_SYMBOLS = 220;
00040 constexpr int32_t BEIDOU_B1I_SECONDARY_CODE_LENGTH = 20;
00041 constexpr int32_t BEIDOU_B1I_GEO_PREAMBLE_LENGTH_SYMBOLS = 22;
00042 constexpr int32_t BEIDOU_B1I_PREAMBLE_DURATION_MS = 220;
00043 constexpr int32_t BEIDOU_B1I_TELEMETRY_RATE_BITS_SECOND = 50;
00044 constexpr int32_t BEIDOU_B1I_TELEMETRY_SYMBOLS_PER_BIT = 20;
00045 constexpr int32_t BEIDOU_B1I_GEO_TELEMETRY_SYMBOLS_PER_BIT = 2;
00046 constexpr int32_t BEIDOU_B1I_TELEMETRY_SYMBOL_PERIOD_MS =
    static_cast<int32_t>(static_cast<uint32_t>(BEIDOU_B1I_TELEMETRY_SYMBOLS_PER_BIT) *
    BEIDOU_B1I_CODE_PERIOD_MS);
00047 constexpr int32_t BEIDOU_B1I_TELEMETRY_RATE_SYMBOLS_SECOND = BEIDOU_B1I_TELEMETRY_RATE_BITS_SECOND *
    BEIDOU_B1I_TELEMETRY_SYMBOLS_PER_BIT;
00048 constexpr char BEIDOU_B1I_SECONDARY_CODE_STR[21] = "00000100110101001110";
00049 constexpr char BEIDOU_B1I_GEO_PREAMBLE_SYMBOLS_STR[23] = "1111110000001100001100";
00050 constexpr char BEIDOU_B1I_D2_SECONDARY_CODE_STR[3] = "00";
00051
00052
00053 /** \} */
00054 /** \} */
00055 #endif // GNSS_SDR_BEIDOU_B1I_H

```

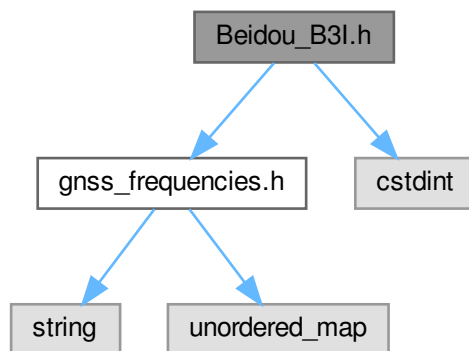
## 11.675 Beidou\_B3I.h File Reference

Defines system parameters for BeiDou B3I signal and DNAV data.

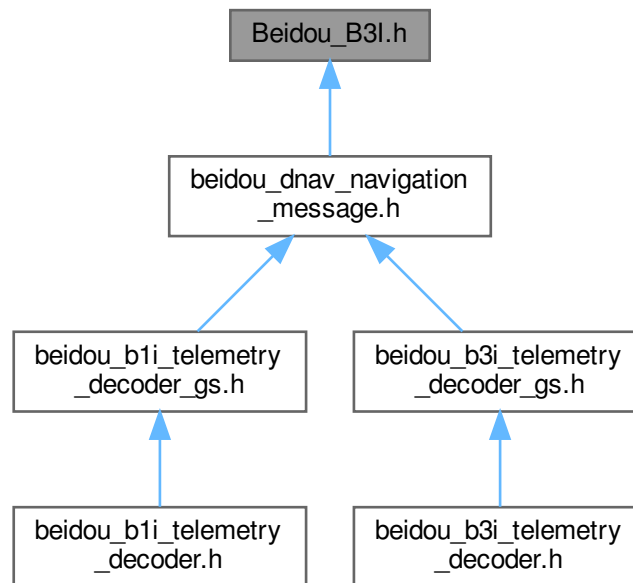
```
#include "gnss_frequencies.h"
```

```
#include <cstdint>
```

Include dependency graph for Beidou\_B3I.h:



This graph shows which files directly or indirectly include this file:



## Variables

- `constexpr double BEIDOU_B3I_FREQ_HZ = FREQ3_BDS`  
*BeiDou B3I [Hz].*
- `constexpr double BEIDOU_B3I_CODE_RATE_CPS = 10.23e6`  
*BeiDou B3I code rate [chips/s].*
- `constexpr double BEIDOU_B3I_CODE_LENGTH_CHIPS = 10230.0`  
*BeiDou B3I code length [chips].*
- `constexpr double BEIDOU_B3I_CODE_PERIOD_S = 0.001`  
*BeiDou B3I code period [seconds].*
- `constexpr double BEIDOU_B3I_PREAMBLE_DURATION_S = 0.220`
- `constexpr uint32_t BEIDOU_B3I_CODE_PERIOD_MS = 1`  
*BeiDou B3I code period [ms].*
- `constexpr uint32_t BEIDOU_B3I_PREAMBLE_LENGTH_BITS = 11`
- `constexpr uint32_t BEIDOU_B3I_PREAMBLE_LENGTH_SYMBOLS = 220`
- `constexpr int32_t BEIDOU_B3I_SECONDARY_CODE_LENGTH = 20`
- `constexpr int32_t BEIDOU_B3I_GEO_PREAMBLE_LENGTH_SYMBOLS = 22`
- `constexpr int32_t BEIDOU_B3I_PREAMBLE_DURATION_MS = 220`
- `constexpr int32_t BEIDOU_B3I_TELEMETRY_RATE_BITS_SECOND = 50`  
*D1 NAV message bit rate [bits/s].*
- `constexpr int32_t BEIDOU_B3I_TELEMETRY_SYMBOLS_PER_BIT = 20`
- `constexpr int32_t BEIDOU_B3I_GEO_TELEMETRY_SYMBOLS_PER_BIT = 2`
- `constexpr int32_t BEIDOU_B3I_TELEMETRY_SYMBOL_PERIOD_MS = static_cast<int32_t>(static_cast<uint32_t>(BEIDOU_B3I_TELEMETRY_SYMBOLS_PER_BIT) * BEIDOU_B3I_CODE_PERIOD_MS)`
- `constexpr int32_t BEIDOU_B3I_TELEMETRY_RATE_SYMBOLS_SECOND = BEIDOU_B3I_TELEMETRY_RATE_BITS_SECOND * BEIDOU_B3I_TELEMETRY_SYMBOLS_PER_BIT`
- `constexpr char BEIDOU_B3I_SECONDARY_CODE_STR [21] = "00000100110101001110"`
- `constexpr char BEIDOU_B3I_GEO_PREAMBLE_SYMBOLS_STR [23] = "1111110000001100001100"`
- `constexpr char BEIDOU_B3I_D2_SECONDARY_CODE_STR [3] = "00"`

## 11.675.1 Detailed Description

Defines system parameters for BeiDou B3I signal and DNAV data.

Author

Damian Miralles, 2019. [dmiralles2009@gmail.com](mailto:dmiralles2009@gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [Beidou\\_B3I.h](#).

---

## 11.676 Beidou\_B3I.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file Beidou_B3I.h
00003  * \brief Defines system parameters for BeiDou B3I signal and DNAV data
00004  * \author Damian Miralles, 2019. dmiralles2009@gmail.com
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_BEIDOU_B3I_H
00019 #define GNSS_SDR_BEIDOU_B3I_H
00020
00021 #include "gnss_frequencies.h"
00022 #include <stdint>
00023
00024 /** \addtogroup Core
00025  * \{ */
00026 /** \addtogroup System_Parameters
00027  * \{ */
00028
00029
00030 // carrier and code frequencies
00031 constexpr double BEIDOU_B3I_FREQ_HZ = FREQ3_BDS;          //!< BeiDou B3I [Hz]
00032 constexpr double BEIDOU_B3I_CODE_RATE_CPS = 10.23e6;      //!< BeiDou B3I code rate [chips/s]
00033 constexpr double BEIDOU_B3I_CODE_LENGTH_CHIPS = 10230.0;  //!< BeiDou B3I code length [chips]
00034 constexpr double BEIDOU_B3I_CODE_PERIOD_S = 0.001;        //!< BeiDou B3I code period [seconds]
00035 constexpr double BEIDOU_B3I_PREAMBLE_DURATION_S = 0.220;
00036 constexpr uint32_t BEIDOU_B3I_CODE_PERIOD_MS = 1;          //!< BeiDou B3I code period [ms]
00037 constexpr uint32_t BEIDOU_B3I_PREAMBLE_LENGTH_BITS = 11;
00038 constexpr uint32_t BEIDOU_B3I_PREAMBLE_LENGTH_SYMBOLS = 220; // *****
00039 constexpr int32_t BEIDOU_B3I_SECONDARY_CODE_LENGTH = 20;
00040 constexpr int32_t BEIDOU_B3I_GEO_PREAMBLE_LENGTH_SYMBOLS = 22;
00041 constexpr int32_t BEIDOU_B3I_PREAMBLE_DURATION_MS = 220;
00042 constexpr int32_t BEIDOU_B3I_TELEMETRY_RATE_BITS_SECOND = 50; //!< D1 NAV message bit rate [bits/s]
00043 constexpr int32_t BEIDOU_B3I_TELEMETRY_SYMBOLS_PER_BIT = 20;
00044 constexpr int32_t BEIDOU_B3I_GEO_TELEMETRY_SYMBOLS_PER_BIT = 2; // *****
00045 constexpr int32_t BEIDOU_B3I_TELEMETRY_SYMBOL_PERIOD_MS =
    static_cast<int32_t>(static_cast<uint32_t>(BEIDOU_B3I_TELEMETRY_SYMBOLS_PER_BIT) *
    BEIDOU_B3I_CODE_PERIOD_MS);
00046 constexpr int32_t BEIDOU_B3I_TELEMETRY_RATE_SYMBOLS_SECOND = BEIDOU_B3I_TELEMETRY_RATE_BITS_SECOND *
    BEIDOU_B3I_TELEMETRY_SYMBOLS_PER_BIT;
00047 constexpr char BEIDOU_B3I_SECONDARY_CODE_STR[21] = "00000100110101001110";
00048 constexpr char BEIDOU_B3I_GEO_PREAMBLE_SYMBOLS_STR[23] = "1111110000001100001100";
00049 constexpr char BEIDOU_B3I_D2_SECONDARY_CODE_STR[3] = "00";
00050
00051
00052 /** \} */
00053 /** \} */
00054 #endif // GNSS_SDR_BEIDOU_B3I_H
```

## 11.677 Beidou\_DNAV.h File Reference

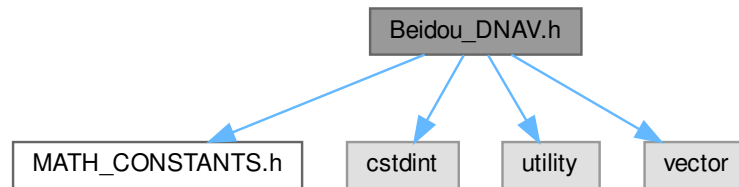
Defines system parameters for BeiDou DNAV data processing.

```
#include "MATH_CONSTANTS.h"
#include <stdint>
```

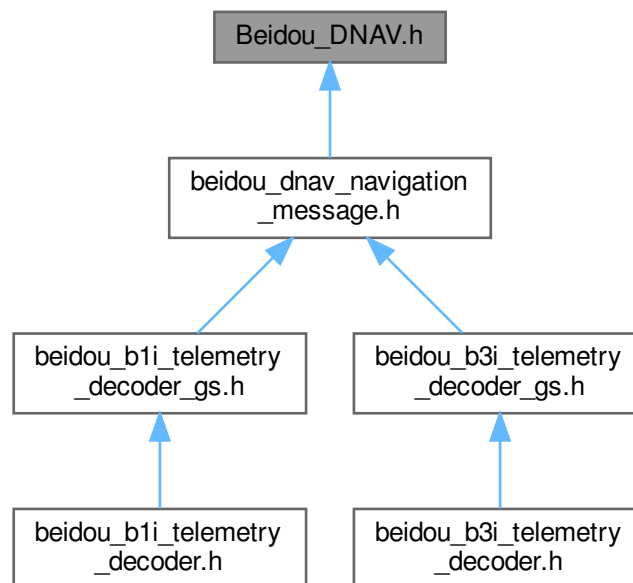
```
#include <utility>
```

```
#include <vector>
```

Include dependency graph for Beidou\_DNAV.h:



This graph shows which files directly or indirectly include this file:



## Functions

- `const std::vector< std::pair< int32_t, int32_t > > > D1_PRE {{{1, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > > D1_FRAID {{{16, 3}}}`
- `const std::vector< std::pair< int32_t, int32_t > > > D1_SOW {{{19, 8}, {31, 12}}}`
- `const std::vector< std::pair< int32_t, int32_t > > > D1_PNUM {{{44, 7}}}`
- `const std::vector< std::pair< int32_t, int32_t > > > D1_SAT_H1 {{{43, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > > D1_AODC {{{44, 5}}}`
- `const std::vector< std::pair< int32_t, int32_t > > > D1_URAI {{{49, 4}}}`
- `const std::vector< std::pair< int32_t, int32_t > > > D1_WN {{{61, 13}}}`
- `const std::vector< std::pair< int32_t, int32_t > > > D1_TOC {{{74, 9}, {91, 8}}}`

- `const std::vector< std::pair< int32_t, int32_t > > D1_TGD1 {{{99, 10}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_TGD2 {{{109, 4}, {121, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_ALPHA0 {{{127, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_ALPHA1 {{{135, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_ALPHA2 {{{151, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_ALPHA3 {{{159, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_BETA0 {{{167, 6}, {181, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_BETA1 {{{183, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_BETA2 {{{191, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_BETA3 {{{199, 4}, {211, 4}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_A2 {{{215, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_A0 {{{226, 7}, {241, 17}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_A1 {{{258, 5}, {271, 17}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_AODE {{{288, 5}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_DELTA_N {{{43, 10}, {61, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_CUC {{{67, 16}, {91, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_M0 {{{93, 20}, {121, 12}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_E {{{133, 10}, {151, 22}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_CUS {{{181, 18}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_CRC {{{199, 4}, {211, 14}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_CRS {{{225, 8}, {241, 10}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_SQRT_A {{{251, 12}, {271, 20}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_TOE_SF2 {{{291, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_TOE_SF3 {{{43, 10}, {61, 5}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_I0 {{{66, 17}, {91, 15}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_CIC {{{106, 7}, {121, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_OMEGA_DOT {{{132, 11}, {151, 13}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_CIS {{{164, 9}, {181, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_IDOT {{{190, 13}, {211, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_OMEGA0 {{{212, 21}, {241, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_OMEGA {{{252, 11}, {271, 21}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_SQRT_A_ALMANAC {{{51, 2}, {61, 22}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_A1_ALMANAC {{{91, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_A0_ALMANAC {{{102, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_OMEGA0_ALMANAC {{{121, 22}, {151, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_E_ALMANAC {{{153, 17}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_DELTA_I {{{170, 3}, {181, 13}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_TOA {{{194, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_OMEGA_DOT_ALMANAC {{{202, 1}, {211, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_OMEGA_ALMANAC {{{227, 6}, {241, 18}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_M0_ALMANAC {{{259, 4}, {271, 20}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA1 {{{51, 2}, {61, 7}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA2 {{{68, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA3 {{{77, 6}, {91, 3}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA4 {{{94, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA5 {{{103, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA6 {{{112, 1}, {121, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA7 {{{129, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA8 {{{138, 5}, {151, 4}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA9 {{{155, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA10 {{{164, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA11 {{{181, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA12 {{{190, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA13 {{{199, 4}, {211, 5}}}`
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA14 {{{216, 9}}}`

- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA15` {{{225, 8}, {241, 1}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA16` {{{242, 9}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA17` {{{251, 9}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA18` {{{260, 3}, {271, 6}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA19` {{{277, 9}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA20` {{{51, 2}, {61, 7}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA21` {{{68, 9}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA22` {{{77, 6}, {91, 3}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA23` {{{94, 9}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA24` {{{103, 9}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA25` {{{112, 1}, {121, 8}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA26` {{{129, 9}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA27` {{{138, 5}, {151, 4}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA28` {{{155, 9}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA29` {{{164, 9}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_HEA30` {{{181, 9}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_WNA` {{{190, 8}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_TOA2` {{{198, 5}, {211, 3}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_A0GPS` {{{97, 14}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_A1GPS` {{{111, 2}, {121, 14}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_A0GAL` {{{135, 8}, {151, 6}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_A1GAL` {{{157, 16}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_A0GLO` {{{181, 14}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_A1GLO` {{{195, 8}, {211, 8}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_DELTA_T_LS` {{{51, 2}, {61, 6}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_DELTA_T_LSF` {{{67, 8}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_WN_LSF` {{{75, 8}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_A0UTC` {{{91, 22}, {121, 10}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_A1UTC` {{{131, 12}, {151, 12}}}
- `const std::vector< std::pair< int32_t, int32_t > > D1_DN` {{{163, 8}}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_PRE` {{{1, 11}}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_FRAID` {{{16, 3}}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_SOW` {{{19, 8}, {31, 12}}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_PNUM` {{{43, 4}}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_SAT_H1` {{{47, 1}}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_AODC` {{{48, 5}}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_URAI` {{{61, 4}}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_WN` {{{65, 13}}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_TOC` {{{78, 5}, {91, 12}}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_TGD1` {{{103, 10}}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_TGD2` {{{121, 10}}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_ALPHA0` {{{47, 6}, {61, 2}}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_ALPHA1` {{{63, 8}}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_ALPHA2` {{{71, 8}}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_ALPHA3` {{{79, 4}, {91, 4}}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_BETA0` {{{95, 8}}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_BETA1` {{{103, 8}}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_BETA2` {{{111, 2}, {121, 6}}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_BETA3` {{{127, 8}}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_A0` {{{101, 12}, {121, 12}}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_A1_MSB` {{{133, 4}}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_A1_LSB` {{{47, 6}, {61, 12}}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_A1` {{{279, 22}}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_A2` {{{73, 10}, {91, 1}}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_AODE` {{{92, 5}}}



- `const std::vector< std::pair< int32_t, int32_t > > D2_DELTA_N` ({97, 16}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_CUC_MSB` ({121, 14}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_CUC_LSB` ({47, 4}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_CUC` ({283, 18}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_M0` ({51, 2}, {61, 22}, {91, 8}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_CUS` ({99, 14}, {121, 4}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_E_MSB` ({125, 10}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_E_LSB` ({47, 6}, {61, 16}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_SQRT_A` ({77, 6}, {91, 22}, {121, 4}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_CIC_MSB` ({125, 10}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_CIC_LSB` ({47, 6}, {61, 2}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_CIC` ({283, 18}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_CIS` ({63, 18}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_TOE` ({81, 2}, {91, 15}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_I0_MSB` ({106, 7}, {121, 14}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_I0_LSB` ({47, 6}, {61, 5}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_I0` ({269, 32}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_CRC` ({66, 17}, {91, 1}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_CRS` ({92, 18}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_OMEGA_DOT_MSB` ({110, 3}, {121, 16}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_OMEGA_DOT_LSB` ({47, 5}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_OMEGA_DOT` ({277, 24}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_OMEGA0` ({52, 1}, {61, 22}, {91, 9}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_OMEGA_MSB` ({100, 13}, {121, 14}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_OMEGA_LSB` ({47, 5}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_OMEGA` ({269, 32}}
- `const std::vector< std::pair< int32_t, int32_t > > D2_IDOT` ({52, 1}, {61, 13}}

## Variables

- `constexpr double D1_TOC_LSB` = TWO\_P3
- `constexpr double D1_TGD1_LSB` = 0.1e-9
- `constexpr double D1_TGD2_LSB` = 0.1e-9
- `constexpr double D1_ALPHA0_LSB` = TWO\_N30
- `constexpr double D1_ALPHA1_LSB` = TWO\_N27
- `constexpr double D1_ALPHA2_LSB` = TWO\_N24
- `constexpr double D1_ALPHA3_LSB` = TWO\_N24
- `constexpr double D1_BETA0_LSB` = TWO\_P11
- `constexpr double D1_BETA1_LSB` = TWO\_P14
- `constexpr double D1_BETA2_LSB` = TWO\_P16
- `constexpr double D1_BETA3_LSB` = TWO\_P16
- `constexpr double D1_A2_LSB` = TWO\_N66
- `constexpr double D1_A0_LSB` = TWO\_N33
- `constexpr double D1_A1_LSB` = TWO\_N50
- `constexpr double D1_DELTA_N_LSB` = PI\_TWO\_N43
- `constexpr double D1_CUC_LSB` = TWO\_N31
- `constexpr double D1_M0_LSB` = PI\_TWO\_N31
- `constexpr double D1_E_LSB` = TWO\_N33
- `constexpr double D1_CUS_LSB` = TWO\_N31
- `constexpr double D1_CRC_LSB` = TWO\_N6
- `constexpr double D1_CRS_LSB` = TWO\_N6
- `constexpr double D1_SQRT_A_LSB` = TWO\_N19
- `constexpr double D1_TOE_LSB` = TWO\_P3
- `constexpr double D1_I0_LSB` = PI\_TWO\_N31
- `constexpr double D1_CIC_LSB` = TWO\_N31

- constexpr double `D1_OMEGA_DOT_LSB` = `PI_TWO_N43`
  - constexpr double `D1_CIS_LSB` = `TWO_N31`
  - constexpr double `D1_IDOT_LSB` = `PI_TWO_N43`
  - constexpr double `D1_OMEGA0_LSB` = `PI_TWO_N31`
  - constexpr double `D1_OMEGA_LSB` = `PI_TWO_N31`
  - constexpr double `D1_SQRT_A_ALMANAC_LSB` = `TWO_N11`
  - constexpr double `D1_A1_ALMANAC_LSB` = `TWO_N38`
  - constexpr double `D1_A0_ALMANAC_LSB` = `TWO_N20`
  - constexpr double `D1_OMEGA0_ALMANAC_LSB` = `PI_TWO_N23`
  - constexpr double `D1_E_ALMANAC_LSB` = `TWO_N21`
  - constexpr double `D1_DELTA_I_LSB` = `PI_TWO_N19`
  - constexpr double `D1_TOA_LSB` = `TWO_P12`
  - constexpr double `D1_OMEGA_DOT_ALMANAC_LSB` = `PI_TWO_N38`
  - constexpr double `D1_OMEGA_ALMANAC_LSB` = `PI_TWO_N23`
  - constexpr double `D1_M0_ALMANAC_LSB` = `PI_TWO_N23`
  - constexpr double `D1_A0GPS_LSB` = `0.1e-9`
  - constexpr double `D1_A1GPS_LSB` = `0.1e-9`
  - constexpr double `D1_A0GAL_LSB` = `0.1e-9`
  - constexpr double `D1_A1GAL_LSB` = `0.1e-9`
  - constexpr double `D1_A0GLO_LSB` = `0.1e-9`
  - constexpr double `D1_A1GLO_LSB` = `0.1e-9`
  - constexpr double `D1_A0UTC_LSB` = `TWO_N30`
  - constexpr double `D1_A1UTC_LSB` = `TWO_N50`
  - constexpr int32\_t `BEIDOU_DNAV_PREAMBLE_LENGTH_BITS` = 11
  - constexpr int32\_t `BEIDOU_DNAV_PREAMBLE_LENGTH_SYMBOLS` = 11
  - constexpr int32\_t `BEIDOU_DNAV_PREAMBLE_PERIOD_SYMBOLS` = 300
  - constexpr int32\_t `BEIDOU_DNAV_SUBFRAME_DATA_BITS` = 300
- Number of bits per subframe in the NAV message [bits].*
- constexpr int32\_t `BEIDOU_DNAV_BDT2GPST_LEAP_SEC_OFFSET` = 14
  - constexpr int32\_t `BEIDOU_DNAV_BDT2GPST_WEEK_NUM_OFFSET` = 1356
  - constexpr uint32\_t `BEIDOU_DNAV_SUBFRAME_SYMBOLS` = 300
  - constexpr uint32\_t `BEIDOU_DNAV_WORDS_SUBFRAME` = 10
  - constexpr uint32\_t `BEIDOU_DNAV_WORD_LENGTH_BITS` = 30
  - constexpr char `BEIDOU_DNAV_PREAMBLE` [12] = "11100010010"

### 11.677.1 Detailed Description

Defines system parameters for BeiDou DNAV data processing.

#### Author

Damian Miralles, 2018. [dmiralles2009@gmail.com](mailto:dmiralles2009@gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [Beidou\\_DNAV.h](#).

---

## 11.678 Beidou\_DNAV.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file Beidou_DNAV.h
00003  * \brief Defines system parameters for BeiDou DNAV data processing
00004  * \author Damian Miralles, 2018. dmiralles2009@gmail.com
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
```

```

00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_BEIDOU_DNAV_H
00019 #define GNSS_SDR_BEIDOU_DNAV_H
00020
00021 #include "MATH_CONSTANTS.h"
00022 #include <stdint>
00023 #include <utility>
00024 #include <vector>
00025
00026 /** \addtogroup Core
00027  * \{ */
00028 /** \addtogroup System_Parameters
00029  * \{ */
00030
00031
00032 // DNAV SCALE FACTORS
00033 // EPH
00034 constexpr double D1_TOC_LSB = TWO_P3;
00035 constexpr double D1_TGD1_LSB = 0.1e-9;
00036 constexpr double D1_TGD2_LSB = 0.1e-9;
00037 constexpr double D1_ALPHA0_LSB = TWO_N30;
00038 constexpr double D1_ALPHA1_LSB = TWO_N27;
00039 constexpr double D1_ALPHA2_LSB = TWO_N24;
00040 constexpr double D1_ALPHA3_LSB = TWO_N24;
00041 constexpr double D1_BETA0_LSB = TWO_P11;
00042 constexpr double D1_BETA1_LSB = TWO_P14;
00043 constexpr double D1_BETA2_LSB = TWO_P16;
00044 constexpr double D1_BETA3_LSB = TWO_P16;
00045 constexpr double D1_A2_LSB = TWO_N66;
00046 constexpr double D1_A0_LSB = TWO_N33;
00047 constexpr double D1_A1_LSB = TWO_N50;
00048 constexpr double D1_DELTA_N_LSB = PI_TWO_N43;
00049 constexpr double D1_CUC_LSB = TWO_N31;
00050 constexpr double D1_M0_LSB = PI_TWO_N31;
00051 constexpr double D1_E_LSB = TWO_N33;
00052 constexpr double D1_CUS_LSB = TWO_N31;
00053 constexpr double D1_CRC_LSB = TWO_N6;
00054 constexpr double D1_CRS_LSB = TWO_N6;
00055 constexpr double D1_SQRT_A_LSB = TWO_N19;
00056 constexpr double D1_TOE_LSB = TWO_P3;
00057 constexpr double D1_I0_LSB = PI_TWO_N31;
00058 constexpr double D1_CIC_LSB = TWO_N31;
00059 constexpr double D1_OMEGA_DOT_LSB = PI_TWO_N43;
00060 constexpr double D1_CIS_LSB = TWO_N31;
00061 constexpr double D1_IDOT_LSB = PI_TWO_N43;
00062 constexpr double D1_OMEGA0_LSB = PI_TWO_N31;
00063 constexpr double D1_OMEGA_LSB = PI_TWO_N31;
00064 // ALM
00065 constexpr double D1_SQRT_A_ALMANAC_LSB = TWO_N11;
00066 constexpr double D1_A1_ALMANAC_LSB = TWO_N38;
00067 constexpr double D1_A0_ALMANAC_LSB = TWO_N20;
00068 constexpr double D1_OMEGA0_ALMANAC_LSB = PI_TWO_N23;
00069 constexpr double D1_E_ALMANAC_LSB = TWO_N21;
00070 constexpr double D1_DELTA_I_LSB = PI_TWO_N19;
00071 constexpr double D1_TOA_LSB = TWO_P12;
00072 constexpr double D1_OMEGA_DOT_ALMANAC_LSB = PI_TWO_N38;
00073 constexpr double D1_OMEGA_ALMANAC_LSB = PI_TWO_N23;
00074 constexpr double D1_M0_ALMANAC_LSB = PI_TWO_N23;
00075 constexpr double D1_A0GPS_LSB = 0.1e-9;
00076 constexpr double D1_A1GPS_LSB = 0.1e-9;
00077 constexpr double D1_A0GAL_LSB = 0.1e-9;
00078 constexpr double D1_A1GAL_LSB = 0.1e-9;
00079 constexpr double D1_A0GLO_LSB = 0.1e-9;
00080 constexpr double D1_A1GLO_LSB = 0.1e-9;
00081 constexpr double D1_A0UTC_LSB = TWO_N30;
00082 constexpr double D1_A1UTC_LSB = TWO_N50;
00083
00084 constexpr int32_t BEIDOU_DNAV_PREAMBLE_LENGTH_BITS = 11;
00085 constexpr int32_t BEIDOU_DNAV_PREAMBLE_LENGTH_SYMBOLS = 11; // *****
00086 constexpr int32_t BEIDOU_DNAV_PREAMBLE_PERIOD_SYMBOLS = 300;
00087 constexpr int32_t BEIDOU_DNAV_SUBFRAME_DATA_BITS = 300; //!< Number of bits per subframe in the NAV
    message [bits]
00088 // Number of leap seconds passed from the start of the GPS epoch up to the start of BeiDou epoch
00089 constexpr int32_t BEIDOU_DNAV_BDT2GPST_LEAP_SEC_OFFSET = 14;
00090 // Number of weeks passed from the start of the GPS epoch up to the start of BeiDou epoch
00091 constexpr int32_t BEIDOU_DNAV_BDT2GPST_WEEK_NUM_OFFSET = 1356;
00092
00093 constexpr uint32_t BEIDOU_DNAV_SUBFRAME_SYMBOLS = 300;
00094 constexpr uint32_t BEIDOU_DNAV_WORDS_SUBFRAME = 10;
00095 constexpr uint32_t BEIDOU_DNAV_WORD_LENGTH_BITS = 30;
00096 constexpr char BEIDOU_DNAV_PREAMBLE[12] = "11100010010";

```

```
00097
00098
00099 // BEIDOU D1 NAVIGATION MESSAGE STRUCTURE
00100 // GENERAL
00101 const std::vector<std::pair<int32_t, int32_t> > D1_PRE({{1, 1}});
00102 const std::vector<std::pair<int32_t, int32_t> > D1_FRAID({{16, 3}});
00103 const std::vector<std::pair<int32_t, int32_t> > D1_SOW({{19, 8}, {31, 12}});
00104 const std::vector<std::pair<int32_t, int32_t> > D1_PNUM({{44, 7}});
00105
00106 // SUBFRAME 1
00107 const std::vector<std::pair<int32_t, int32_t> > D1_SAT_H1({{43, 1}});
00108 const std::vector<std::pair<int32_t, int32_t> > D1_AODC({{44, 5}});
00109 const std::vector<std::pair<int32_t, int32_t> > D1_URAI({{49, 4}});
00110 const std::vector<std::pair<int32_t, int32_t> > D1_WN({{61, 13}});
00111 const std::vector<std::pair<int32_t, int32_t> > D1_TOC({{74, 9}, {91, 8}});
00112 const std::vector<std::pair<int32_t, int32_t> > D1_TGD1({{99, 10}});
00113 const std::vector<std::pair<int32_t, int32_t> > D1_TGD2({{109, 4}, {121, 6}});
00114 const std::vector<std::pair<int32_t, int32_t> > D1_ALPHA0({{127, 8}});
00115 const std::vector<std::pair<int32_t, int32_t> > D1_ALPHA1({{135, 8}});
00116 const std::vector<std::pair<int32_t, int32_t> > D1_ALPHA2({{151, 8}});
00117 const std::vector<std::pair<int32_t, int32_t> > D1_ALPHA3({{159, 8}});
00118 const std::vector<std::pair<int32_t, int32_t> > D1_BETA0({{167, 6}, {181, 2}});
00119 const std::vector<std::pair<int32_t, int32_t> > D1_BETA1({{183, 8}});
00120 const std::vector<std::pair<int32_t, int32_t> > D1_BETA2({{191, 8}});
00121 const std::vector<std::pair<int32_t, int32_t> > D1_BETA3({{199, 4}, {211, 4}});
00122 const std::vector<std::pair<int32_t, int32_t> > D1_A2({{215, 11}});
00123 const std::vector<std::pair<int32_t, int32_t> > D1_A0({{226, 7}, {241, 17}});
00124 const std::vector<std::pair<int32_t, int32_t> > D1_A1({{258, 5}, {271, 17}});
00125 const std::vector<std::pair<int32_t, int32_t> > D1_AODE({{288, 5}});
00126
00127 // SUBFRAME 2
00128 const std::vector<std::pair<int32_t, int32_t> > D1_DELTA_N({{43, 10}, {61, 6}});
00129 const std::vector<std::pair<int32_t, int32_t> > D1_CUC({{67, 16}, {91, 2}});
00130 const std::vector<std::pair<int32_t, int32_t> > D1_M0({{93, 20}, {121, 12}});
00131 const std::vector<std::pair<int32_t, int32_t> > D1_E({{133, 10}, {151, 22}});
00132 const std::vector<std::pair<int32_t, int32_t> > D1_CUS({{181, 18}});
00133 const std::vector<std::pair<int32_t, int32_t> > D1_CRC({{199, 4}, {211, 14}});
00134 const std::vector<std::pair<int32_t, int32_t> > D1_CRS({{225, 8}, {241, 10}});
00135 const std::vector<std::pair<int32_t, int32_t> > D1_SQRT_A({{251, 12}, {271, 20}});
00136 const std::vector<std::pair<int32_t, int32_t> > D1_TOE_SF2({{291, 2}});
00137
00138 // SUBFRAME 3
00139 const std::vector<std::pair<int32_t, int32_t> > D1_TOE_SF3({{43, 10}, {61, 5}});
00140 const std::vector<std::pair<int32_t, int32_t> > D1_IO({{66, 17}, {91, 15}});
00141 const std::vector<std::pair<int32_t, int32_t> > D1_CIC({{106, 7}, {121, 11}});
00142 const std::vector<std::pair<int32_t, int32_t> > D1_OMEGA_DOT({{132, 11}, {151, 13}});
00143 const std::vector<std::pair<int32_t, int32_t> > D1_CIS({{164, 9}, {181, 9}});
00144 const std::vector<std::pair<int32_t, int32_t> > D1_IDOT({{190, 13}, {211, 1}});
00145 const std::vector<std::pair<int32_t, int32_t> > D1_OMEGA0({{212, 21}, {241, 11}});
00146 const std::vector<std::pair<int32_t, int32_t> > D1_OMEGA({{252, 11}, {271, 21}});
00147
00148 // SUBFRAME 4 AND PAGES 1 THROUGH 6 IN SUBFRAME 5
00149 const std::vector<std::pair<int32_t, int32_t> > D1_SQRT_A_ALMANAC({{51, 2}, {61, 22}});
00150 const std::vector<std::pair<int32_t, int32_t> > D1_A1_ALMANAC({{91, 11}});
00151 const std::vector<std::pair<int32_t, int32_t> > D1_A0_ALMANAC({{102, 11}});
00152 const std::vector<std::pair<int32_t, int32_t> > D1_OMEGA0_ALMANAC({{121, 22}, {151, 2}});
00153 const std::vector<std::pair<int32_t, int32_t> > D1_E_ALMANAC({{153, 17}});
00154 const std::vector<std::pair<int32_t, int32_t> > D1_DELTA_I({{170, 3}, {181, 13}});
00155 const std::vector<std::pair<int32_t, int32_t> > D1_TOA({{194, 8}});
00156 const std::vector<std::pair<int32_t, int32_t> > D1_OMEGA_DOT_ALMANAC({{202, 1}, {211, 16}});
00157 const std::vector<std::pair<int32_t, int32_t> > D1_OMEGA_ALMANAC({{227, 6}, {241, 18}});
00158 const std::vector<std::pair<int32_t, int32_t> > D1_M0_ALMANAC({{259, 4}, {271, 20}});
00159
00160 // SUBFRAME 5 PAGE 7
00161 const std::vector<std::pair<int32_t, int32_t> > D1_HEA1({{51, 2}, {61, 7}});
00162 const std::vector<std::pair<int32_t, int32_t> > D1_HEA2({{68, 9}});
00163 const std::vector<std::pair<int32_t, int32_t> > D1_HEA3({{77, 6}, {91, 3}});
00164 const std::vector<std::pair<int32_t, int32_t> > D1_HEA4({{94, 9}});
00165 const std::vector<std::pair<int32_t, int32_t> > D1_HEA5({{103, 9}});
00166 const std::vector<std::pair<int32_t, int32_t> > D1_HEA6({{112, 1}, {121, 8}});
00167 const std::vector<std::pair<int32_t, int32_t> > D1_HEA7({{129, 9}});
00168 const std::vector<std::pair<int32_t, int32_t> > D1_HEA8({{138, 5}, {151, 4}});
00169 const std::vector<std::pair<int32_t, int32_t> > D1_HEA9({{155, 9}});
00170 const std::vector<std::pair<int32_t, int32_t> > D1_HEA10({{164, 9}});
00171 const std::vector<std::pair<int32_t, int32_t> > D1_HEA11({{181, 9}});
00172 const std::vector<std::pair<int32_t, int32_t> > D1_HEA12({{190, 9}});
00173 const std::vector<std::pair<int32_t, int32_t> > D1_HEA13({{199, 4}, {211, 5}});
00174 const std::vector<std::pair<int32_t, int32_t> > D1_HEA14({{216, 9}});
00175 const std::vector<std::pair<int32_t, int32_t> > D1_HEA15({{225, 8}, {241, 1}});
00176 const std::vector<std::pair<int32_t, int32_t> > D1_HEA16({{242, 9}});
00177 const std::vector<std::pair<int32_t, int32_t> > D1_HEA17({{251, 9}});
00178 const std::vector<std::pair<int32_t, int32_t> > D1_HEA18({{260, 3}, {271, 6}});
00179 const std::vector<std::pair<int32_t, int32_t> > D1_HEA19({{277, 9}});
00180
00181 // SUBFRAME 5 PAGE 8
00182 const std::vector<std::pair<int32_t, int32_t> > D1_HEA20({{51, 2}, {61, 7}});
00183 const std::vector<std::pair<int32_t, int32_t> > D1_HEA21({{68, 9}});
```

```
00184 const std::vector<std::pair<int32_t, int32_t> > D1_HEA22({{77, 6}, {91, 3}});
00185 const std::vector<std::pair<int32_t, int32_t> > D1_HEA23({{94, 9}});
00186 const std::vector<std::pair<int32_t, int32_t> > D1_HEA24({{103, 9}});
00187 const std::vector<std::pair<int32_t, int32_t> > D1_HEA25({{112, 1}, {121, 8}});
00188 const std::vector<std::pair<int32_t, int32_t> > D1_HEA26({{129, 9}});
00189 const std::vector<std::pair<int32_t, int32_t> > D1_HEA27({{138, 5}, {151, 4}});
00190 const std::vector<std::pair<int32_t, int32_t> > D1_HEA28({{155, 9}});
00191 const std::vector<std::pair<int32_t, int32_t> > D1_HEA29({{164, 9}});
00192 const std::vector<std::pair<int32_t, int32_t> > D1_HEA30({{181, 9}});
00193 const std::vector<std::pair<int32_t, int32_t> > D1_WNA({{190, 8}});
00194 const std::vector<std::pair<int32_t, int32_t> > D1_TOA2({{198, 5}, {211, 3}});
00195
00196 // SUBFRAME 5 PAGE 9
00197 const std::vector<std::pair<int32_t, int32_t> > D1_AOGPS({{97, 14}});
00198 const std::vector<std::pair<int32_t, int32_t> > D1_AIGPS({{111, 2}, {121, 14}});
00199 const std::vector<std::pair<int32_t, int32_t> > D1_AOGAL({{135, 8}, {151, 6}});
00200 const std::vector<std::pair<int32_t, int32_t> > D1_AIGAL({{157, 16}});
00201 const std::vector<std::pair<int32_t, int32_t> > D1_AOGLO({{181, 14}});
00202 const std::vector<std::pair<int32_t, int32_t> > D1_AIGLO({{195, 8}, {211, 8}});
00203
00204 // SUBFRAME 5 PAGE 10
00205 const std::vector<std::pair<int32_t, int32_t> > D1_DELTA_T_LS({{51, 2}, {61, 6}});
00206 const std::vector<std::pair<int32_t, int32_t> > D1_DELTA_T_LSF({{67, 8}});
00207 const std::vector<std::pair<int32_t, int32_t> > D1_WN_LSF({{75, 8}});
00208 const std::vector<std::pair<int32_t, int32_t> > D1_AOUTC({{91, 22}, {121, 10}});
00209 const std::vector<std::pair<int32_t, int32_t> > D1_AUTC({{131, 12}, {151, 12}});
00210 const std::vector<std::pair<int32_t, int32_t> > D1_DN({{163, 8}});
00211
00212 // D2 NAV Message Decoding Information
00213 const std::vector<std::pair<int32_t, int32_t> > D2_PRE({{1, 11}});
00214 const std::vector<std::pair<int32_t, int32_t> > D2_FRAID({{16, 3}});
00215 const std::vector<std::pair<int32_t, int32_t> > D2_SOW({{19, 8}, {31, 12}});
00216 const std::vector<std::pair<int32_t, int32_t> > D2_PNUM({{43, 4}});
00217
00218 // D2 NAV, SUBFRAME 1, PAGE 1
00219 const std::vector<std::pair<int32_t, int32_t> > D2_SAT_H1({{47, 1}});
00220 const std::vector<std::pair<int32_t, int32_t> > D2_AODC({{48, 5}});
00221 const std::vector<std::pair<int32_t, int32_t> > D2_URAI({{61, 4}});
00222 const std::vector<std::pair<int32_t, int32_t> > D2_WN({{65, 13}});
00223 const std::vector<std::pair<int32_t, int32_t> > D2_TOC({{78, 5}, {91, 12}});
00224 const std::vector<std::pair<int32_t, int32_t> > D2_TGD1({{103, 10}});
00225 const std::vector<std::pair<int32_t, int32_t> > D2_TGD2({{121, 10}});
00226
00227 // D2 NAV, SUBFRAME 1, PAGE 2
00228 const std::vector<std::pair<int32_t, int32_t> > D2_ALPHA0({{47, 6}, {61, 2}});
00229 const std::vector<std::pair<int32_t, int32_t> > D2_ALPHA1({{63, 8}});
00230 const std::vector<std::pair<int32_t, int32_t> > D2_ALPHA2({{71, 8}});
00231 const std::vector<std::pair<int32_t, int32_t> > D2_ALPHA3({{79, 4}, {91, 4}});
00232 const std::vector<std::pair<int32_t, int32_t> > D2_BETA0({{95, 8}});
00233 const std::vector<std::pair<int32_t, int32_t> > D2_BETA1({{103, 8}});
00234 const std::vector<std::pair<int32_t, int32_t> > D2_BETA2({{111, 2}, {121, 6}});
00235 const std::vector<std::pair<int32_t, int32_t> > D2_BETA3({{127, 8}});
00236
00237 // D2 NAV, SUBFRAME 1, PAGE 3
00238 const std::vector<std::pair<int32_t, int32_t> > D2_A0({{101, 12}, {121, 12}});
00239 const std::vector<std::pair<int32_t, int32_t> > D2_A1_MSB({{133, 4}});
00240 const std::vector<std::pair<int32_t, int32_t> > D2_A1_LSB({{47, 6}, {61, 12}});
00241 const std::vector<std::pair<int32_t, int32_t> > D2_A1({{279, 22}});
00242
00243 // D2 NAV, SUBFRAME 1, PAGE 4
00244 const std::vector<std::pair<int32_t, int32_t> > D2_A2({{73, 10}, {91, 1}});
00245 const std::vector<std::pair<int32_t, int32_t> > D2_AODE({{92, 5}});
00246 const std::vector<std::pair<int32_t, int32_t> > D2_DELTA_N({{97, 16}});
00247 const std::vector<std::pair<int32_t, int32_t> > D2_CUC_MSB({{121, 14}});
00248 const std::vector<std::pair<int32_t, int32_t> > D2_CUC_LSB({{47, 4}});
00249 const std::vector<std::pair<int32_t, int32_t> > D2_CUC({{283, 18}});
00250
00251 // D2 NAV, SUBFRAME 1, PAGE 5
00252 const std::vector<std::pair<int32_t, int32_t> > D2_M0({{51, 2}, {61, 22}, {91, 8}});
00253 const std::vector<std::pair<int32_t, int32_t> > D2_CUS({{99, 14}, {121, 4}});
00254 const std::vector<std::pair<int32_t, int32_t> > D2_E_MSB({{125, 10}});
00255
00256 // D2 NAV, SUBFRAME 1, PAGE 6
00257 const std::vector<std::pair<int32_t, int32_t> > D2_E_LSB({{47, 6}, {61, 16}});
00258 const std::vector<std::pair<int32_t, int32_t> > D2_SQRT_A({{77, 6}, {91, 22}, {121, 4}});
00259 const std::vector<std::pair<int32_t, int32_t> > D2_CIC_MSB({{125, 10}});
00260 const std::vector<std::pair<int32_t, int32_t> > D2_CIC_LSB({{47, 6}, {61, 2}});
00261 const std::vector<std::pair<int32_t, int32_t> > D2_CIC({{283, 18}});
00262
00263 // D2 NAV, SUBFRAME 1, PAGE 7
00264 const std::vector<std::pair<int32_t, int32_t> > D2_CIS({{63, 18}});
00265 const std::vector<std::pair<int32_t, int32_t> > D2_TOE({{81, 2}, {91, 15}});
00266 const std::vector<std::pair<int32_t, int32_t> > D2_I0_MSB({{106, 7}, {121, 14}});
00267 const std::vector<std::pair<int32_t, int32_t> > D2_I0_LSB({{47, 6}, {61, 5}});
00268 const std::vector<std::pair<int32_t, int32_t> > D2_I0({{269, 32}});
00269
00270 // D2 NAV, SUBFRAME 1, PAGE 8
```

```

00271 const std::vector<std::pair<int32_t, int32_t> > D2_CRC({{66, 17}, {91, 1}});
00272 const std::vector<std::pair<int32_t, int32_t> > D2_CRS({{92, 18}});
00273 const std::vector<std::pair<int32_t, int32_t> > D2_OMEGA_DOT_MSB({{110, 3}, {121, 16}});
00274 const std::vector<std::pair<int32_t, int32_t> > D2_OMEGA_DOT_LSB({{47, 5}});
00275 const std::vector<std::pair<int32_t, int32_t> > D2_OMEGA_DOT({{277, 24}});
00276
00277 // D2 NAV, SUBFRAME 1, PAGE 9
00278 const std::vector<std::pair<int32_t, int32_t> > D2_OMEGA0({{52, 1}, {61, 22}, {91, 9}});
00279 const std::vector<std::pair<int32_t, int32_t> > D2_OMEGA_MSB({{100, 13}, {121, 14}});
00280 const std::vector<std::pair<int32_t, int32_t> > D2_OMEGA_LSB({{47, 5}});
00281 const std::vector<std::pair<int32_t, int32_t> > D2_OMEGA({{269, 32}});
00282
00283 // D2 NAV, SUBFRAME 1, PAGE 10
00284 const std::vector<std::pair<int32_t, int32_t> > D2_IDOT({{52, 1}, {61, 13}});
00285
00286
00287 /** \} */
00288 /** \} */
00289 #endif // GNSS_SDR_BEIDOU_DNAV_H

```

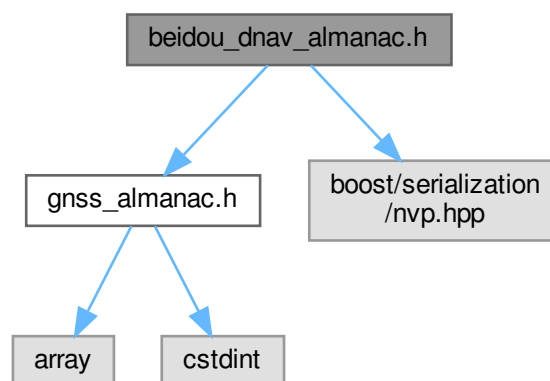
## 11.679 beidou\_dnav\_almanac.h File Reference

Interface of a Beidou DNAV Almanac storage.

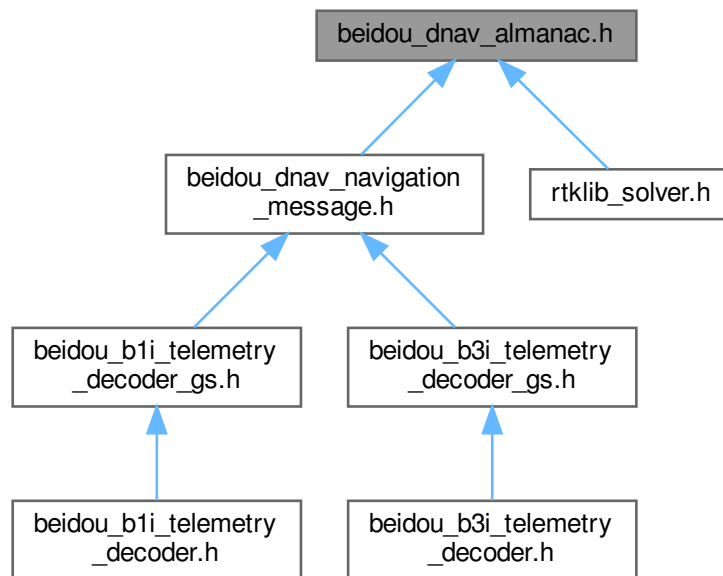
```
#include "gnss_almanac.h"
```

```
#include <boost/serialization/nvp.hpp>
```

Include dependency graph for beidou\_dnav\_almanac.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Beidou\\_Dnav\\_Almanac](#)

*This class is a storage for the BeiDou D1 almanac.*

## 11.679.1 Detailed Description

Interface of a Beidou DNAV Almanac storage.

### Author

Sergi Segura, 2018. [sergi.segura.munoz\(at\)gmail.com](mailto:sergi.segura.munoz(at)gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [beidou\\_dnav\\_almanac.h](#).

---

## 11.680 beidou\_dnav\_almanac.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file beidou_dnav_almanac.h
00003  * \brief Interface of a Beidou DNAV Almanac storage
00004  * \author Sergi Segura, 2018. sergi.segura.munoz\(at\)gmail.com
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016

```



```

00017
00018 #ifndef GNSS_SDR_BEIDOU_DNAV_ALMANAC_H
00019 #define GNSS_SDR_BEIDOU_DNAV_ALMANAC_H
00020
00021 #include "gnss_almanac.h"
00022 #include <boost/serialization/nvp.hpp>
00023
00024 /** \addtogroup Core
00025  * \{ */
00026 /** \addtogroup System_Parameters
00027  * \{ */
00028
00029
00030 /*!
00031  * \brief This class is a storage for the BeiDou D1 almanac
00032  */
00033 class Beidou_Dnav_Almanac : public Gnss_Almanac
00034 {
00035 public:
00036     /*!
00037      * Default constructor
00038      */
00039     Beidou_Dnav_Almanac()
00040     {
00041         this->System = 'B';
00042     };
00043
00044     int SV_health{};    //!< SV Health
00045
00046     template <class Archive>
00047
00048     void serialize(Archive& ar, const unsigned int version)
00049     {
00050         if (version)
00051         {
00052             };
00053             ar& BOOST_SERIALIZATION_NVP (PRN);
00054             ar& BOOST_SERIALIZATION_NVP (delta_i);
00055             ar& BOOST_SERIALIZATION_NVP (toa);
00056             ar& BOOST_SERIALIZATION_NVP (WNa);
00057             ar& BOOST_SERIALIZATION_NVP (M_0);
00058             ar& BOOST_SERIALIZATION_NVP (ecc);
00059             ar& BOOST_SERIALIZATION_NVP (sqrtA);
00060             ar& BOOST_SERIALIZATION_NVP (OMEGA_0);
00061             ar& BOOST_SERIALIZATION_NVP (omega);
00062             ar& BOOST_SERIALIZATION_NVP (OMEGAdot);
00063             ar& BOOST_SERIALIZATION_NVP (af0);
00064             ar& BOOST_SERIALIZATION_NVP (af1);
00065             ar& BOOST_SERIALIZATION_NVP (SV_health);
00066         }
00067     };
00068
00069
00070 /** \} */
00071 /** \} */
00072 #endif // GNSS_SDR_BEIDOU_DNAV_ALMANAC_H

```

## 11.681 beidou\_dnav\_ephemeris.h File Reference

Interface of a BEIDOU EPHEMERIS storage.

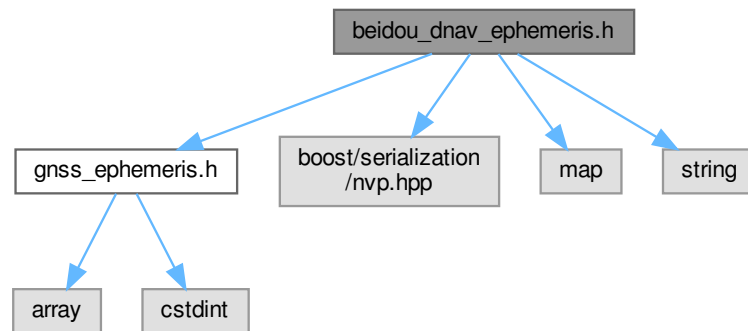
```

#include "gnss_ephemeris.h"
#include <boost/serialization/nvp.hpp>
#include <map>
#include <string>

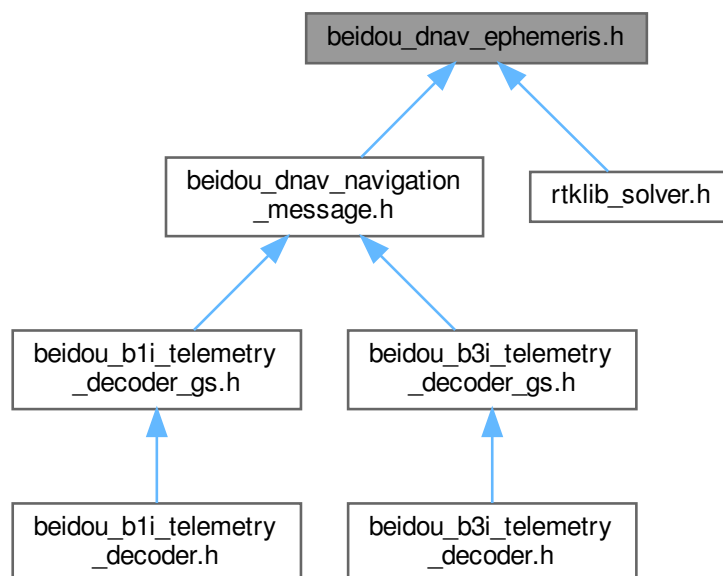
```



Include dependency graph for beidou\_dnav\_ephemeris.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Beidou\\_Dnav\\_Ephemeris](#)

*This is a storage class for the Beidou SV ephemeris data as described in BeiDou Navigation Satellite System Signal In Space Interface Control Document Open Service Signal B1I (Version 3.0)*

## 11.681.1 Detailed Description

Interface of a BEIDOU EPHEMERIS storage.

## Author

Sergi Segura, 2018. [sergi.segura.munoz\(at\)gmail.com](mailto:sergi.segura.munoz(at)gmail.com)

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [beidou\\_dnav\\_ephemeris.h](#).

## 11.682 beidou\_dnav\_ephemeris.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file beidou_dnav_ephemeris.h
00003  * \brief Interface of a BEIDOU EPHEMERIS storage
00004  * \author Sergi Segura, 2018. sergi.segura.munoz\(at\)gmail.com
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_BEIDOU_DNAV_EPHEMERIS_H
00019 #define GNSS_SDR_BEIDOU_DNAV_EPHEMERIS_H
00020
00021 #include "gnss_ephemeris.h"
00022 #include <boost/serialization/nvp.hpp>
00023 #include <map>
00024 #include <string>
00025
00026 /** \addtogroup Core
00027  * \{ */
00028 /** \addtogroup System_Parameters
00029  * \{ */
00030
00031
00032  /*!
00033  * \brief This is a storage class for the Beidou SV ephemeris data as described in
00034  * Beidou Navigation Satellite System Signal In Space Interface Control Document
00035  * Open Service Signal B1I (Version 3.0)
00036  *
00037  * See http://en.beidou.gov.cn/SYSTEMS/Officialdocument/201902/P020190227601370045731.pdf
00038  */
00039 class Beidou_Dnav_Ephemeris : public Gnss_Ephemeris
00040 {
00041 public:
00042  /*!
00043  * Default constructor
00044  */
00045  Beidou_Dnav_Ephemeris();
00046
00047  int SV_accuracy{}; //!< User Range Accuracy (URA) index of the SV (reference paragraph 5.2.4.5)
00048  for the standard positioning service user
00049  int SV_health{};
00050  double TGD1{}; //!< Estimated Group Delay Differential on B1I [s]
00051  double TGD2{}; //!< Estimated Group Delay Differential on B2I [s]
00052  double AODC{}; //!< Age of Data, Clock
00053  double AODE{}; //!< Age of Data, Ephemeris
00054  int AODO{}; //!< Age of Data Offset (AODO) term for the navigation message correction table
00055  (NMCT) contained in subframe 4 (reference paragraph 20.3.3.5.1.9) [s]
00056
00057  int sig_type{}; //!< BDS: data source (0:unknown,1:B1I,2:B1Q,3:B2I,4:B2Q,5:B3I,6:B3Q) */
00058  int nav_type{}; //!< BDS: nav type (0:unknown,1:IGSO/MEO,2:GEO) */
00059
00060  bool fit_interval_flag{}; //!< Curve-fit interval used by the CS (Block II/IIA/IIR/IIR-M/IIF) and
00061  SS (Block IIIA) in determining the ephemeris parameters, as follows: 0 = 4 hours, 1 = greater than 4
00062  hours.
00063  double spare1{};
00064  double spare2{};
00065
00066  /*! \brief If true, enhanced level of integrity assurance.
00067  *
00068  * If false, indicates that the conveying signal is provided with the
00069  * legacy level of integrity assurance. That is, the probability that the
00070  * instantaneous URE of the conveying signal exceeds 4.42 times the upper
00071  * bound value of the current broadcast URA index, for more than 5.2
00072  * seconds, without an accompanying alert, is less than 1E-5 per hour. If

```

```

00069     * true, indicates that the conveying signal is provided with an enhanced
00070     * level of integrity assurance. That is, the probability that the
00071     * instantaneous URE of the conveying signal exceeds 5.73 times the upper
00072     * bound value of the current broadcast URA index, for more than 5.2
00073     * seconds, without an accompanying alert, is less than 1E-8 per hour.
00074     */
00075     bool integrity_status_flag{};
00076     bool alert_flag{}; //!< If true, indicates that the SV URA may be worse than indicated in
d_SV_accuracy, use that SV at our own risk.
00077     bool antispoofing_flag{}; //!< If true, the AntiSpoofing mode is ON in that SV
00078
00079     std::map<int, std::string> satelliteBlock; //!< Map that stores to which block the PRN belongs
00080
00081     template <class Archive>
00082
00083     /*!
00084     * \brief Serialize is a boost standard method to be called by the boost XML
00085     * serialization. Here is used to save the ephemeris data on disk file.
00086     */
00087     void serialize(Archive& archive, const unsigned int version)
00088     {
00089         using boost::serialization::make_nvp;
00090         if (version)
00091         {
00092             };
00093
00094             archive& BOOST_SERIALIZATION_NVP(PRN);
00095             archive& BOOST_SERIALIZATION_NVP(M_0);
00096             archive& BOOST_SERIALIZATION_NVP(delta_n);
00097             archive& BOOST_SERIALIZATION_NVP(ecc);
00098             archive& BOOST_SERIALIZATION_NVP(sqrtA);
00099             archive& BOOST_SERIALIZATION_NVP(OMEGA_0);
00100             archive& BOOST_SERIALIZATION_NVP(i_0);
00101             archive& BOOST_SERIALIZATION_NVP(omega);
00102             archive& BOOST_SERIALIZATION_NVP(OMEGAdot);
00103             archive& BOOST_SERIALIZATION_NVP(idot);
00104             archive& BOOST_SERIALIZATION_NVP(Cuc);
00105             archive& BOOST_SERIALIZATION_NVP(Cus);
00106             archive& BOOST_SERIALIZATION_NVP(Crc);
00107             archive& BOOST_SERIALIZATION_NVP(Crs);
00108             archive& BOOST_SERIALIZATION_NVP(Cic);
00109             archive& BOOST_SERIALIZATION_NVP(Cis);
00110             archive& BOOST_SERIALIZATION_NVP(toe);
00111             archive& BOOST_SERIALIZATION_NVP(toc);
00112             archive& BOOST_SERIALIZATION_NVP(af0);
00113             archive& BOOST_SERIALIZATION_NVP(af1);
00114             archive& BOOST_SERIALIZATION_NVP(af2);
00115             archive& BOOST_SERIALIZATION_NVP(WN);
00116             archive& BOOST_SERIALIZATION_NVP(tow);
00117             archive& BOOST_SERIALIZATION_NVP(satClkDrift);
00118             archive& BOOST_SERIALIZATION_NVP(dtr);
00119
00120             archive& BOOST_SERIALIZATION_NVP(AODE);
00121             archive& BOOST_SERIALIZATION_NVP(SV_accuracy);
00122             archive& BOOST_SERIALIZATION_NVP(SV_health);
00123             archive& BOOST_SERIALIZATION_NVP(AODC);
00124             archive& BOOST_SERIALIZATION_NVP(TGD1);
00125             archive& BOOST_SERIALIZATION_NVP(TGD2);
00126             archive& BOOST_SERIALIZATION_NVP(sig_type);
00127             archive& BOOST_SERIALIZATION_NVP(nav_type);
00128             archive& BOOST_SERIALIZATION_NVP(AODO);
00129             archive& BOOST_SERIALIZATION_NVP(fit_interval_flag);
00130             archive& BOOST_SERIALIZATION_NVP(spare1);
00131             archive& BOOST_SERIALIZATION_NVP(spare2);
00132             archive& BOOST_SERIALIZATION_NVP(integrity_status_flag);
00133             archive& BOOST_SERIALIZATION_NVP(alert_flag);
00134             archive& BOOST_SERIALIZATION_NVP(antispoofing_flag);
00135         }
00136     };
00137
00138
00139     /** \} */
00140     /** \} */
00141     #endif // GNSS_SDR_BEIDOU_DNAV_EPHEMERIS_H

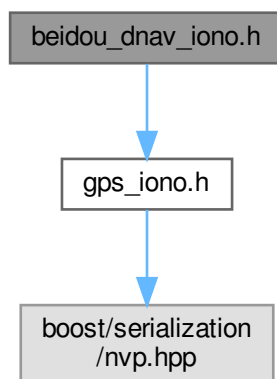
```

## 11.683 beidou\_dnav\_iono.h File Reference

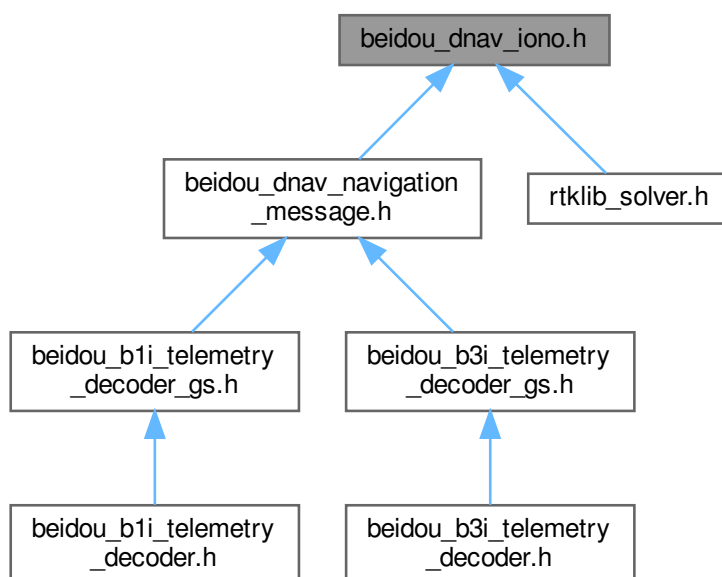
Interface of a BEIDOU IONOSPHERIC MODEL storage.

```
#include "gps_iono.h"
```

Include dependency graph for beidou\_dnav\_iono.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Beidou\\_Dnav\\_Iono](#)

*This class is a storage for the BEIDOU IONOSPHERIC data as described in ICD v2.1.*

### 11.683.1 Detailed Description

Interface of a BEIDOU IONOSPHERIC MODEL storage.

#### Author

Sergi Segura, 2018. sergi.segura.munoz(at)gmail.com

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [beidou\\_dnav\\_iono.h](#).

---

## 11.684 beidou\_dnav\_iono.h

[Go to the documentation of this file.](#)

```
00001  /*!
00002  * \file beidou_dnav_iono.h
00003  * \brief Interface of a BEIDOU IONOSPHERIC MODEL storage
00004  * \author Sergi Segura, 2018. sergi.segura.munoz(at)gmail.com
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_BEIDOU_DNAV_IONO_H
00019 #define GNSS_SDR_BEIDOU_DNAV_IONO_H
00020
00021 #include "gps_iono.h"
00022
00023 /** \addtogroup Core
00024  * \{ */
00025 /** \addtogroup System_Parameters
00026  * \{ */
00027
00028
00029 /*!
00030 * \brief This class is a storage for the BEIDOU IONOSPHERIC data as described
00031 * in ICD v2.1
00032 */
00033 class Beidou_Dnav_Iono : public Gps_Iono
00034 {
00035 public:
00036     Beidou_Dnav_Iono() = default; //!< Default constructor
00037 };
00038
00039
00040 /** \} */
00041 /** \} */
00042 #endif // GNSS_SDR_BEIDOU_DNAV_IONO_H
```

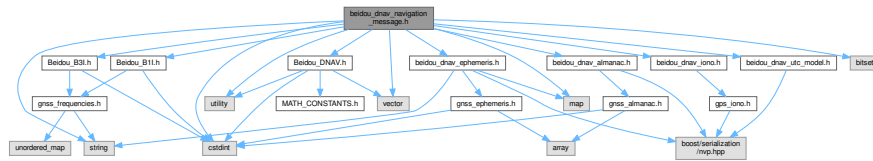
## 11.685 beidou\_dnav\_navigation\_message.h File Reference

Interface of a BeiDou DNAV Data message decoder.

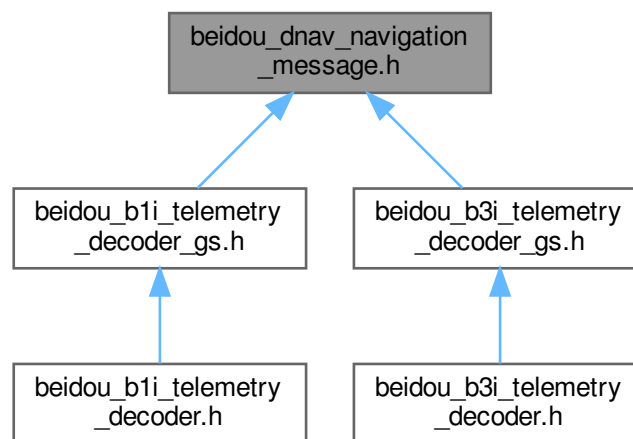
```
#include "Beidou_B1I.h"
#include "Beidou_B3I.h"
#include "Beidou_DNAV.h"
#include "beidou_dnav_almanac.h"
#include "beidou_dnav_ephemeris.h"
#include "beidou_dnav_iono.h"
#include "beidou_dnav_utc_model.h"
#include <bitset>
#include <cstdint>
#include <map>
#include <string>
#include <utility>
```

```
#include <vector>
```

Include dependency graph for beidou\_dnav\_navigation\_message.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Beidou\\_Dnav\\_Navigation\\_Message](#)

*This class decodes a BeiDou D1 NAV Data message.*

## 11.685.1 Detailed Description

Interface of a BeiDou DNAV Data message decoder.

### Author

Sergi Segura, 2018. [sergi.segura.munoz\(at\)gmail.com](mailto:sergi.segura.munoz(at)gmail.com)

Damian Miralles, 2018. [dmiralles2009@gmail.com](mailto:dmiralles2009@gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [beidou\\_dnav\\_navigation\\_message.h](#).

---

## 11.686 beidou\_dnav\_navigation\_message.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file beidou_dnav_navigation_message.h
00003  * \brief Interface of a BeiDou DNAV Data message decoder
```

```

00004  * \author Sergi Segura, 2018. sergi.segura.munoz(at)gmail.com
00005  * \author Damian Miralles, 2018. dmiralles2009@gmail.com
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018
00019 #ifndef GNSS_SDR_BEIDOU_DNAV_NAVIGATION_MESSAGE_H
00020 #define GNSS_SDR_BEIDOU_DNAV_NAVIGATION_MESSAGE_H
00021
00022
00023 #include "Beidou_B1I.h"
00024 #include "Beidou_B3I.h"
00025 #include "Beidou_DNAV.h"
00026 #include "beidou_dnav_almanac.h"
00027 #include "beidou_dnav_ephemeris.h"
00028 #include "beidou_dnav_iono.h"
00029 #include "beidou_dnav_utc_model.h"
00030 #include <bitset>
00031 #include <cstdint>
00032 #include <map>
00033 #include <string>
00034 #include <utility>
00035 #include <vector>
00036
00037 /** \addtogroup Core
00038  * \{ */
00039 /** \addtogroup System_Parameters
00040  * \{ */
00041
00042
00043 /*!
00044  * \brief This class decodes a BeiDou D1 NAV Data message
00045  */
00046 class Beidou_Dnav_Navigation_Message
00047 {
00048 public:
00049     /*!
00050      * Default constructor
00051      */
00052     Beidou_Dnav_Navigation_Message();
00053
00054     /*!
00055      * \brief Obtain a BDS SV Ephemeris class filled with current SV data
00056      */
00057     Beidou_Dnav_Ephemeris get_ephemeris() const;
00058
00059     /*!
00060      * \brief Obtain a BDS ionospheric correction parameters class filled with current SV data
00061      */
00062     Beidou_Dnav_Iono get_iono();
00063
00064     /*!
00065      * \brief Obtain a BDS UTC model parameters class filled with current SV data
00066      */
00067     Beidou_Dnav_Utc_Model get_utc_model();
00068
00069     /*!
00070      * \brief Decodes the BDS D1 NAV message
00071      */
00072     int32_t d1_subframe_decoder(std::string const& subframe);
00073
00074     /*!
00075      * \brief Decodes the BDS D2 NAV message
00076      */
00077     int32_t d2_subframe_decoder(std::string const& subframe);
00078
00079     /*!
00080      * \brief Computes the Coordinated Universal Time (UTC) and
00081      * returns it in [s]
00082      */
00083     double utc_time(double beidoutime_corrected) const;
00084
00085     bool satellite_validation();
00086
00087     /*!
00088      * \brief Returns true if new Ephemeris has arrived. The flag is set to false when the function is
00089     executed
00089      */

```

```

00090     bool have_new_ephemeris();
00091
00092     /*!
00093     * \brief Returns true if new Iono model has arrived. The flag is set to false when the function
is executed
00094     */
00095     bool have_new_iono() const;
00096
00097     /*!
00098     * \brief Returns true if new UTC model has arrived. The flag is set to false when the function is
executed
00099     */
00100     bool have_new_utc_model();
00101
00102     /*!
00103     * \brief Returns true if new UTC model has arrived. The flag is set to false when the function is
executed
00104     */
00105     bool have_new_almanac();
00106
00107     /*!
00108     * \brief Sets satellite PRN number
00109     */
00110     inline void set_satellite_PRN(uint32_t prn)
00111     {
00112         i_satellite_PRN = prn;
00113     }
00114
00115     inline void set_signal_type(int32_t signal_type)
00116     {
00117         i_signal_type = signal_type;
00118     }
00119
00120     inline bool get_flag_CRC_test() const
00121     {
00122         return flag_crc_test;
00123     }
00124
00125     inline bool get_flag_new_SOW_available() const
00126     {
00127         return flag_new_SOW_available;
00128     }
00129
00130     inline void set_flag_new_SOW_available(bool new_SOW_available)
00131     {
00132         flag_new_SOW_available = new_SOW_available;
00133     }
00134
00135     inline double get_SOW() const
00136     {
00137         return d_SOW;
00138     }
00139
00140 private:
00141     uint64_t read_navigation_unsigned(const std::bitset<BEIDOU_DNAV_SUBFRAME_DATA_BITS>& bits, const
std::vector<std::pair<int32_t, int32_t>& parameter> const;
00142     int64_t read_navigation_signed(const std::bitset<BEIDOU_DNAV_SUBFRAME_DATA_BITS>& bits, const
std::vector<std::pair<int32_t, int32_t>& parameter> const;
00143     bool read_navigation_bool(const std::bitset<BEIDOU_DNAV_SUBFRAME_DATA_BITS>& bits, const
std::vector<std::pair<int32_t, int32_t>& parameter> const;
00144     void print_beidou_word_bytes(uint32_t BEIDOU_word) const;
00145
00146     // broadcast orbit 1
00147     double d_SOW{}; // Time of BeiDou Week of the ephemeris set (taken from subframes SOW) [s]
00148     double d_SOW_SF1{}; // Time of BeiDou Week from HOW word of Subframe 1 [s]
00149     double d_SOW_SF2{}; // Time of BeiDou Week from HOW word of Subframe 2 [s]
00150     double d_SOW_SF3{}; // Time of BeiDou Week from HOW word of Subframe 3 [s]
00151     double d_SOW_SF4{}; // Time of BeiDou Week from HOW word of Subframe 4 [s]
00152     double d_SOW_SF5{}; // Time of BeiDou Week from HOW word of Subframe 5 [s]
00153
00154     double d_AODE{};
00155     double d_Crs{}; // Amplitude of the Sine Harmonic Correction Term to the Orbit Radius [m]
00156     double d_Delta_n{}; // Mean Motion Difference From Computed Value [semi-circles/s]
00157     double d_M_0{}; // Mean Anomaly at Reference Time [semi-circles]
00158
00159     // broadcast orbit 2
00160     double d_Cuc{}; // Amplitude of the Cosine Harmonic Correction Term to the Argument of
Latitude [rad]
00161     double d_eccentricity{}; // Eccentricity [dimensionless]
00162     double d_Cus{}; // Amplitude of the Sine Harmonic Correction Term to the Argument of
Latitude [rad]
00163     double d_sqrt_A{}; // Square Root of the Semi-Major Axis [sqrt(m)]
00164
00165     // broadcast orbit 3
00166     double d_Toe_sf2{}; // Ephemeris data reference time of week in subframe 2, D1 Message
00167     double d_Toe_sf3{}; // Ephemeris data reference time of week in subframe 3, D1 Message
00168     double d_Toe{}; // Ephemeris data reference time of week in subframe 1, D2 Message

```



```

00169     double d_Toc{};           // clock data reference time [s]
00170     double d_Cic{};           // Amplitude of the Cosine Harmonic Correction Term to the Angle of
Inclination [rad]
00171     double d_OMEGA0{};        // Longitude of Ascending Node of Orbit Plane at Weekly Epoch [semi-circles]
00172     double d_Cis{};           // Amplitude of the Sine Harmonic Correction Term to the Angle of Inclination
[rad]
00173
00174     // broadcast orbit 4
00175     double d_i_0{};           // Inclination Angle at Reference Time [semi-circles]
00176     double d_Crc{};           // Amplitude of the Cosine Harmonic Correction Term to the Orbit Radius [m]
00177     double d_OMEGA{};         // Argument of Perigee [semi-circles]
00178     double d_OMEGA_DOT{};     // Rate of Right Ascension [semi-circles/s]
00179
00180     // broadcast orbit 5
00181     double d_IDOT{};           // Rate of Inclination Angle [semi-circles/s]
00182     int32_t i_BEIDOU_week{};   // BeiDou week number, aka WN [week]
00183
00184     // broadcast orbit 6
00185     int32_t i_SV_accuracy{};   // User Range Accuracy (URA) index of the SV
00186     int32_t i_SV_health{};
00187     double d_TGD1{};          // Estimated Group Delay Differential in B1 [s]
00188     double d_TGD2{};          // Estimated Group Delay Differential in B2 [s]
00189     double d_AODC{};          // Age of Data, Clock
00190
00191     // broadcast orbit 7
00192     // int32_t i_AODO{};        // Age of Data Offset (AODO) term for the navigation message
correction table (NMCT) contained in subframe 4 (reference paragraph 20.3.3.5.1.9) [s]
00193
00194     // bool b_fit_interval_flag{}; // indicates the curve-fit interval used by the CS (Block
II/IIA/IIR/IIR-M/IIF) and SS (Block IIIA) in determining the ephemeris parameters, as follows: 0 = 4
hours, 1 = greater than 4 hours.
00195     // double d_spare1{};
00196     // double d_spare2{};
00197
00198     double d_A_f0{};          // Clock correction parameters. Coefficient 0 of code phase offset model [s]
00199     double d_A_f1{};          // Clock correction parameters. Coefficient 1 of code phase offset model [s/s]
00200     double d_A_f2{};          // Clock correction parameters. Coefficient 2 of code phase offset model [s/s^2]
00201
00202     // D2 NAV Message Decoding
00203     uint64_t d_A_f1_msb_bits{}; // Clock correction parameters, D2 NAV MSB
00204     uint64_t d_A_f1_lsb_bits{}; // Clock correction parameters, D2 NAV LSB
00205     uint64_t d_Cuc_msb_bits{};   // Amplitude of the Cosine Harmonic Correction Term to the
Argument of Latitude [rad]
00206     uint64_t d_Cuc_lsb_bits{};   // Amplitude of the Cosine Harmonic Correction Term to the
Argument of Latitude [rad]
00207     uint64_t d_eccentricity_msb{}; // Eccentricity [dimensionless]
00208     uint64_t d_eccentricity_lsb{}; // Eccentricity [dimensionless]
00209     uint64_t d_Cic_msb_bits{};   // Amplitude of the Cosine Harmonic Correction Term to the
Argument of Latitude [rad]
00210     uint64_t d_Cic_lsb_bits{};   // Amplitude of the Cosine Harmonic Correction Term to the
Argument of Latitude [rad]
00211     uint64_t d_eccentricity_msb_bits{}; // Eccentricity [dimensionless]
00212     uint64_t d_eccentricity_lsb_bits{};
00213     uint64_t d_i_0_msb_bits{};   // Inclination Angle at Reference Time [semi-circles]
00214     uint64_t d_i_0_lsb_bits{};   // Inclination Angle at Reference Time [semi-circles]
00215     uint64_t d_OMEGA_msb_bits{}; // Argument of Perigee [semi-circles]
00216     uint64_t d_OMEGA_lsb_bits{}; // Argument of Perigee [semi-circles]
00217     uint64_t d_OMEGA_DOT_msb_bits{}; // Rate of Right Ascension [semi-circles/s]
00218     uint64_t d_OMEGA_DOT_lsb_bits{}; // Rate of Right Ascension [semi-circles/s]
00219
00220     // Almanac
00221     // double d_Toa{};           // Almanac reference time [s]
00222     // int32_t i_WN_A{};         // Modulo 256 of the GPS week number to which the
almanac reference time (d_Toa) is referenced
00223     std::map<int32_t, int32_t> almanacHealth; // Map that stores the health information stored in the
almanac
00224
00225     std::map<int32_t, std::string> satelliteBlock; // Map that stores to which block the PRN belongs
00226
00227     // satellite identification info
00228     int32_t i_signal_type{}; // BDS: data source (0:unknown,1:B1I,2:B1Q,3:B2I,4:B2Q,5:B3I,6:B3Q)
00229     uint32_t i_satellite_PRN{};
00230
00231     // Ionospheric parameters
00232     double d_alpha0{}; // Coefficient 0 of a cubic equation representing the amplitude of the
vertical delay [s]
00233     double d_alpha1{}; // Coefficient 1 of a cubic equation representing the amplitude of the
vertical delay [s/semi-circle]
00234     double d_alpha2{}; // Coefficient 2 of a cubic equation representing the amplitude of the
vertical delay [s(semi-circle)^2]
00235     double d_alpha3{}; // Coefficient 3 of a cubic equation representing the amplitude of the
vertical delay [s(semi-circle)^3]
00236     double d_beta0{}; // Coefficient 0 of a cubic equation representing the period of the model [s]
00237     double d_beta1{}; // Coefficient 1 of a cubic equation representing the period of the model
[s/semi-circle]
00238     double d_beta2{}; // Coefficient 2 of a cubic equation representing the period of the model
[s(semi-circle)^2]

```

```

00239     double d_beta3{};    // Coefficient 3 of a cubic equation representing the period of the model
                                [s(semi-circle)^3]
00240
00241     // UTC parameters
00242     double d_A1UTC{};      // 1st order term of a model that relates GPS and UTC time [s/s]
00243     double d_A0UTC{};      // Constant of a model that relates GPS and UTC time [s]
00244     int32_t i_DeltaT_LS{}; // delta time due to leap seconds [s]. Number of leap seconds since
                                6-Jan-1980 as transmitted by the GPS almanac.
00245     int32_t i_WN_LSF{};    // Week number at the end of which the leap second becomes effective
                                [weeks]
00246     int32_t i_DN{};        // Day number (DN) at the end of which the leap second becomes effective
                                [days]
00247     double d_DeltaT_LSF{}; // Scheduled future or recent past (relative to NAV message upload) value
                                of the delta time due to leap seconds [s]
00248     double d_A1GPS{};
00249     double d_A0GPS{};
00250     double d_A1GAL{};
00251     double d_A0GAL{};
00252     double d_A1GLO{};
00253     double d_A0GLO{};
00254
00255     double d_SQRT_A_ALMANAC{};
00256     double d_A1_ALMANAC{};
00257     double d_A0_ALMANAC{};
00258     double d_OMEGA0_ALMANAC{};
00259     double d_E_ALMANAC{};
00260     double d_DELTA_I{};
00261     double d_TOA{};
00262     double d_OMEGA_DOT_ALMANAC{};
00263     double d_OMEGA_ALMANAC{};
00264     double d_M0_ALMANAC{};
00265     int32_t almanac_WN{};
00266     double d_toa2{};
00267
00268     // System flags for data processing
00269     bool flag_eph_valid{};
00270     bool flag_utc_model_valid{};
00271     bool flag_iono_valid{};
00272     bool flag_dl_sf1{};
00273     bool flag_dl_sf2{};
00274     bool flag_dl_sf3{};
00275     bool flag_dl_sf4{};
00276     bool flag_dl_sf5{};
00277     bool flag_new_SOW_available{};
00278     bool flag_crc_test{};
00279     double d_previous_aode{};
00280
00281     // bool flag_dl_sf5_p7{}; // D1 NAV Message, Subframe 5, Page 09 decoded indicator
00282     // bool flag_dl_sf5_p8{}; // D1 NAV Message, Subframe 5, Page 09 decoded indicator
00283     bool flag_dl_sf5_p9{}; // D1 NAV Message, Subframe 5, Page 09 decoded indicator
00284     bool flag_dl_sf5_p10{}; // D1 NAV Message, Subframe 5, Page 10 decoded indicator
00285
00286     bool flag_sf1_p1{}; // D2 NAV Message, Subframe 1, Page 1 decoded indicator
00287     bool flag_sf1_p2{}; // D2 NAV Message, Subframe 1, Page 2 decoded indicator
00288     bool flag_sf1_p3{}; // D2 NAV Message, Subframe 1, Page 3 decoded indicator
00289     bool flag_sf1_p4{}; // D2 NAV Message, Subframe 1, Page 4 decoded indicator
00290     bool flag_sf1_p5{}; // D2 NAV Message, Subframe 1, Page 5 decoded indicator
00291     bool flag_sf1_p6{}; // D2 NAV Message, Subframe 1, Page 6 decoded indicator
00292     bool flag_sf1_p7{}; // D2 NAV Message, Subframe 1, Page 7 decoded indicator
00293     bool flag_sf1_p8{}; // D2 NAV Message, Subframe 1, Page 8 decoded indicator
00294     bool flag_sf1_p9{}; // D2 NAV Message, Subframe 1, Page 9 decoded indicator
00295     bool flag_sf1_p10{}; // D2 NAV Message, Subframe 1, Page 10 decoded indicator
00296 };
00297
00298
00299 /** \} */
00300 /** \} */
00301 #endif // GNSS_SDR_BEIDOU_DNAV_NAVIGATION_MESSAGE_H

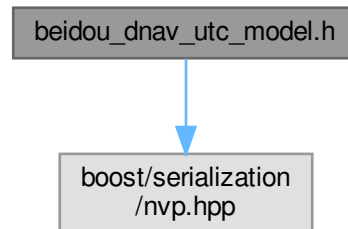
```

## 11.687 beidou\_dnav\_utc\_model.h File Reference

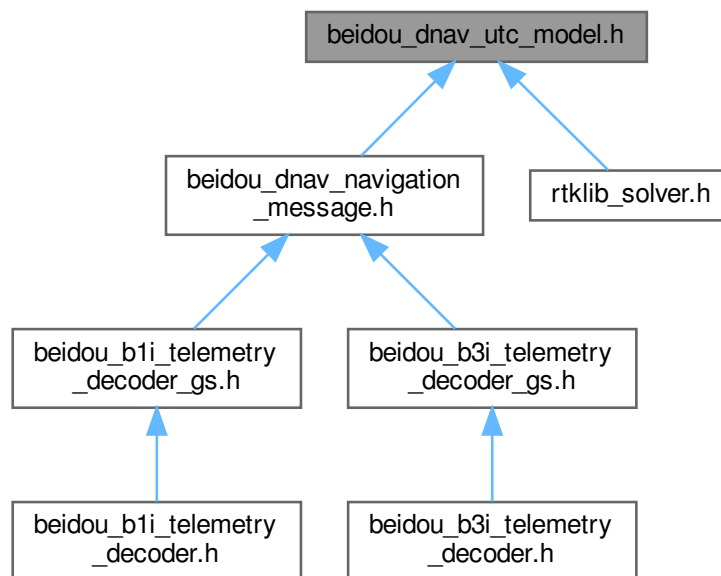
Interface of a BeiDou UTC MODEL storage.

```
#include <boost/serialization/nvp.hpp>
```

Include dependency graph for beidou\_dnav\_utc\_model.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Beidou\\_Dnav\\_Utc\\_Model](#)

*This class is a storage for the BeiDou DNAV UTC Model.*

## 11.687.1 Detailed Description

Interface of a BeiDou UTC MODEL storage.

## Author

Damian Miralles, 2018. [dmiralles2009@gmail.com](mailto:dmiralles2009@gmail.com)

Sergi Segura, 2018. [sergi.segura.munoz\(at\)gmail.com](mailto:sergi.segura.munoz(at)gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [beidou\\_dnav\\_utc\\_model.h](#).

---

## 11.688 beidou\_dnav\_utc\_model.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file beidou_dnav_utc_model.h
00003  * \brief Interface of a BeiDou UTC MODEL storage
00004  * \author Damian Miralles, 2018. dmiralles2009@gmail.com
00005  * \author Sergi Segura, 2018. sergi.segura.munoz(at)gmail.com
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018
00019 #ifndef GNSS_SDR_BEIDOU_DNAV_UTC_MODEL_H
00020 #define GNSS_SDR_BEIDOU_DNAV_UTC_MODEL_H
00021
00022 #include <boost/serialization/nvp.hpp>
00023
00024 /** \addtogroup Core
00025  * \{ */
00026 /** \addtogroup System_Parameters
00027  * \{ */
00028
00029
00030 /*!
00031 * \brief This class is a storage for the BeiDou DNAV UTC Model.
00032 * \details Implementation follows the interface described in the Open Service Signal (Version 2.1)
00033 *
00034 */
00035 class Beidou_Dnav_Utc_Model
00036 {
00037 public:
00038     Beidou_Dnav_Utc_Model() = default;
00039
00040     // BeiDou UTC parameters
00041     double AO_UTC{};    //!< BDT clock bias relative to UTC [s]
00042     double A1_UTC{};    //!< BDT clock rate relative to UTC [s/s]
00043     int DeltaT_LS{};    //!< Delta time due to leap seconds before the new leap second effective
00044     int WN_LSF{};       //!< Week number of the new leap second
00045     int DN{};           //!< Day number of week of the new leap second
00046     double DeltaT_LSF{}; //!< Delta time due to leap seconds after the new leap second effective [s]
00047
00048     // BeiDou to GPS time corrections
00049     double AO_GPS{};    //!< BDT clock bias relative to GPS time [s]
00050     double A1_GPS{};    //!< BDT clock rate relative to GPS time [s/s]
00051
00052     // BeiDou to Galileo time corrections
00053     double AO_GAL{};    //!< BDT clock bias relative to GAL time [s]
00054     double A1_GAL{};    //!< BDT clock rate relative to GAL time [s/s]
00055
00056     // BeiDou to GLONASS time corrections
00057     double AO_GLO{};    //!< BDT clock bias relative to GLO time [s]
00058     double A1_GLO{};    //!< BDT clock rate relative to GLO time [s/s]
00059
00060     bool valid{};
00061
00062     template <class Archive>
00063     /*
00064     * \brief Serialize is a boost standard method to be called by the boost XML
00065     * serialization. Here is used to save the ephemeris data on disk file.
00066     */
00067     inline void serialize(Archive& archive, const unsigned int version)
00068     {
00069         if (version)
00070         {

```

```

00071         };
00072         archive& BOOST_SERIALIZATION_NVP (A1_UTC);
00073         archive& BOOST_SERIALIZATION_NVP (A0_UTC);
00074         archive& BOOST_SERIALIZATION_NVP (DeltaT_LS);
00075         archive& BOOST_SERIALIZATION_NVP (WN_LSF);
00076         archive& BOOST_SERIALIZATION_NVP (DN);
00077         archive& BOOST_SERIALIZATION_NVP (DeltaT_LSF);
00078         archive& BOOST_SERIALIZATION_NVP (A0_GPS);
00079         archive& BOOST_SERIALIZATION_NVP (A1_GPS);
00080         archive& BOOST_SERIALIZATION_NVP (A0_GAL);
00081         archive& BOOST_SERIALIZATION_NVP (A1_GAL);
00082         archive& BOOST_SERIALIZATION_NVP (A0_GLO);
00083         archive& BOOST_SERIALIZATION_NVP (A1_GLO);
00084         archive& BOOST_SERIALIZATION_NVP (valid);
00085     }
00086 };
00087
00088
00089 /** \} */
00090 /** \} */
00091 #endif // GNSS_SDR_BEIDOU_DNAV_UTC_MODEL_H

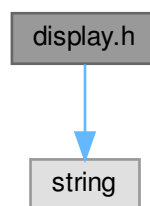
```

## 11.689 display.h File Reference

Defines useful display constants.

```
#include <string>
```

Include dependency graph for display.h:



### Macros

- #define `DISPLAY_COLORS` 1

### Variables

- const std::string `TEXT_RESET` = "\033[0m"
- const std::string `TEXT_BLACK` = "\033[30m"
- const std::string `TEXT_RED` = "\033[31m"
- const std::string `TEXT_GREEN` = "\033[32m"
- const std::string `TEXT_YELLOW` = "\033[33m"
- const std::string `TEXT_BLUE` = "\033[34m"
- const std::string `TEXT_MAGENTA` = "\033[35m"
- const std::string `TEXT_CYAN` = "\033[36m"
- const std::string `TEXT_WHITE` = "\033[37m"
- const std::string `TEXT_BOLD_BLACK` = "\033[1m\033[30m"
- const std::string `TEXT_BOLD_RED` = "\033[1m\033[31m"
- const std::string `TEXT_BOLD_GREEN` = "\033[1m\033[32m"
- const std::string `TEXT_BOLD_YELLOW` = "\033[1m\033[33m"
- const std::string `TEXT_BOLD_BLUE` = "\033[1m\033[34m"
- const std::string `TEXT_BOLD_MAGENTA` = "\033[1m\033[35m"

- `const std::string TEXT_BOLD_CYAN = "\033[1m\033[36m"`
- `const std::string TEXT_BOLD_WHITE = "\033[1m\033[37m"`

### 11.689.1 Detailed Description

Defines useful display constants.

Author

Antonio Ramos, 2018. antonio.ramos(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [display.h](#).

---

## 11.690 display.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file display.h
00003  * \brief Defines useful display constants
00004  * \author Antonio Ramos, 2018. antonio.ramos(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_DISPLAY_H
00018 #define GNSS_SDR_DISPLAY_H
00019
00020 #include <string>
00021
00022 /** \addtogroup Core
00023  * \{ */
00024 /** \addtogroup System_Parameters
00025  * \{ */
00026
00027
00028 #ifndef NO_DISPLAY_COLORS
00029 #define DISPLAY_COLORS 1
00030 #endif
00031
00032 #ifdef DISPLAY_COLORS
00033
00034
00035 const std::string TEXT_RESET = "\033[0m";
00036 const std::string TEXT_BLACK = "\033[30m";
00037 const std::string TEXT_RED = "\033[31m";
00038 const std::string TEXT_GREEN = "\033[32m";
00039 const std::string TEXT_YELLOW = "\033[33m";
00040 const std::string TEXT_BLUE = "\033[34m";
00041 const std::string TEXT_MAGENTA = "\033[35m";
00042 const std::string TEXT_CYAN = "\033[36m";
00043 const std::string TEXT_WHITE = "\033[37m";
00044 const std::string TEXT_BOLD_BLACK = "\033[1m\033[30m";
00045 const std::string TEXT_BOLD_RED = "\033[1m\033[31m";
00046 const std::string TEXT_BOLD_GREEN = "\033[1m\033[32m";
00047 const std::string TEXT_BOLD_YELLOW = "\033[1m\033[33m";
00048 const std::string TEXT_BOLD_BLUE = "\033[1m\033[34m";
00049 const std::string TEXT_BOLD_MAGENTA = "\033[1m\033[35m";
00050 const std::string TEXT_BOLD_CYAN = "\033[1m\033[36m";
00051 const std::string TEXT_BOLD_WHITE = "\033[1m\033[37m";
00052
00053 #else
00054
00055 const std::string TEXT_RESET = "";
00056 const std::string TEXT_BLACK = "";
00057 const std::string TEXT_RED = "";
00058 const std::string TEXT_GREEN = "";
00059 const std::string TEXT_YELLOW = "";
00060 const std::string TEXT_BLUE = "";
00061 const std::string TEXT_MAGENTA = "";
00062 const std::string TEXT_CYAN = "";

```

```

00063 const std::string TEXT_WHITE = "";
00064 const std::string TEXT_BOLD_BLACK = "";
00065 const std::string TEXT_BOLD_RED = "";
00066 const std::string TEXT_BOLD_GREEN = "";
00067 const std::string TEXT_BOLD_YELLOW = "";
00068 const std::string TEXT_BOLD_BLUE = "";
00069 const std::string TEXT_BOLD_MAGENTA = "";
00070 const std::string TEXT_BOLD_CYAN = "";
00071 const std::string TEXT_BOLD_WHITE = "";
00072
00073 #endif // DISPLAY_COLORS
00074
00075
00076 /** \} */
00077 /** \} */
00078 #endif // GNSS_SDR_DISPLAY_H

```

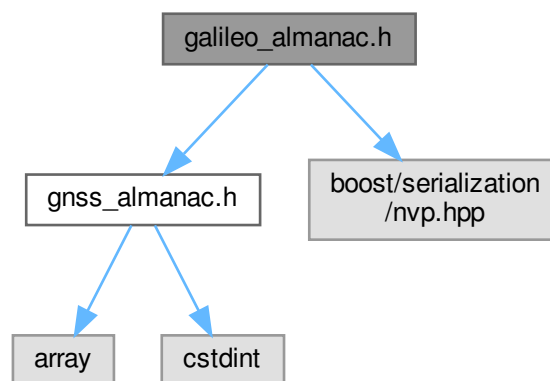
## 11.691 galileo\_almanac.h File Reference

Interface of a Galileo ALMANAC storage.

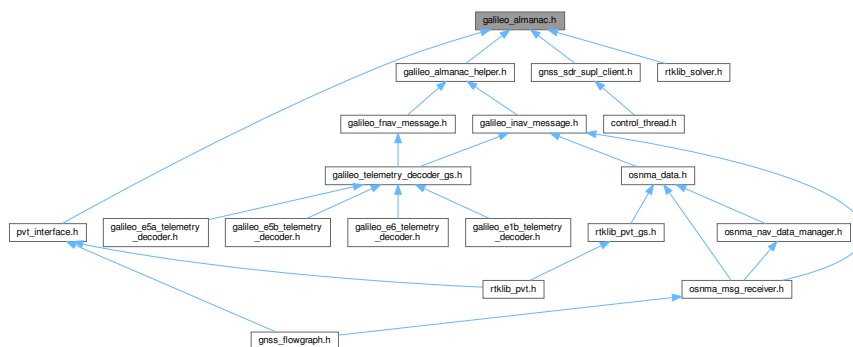
```
#include "gnss_almanac.h"
```

```
#include <boost/serialization/nvp.hpp>
```

Include dependency graph for galileo\_almanac.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Galileo\\_Almanac](#)

*This class is a storage for the Galileo SV ALMANAC data.*

## 11.691.1 Detailed Description

Interface of a Galileo ALMANAC storage.

### Author

Carles Fernandez, 2018. cfernandez(at)cttc.cat

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [galileo\\_almanac.h](#).

---

## 11.692 galileo\_almanac.h

[Go to the documentation of this file.](#)

```
00001  /*!
00002  * \file galileo_almanac.h
00003  * \brief Interface of a Galileo ALMANAC storage
00004  * \author Carles Fernandez, 2018. cfernandez(at)cttc.cat
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_GALILEO_ALMANAC_H
00019 #define GNSS_SDR_GALILEO_ALMANAC_H
00020
00021 #include "gnss_almanac.h"
00022 #include <boost/serialization/nvp.hpp>
00023
00024 /** \addtogroup Core
00025  * \{ */
00026 /** \addtogroup System_Parameters
00027  * \{ */
00028
00029
00030 /*!
00031 * \brief This class is a storage for the Galileo SV ALMANAC data
00032 */
00033 class Galileo_Almanac : public Gnss_Almanac
00034 {
00035 public:
00036     /*!
00037      * Default constructor
00038      */
00039     Galileo_Almanac()
00040     {
00041         this->System = 'E';
00042     };
00043
00044     int32_t IODa{};
00045     int32_t E5b_HS{};
00046     int32_t E1B_HS{};
00047     int32_t E5a_HS{};
00048
00049     template <class Archive>
00050
00051     void serialize(Archive& ar, const unsigned int version)
00052     {
00053         if (version)
00054         {
00055             ar& BOOST_SERIALIZATION_NVP(PRN);
00056             ar& BOOST_SERIALIZATION_NVP(delta_i);
00057             ar& BOOST_SERIALIZATION_NVP(toa);
00058             ar& BOOST_SERIALIZATION_NVP(WNa);
00059         }
00060     }
00061 }
```



```

00060         ar& BOOST_SERIALIZATION_NVP (M_0);
00061         ar& BOOST_SERIALIZATION_NVP (ecc);
00062         ar& BOOST_SERIALIZATION_NVP (sqrtA);
00063         ar& BOOST_SERIALIZATION_NVP (OMEGA_0);
00064         ar& BOOST_SERIALIZATION_NVP (omega);
00065         ar& BOOST_SERIALIZATION_NVP (OMEGA_dot);
00066         ar& BOOST_SERIALIZATION_NVP (af0);
00067         ar& BOOST_SERIALIZATION_NVP (af1);
00068         ar& BOOST_SERIALIZATION_NVP (IODa);
00069         ar& BOOST_SERIALIZATION_NVP (E5b_HS);
00070         ar& BOOST_SERIALIZATION_NVP (E1B_HS);
00071         ar& BOOST_SERIALIZATION_NVP (E5a_HS);
00072     }
00073 };
00074
00075
00076 /** \} */
00077 /** \} */
00078 #endif // GNSS_SDR_GALILEO_ALMANAC_H

```

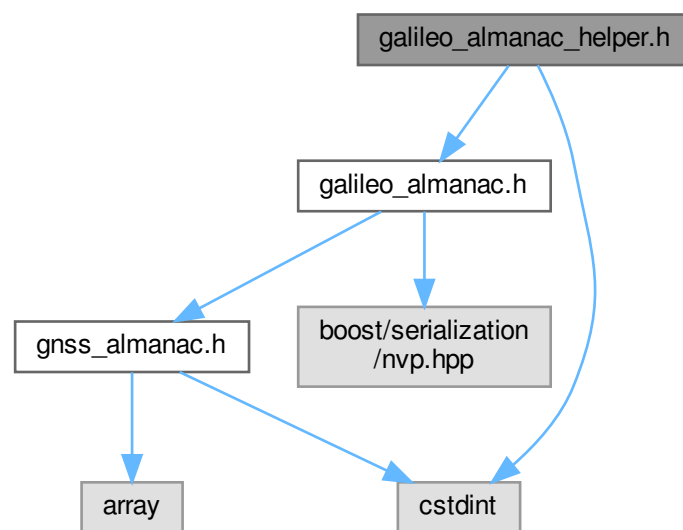
## 11.693 galileo\_almanac\_helper.h File Reference

Interface of a Galileo ALMANAC storage helper.

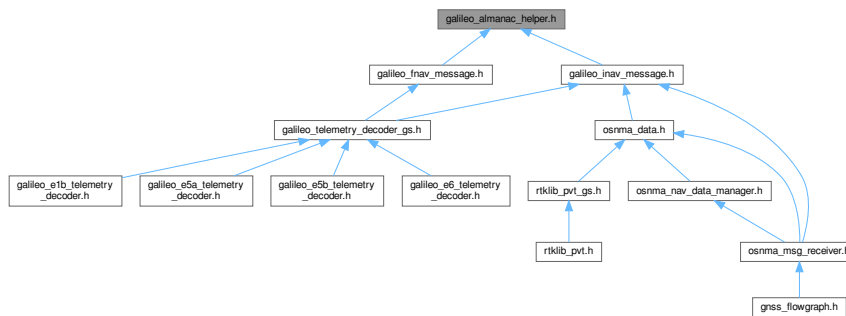
```
#include "galileo_almanac.h"
```

```
#include <cstdint>
```

Include dependency graph for galileo\_almanac\_helper.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Galileo\\_Almanac\\_Helper](#)

*This class is a storage for the GALILEO ALMANAC data as described in GALILEO ICD.*

## 11.693.1 Detailed Description

Interface of a Galileo ALMANAC storage helper.

### Author

Javier Arribas, 2013. jarribas(at)cttc.es

### 11.693.1.1 autotoc\_md688

#### Author

Mara Branzanti 2013. mara.branzanti(at)gmail.com

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [galileo\\_almanac\\_helper.h](#).

## 11.694 galileo\_almanac\_helper.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file galileo_almanac_helper.h
00003  * \brief Interface of a Galileo ALMANAC storage helper
00004  * \author Javier Arribas, 2013. jarribas(at)cttc.es
00005  * \author Mara Branzanti 2013. mara.branzanti(at)gmail.com
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_GALILEO_ALMANAC_HELPER_H
00018 #define GNSS_SDR_GALILEO_ALMANAC_HELPER_H
00019
00020 #include "galileo_almanac.h"
00021 #include <stdint>
00022
00023 /** \addtogroup Core
00024  * \{ */
00025 /** \addtogroup System_Parameters
00026  * \{ */

```

```

00027
00028
00029 /*!
00030  * \brief This class is a storage for the GALILEO ALMANAC data as described in GALILEO ICD
00031  *
00032  * See https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo\_OS\_SIS\_ICD\_v2.0.pdf
00033  * paragraph 5.1.10
00034  */
00034 class Galileo_Almanac_Helper
00035 {
00036 public:
00037     Galileo_Almanac_Helper() = default; ///< Default constructor
00038
00039     Galileo_Almanac get_almanac(int i) const;
00040
00041     // Word type 7: Almanac for SVID1 (1/2), almanac reference time and almanac reference week number
00042     int32_t IOD_a_7{};
00043     int32_t WN_a_7{};
00044     int32_t t0a_7{};
00045     int32_t SVID1_7{};
00046     double DELTA_A_7{};
00047     double e_7{};
00048     double omega_7{};
00049     double delta_i_7{};
00050     double Omega0_7{};
00051     double Omega_dot_7{};
00052     double M0_7{};
00053
00054     // Word type 8: Almanac for SVID1 (2/2) and SVID2 (1/2)
00055     int32_t IOD_a_8{};
00056     double af0_8{};
00057     double af1_8{};
00058     int32_t E5b_HS_8{};
00059     int32_t E1B_HS_8{};
00060     int32_t E5a_HS_8{};
00061     int32_t SVID2_8{};
00062     double DELTA_A_8{};
00063     double e_8{};
00064     double omega_8{};
00065     double delta_i_8{};
00066     double Omega0_8{};
00067     double Omega_dot_8{};
00068
00069     // Word type 9: Almanac for SVID2 (2/2) and SVID3 (1/2)
00070     int32_t IOD_a_9{};
00071     int32_t WN_a_9{};
00072     int32_t t0a_9{};
00073     double M0_9{};
00074     double af0_9{};
00075     double af1_9{};
00076     int32_t E5b_HS_9{};
00077     int32_t E1B_HS_9{};
00078     int32_t E5a_HS_9{};
00079     int32_t SVID3_9{};
00080     double DELTA_A_9{};
00081     double e_9{};
00082     double omega_9{};
00083     double delta_i_9{};
00084
00085     // Word type 10: Almanac for SVID3 (2/2)
00086     int32_t IOD_a_10{};
00087     double Omega0_10{};
00088     double Omega_dot_10{};
00089     double M0_10{};
00090     double af0_10{};
00091     double af1_10{};
00092     int32_t E5b_HS_10{};
00093     int32_t E1B_HS_10{};
00094     int32_t E5a_HS_10{};
00095 };
00096
00097
00098 /** \} */
00099 /** \} */
00100 #endif // GNSS_SDR_GALILEO_ALMANAC_HELPER_H

```

## 11.695 Galileo\_CNAV.h File Reference

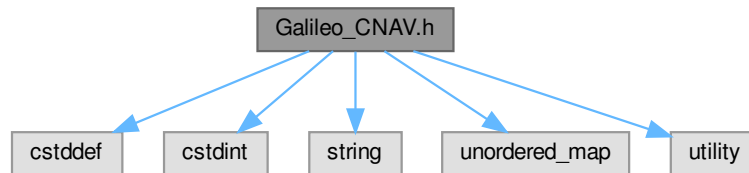
Galileo CNAV message constants. Data from: Galileo High Accuracy Service Signal-In-Space Interface Control Document (HAS SIS ICD) Issue 1.0, May 2022.

```

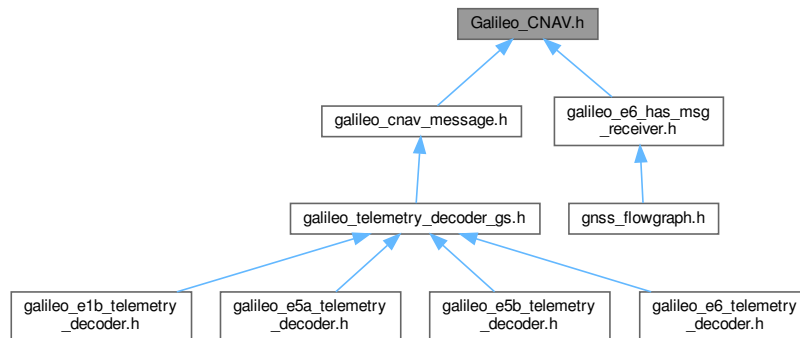
#include <cstdint>
#include <cstdint>

```

```
#include <string>
#include <unordered_map>
#include <utility>
Include dependency graph for Galileo_CNAV.h:
```



This graph shows which files directly or indirectly include this file:



## Functions

- `const std::pair< int32_t, int32_t > GALILEO_HAS_STATUS ({1, 2})`
- `const std::pair< int32_t, int32_t > GALILEO_HAS_RESERVED ({3, 2})`
- `const std::pair< int32_t, int32_t > GALILEO_HAS_MESSAGE_TYPE ({5, 2})`
- `const std::pair< int32_t, int32_t > GALILEO_HAS_MESSAGE_ID ({7, 5})`
- `const std::pair< int32_t, int32_t > GALILEO_HAS_MESSAGE_SIZE ({12, 5})`
- `const std::pair< int32_t, int32_t > GALILEO_HAS_MESSAGE_PAGE_ID ({17, 8})`
- `const std::pair< int32_t, int32_t > GALILEO_MT1_HEADER_TOH ({1, 12})`
- `const std::pair< int32_t, int32_t > GALILEO_MT1_HEADER_MASK_FLAG ({13, 1})`
- `const std::pair< int32_t, int32_t > GALILEO_MT1_HEADER_ORBIT_CORRECTION_FLAG ({14, 1})`
- `const std::pair< int32_t, int32_t > GALILEO_MT1_HEADER_CLOCK_FULLSET_FLAG ({15, 1})`
- `const std::pair< int32_t, int32_t > GALILEO_MT1_HEADER_CLOCK_SUBSET_FLAG ({16, 1})`
- `const std::pair< int32_t, int32_t > GALILEO_MT1_HEADER_CODE_BIAS_FLAG ({17, 1})`
- `const std::pair< int32_t, int32_t > GALILEO_MT1_HEADER_PHASE_BIAS_FLAG ({18, 1})`
- `const std::pair< int32_t, int32_t > GALILEO_MT1_HEADER_RESERVED ({19, 4})`
- `const std::pair< int32_t, int32_t > GALILEO_MT1_HEADER_MASK_ID ({23, 5})`
- `const std::pair< int32_t, int32_t > GALILEO_MT1_HEADER_IOD_SET_ID ({28, 5})`

## Variables

- constexpr size\_t [HAS\\_MSG\\_NSYS\\_LENGTH](#) = 4
- constexpr size\_t [HAS\\_MSG\\_ID\\_MASK\\_LENGTH](#) = 4
- constexpr size\_t [HAS\\_MSG\\_SATELLITE\\_MASK\\_LENGTH](#) = 40
- constexpr size\_t [HAS\\_MSG\\_SIGNAL\\_MASK\\_LENGTH](#) = 16
- constexpr size\_t [HAS\\_MSG\\_NAV\\_MESSAGE\\_LENGTH](#) = 3
- constexpr size\_t [HAS\\_MSG\\_MASK\\_RESERVED\\_LENGTH](#) = 6
- constexpr size\_t [HAS\\_MSG\\_VALIDITY\\_INDEX\\_LENGTH](#) = 4
- constexpr size\_t [HAS\\_MSG\\_IOD\\_GPS\\_LENGTH](#) = 8
- constexpr size\_t [HAS\\_MSG\\_IOD\\_GAL\\_LENGTH](#) = 10
- constexpr size\_t [HAS\\_MSG\\_DELTA\\_RADIAL\\_LENGTH](#) = 13
- constexpr size\_t [HAS\\_MSG\\_DELTA\\_IN\\_TRACK\\_LENGTH](#) = 12
- constexpr size\_t [HAS\\_MSG\\_DELTA\\_CROSS\\_TRACK\\_LENGTH](#) = 12
- constexpr size\_t [HAS\\_MSG\\_DELTA\\_CLOCK\\_MULTIPLIER\\_LENGTH](#) = 2
- constexpr size\_t [HAS\\_MSG\\_DELTA\\_CLOCK\\_CORRECTION\\_LENGTH](#) = 13
- constexpr size\_t [HAS\\_MSG\\_NSYS\\_SUB\\_LENGTH](#) = 4
- constexpr size\_t [HAS\\_MSG\\_ID\\_CLOCK\\_SUBSET\\_LENGTH](#) = 4
- constexpr size\_t [HAS\\_MSG\\_DELTA\\_CLOCK\\_MULTIPLIER\\_SUBSET\\_LENGTH](#) = 2
- constexpr size\_t [HAS\\_MSG\\_DELTA\\_CLOCK\\_CORRECTION\\_SUBSET\\_LENGTH](#) = 13
- constexpr size\_t [HAS\\_MSG\\_CODE\\_BIAS\\_LENGTH](#) = 11
- constexpr size\_t [HAS\\_MSG\\_PHASE\\_BIAS\\_LENGTH](#) = 11
- constexpr size\_t [HAS\\_MSG\\_PHASE\\_DISCONTINUITY\\_INDICATOR\\_LENGTH](#) = 2
- constexpr uint64\_t [MAX\\_SECONDS\\_REMEMBERING\\_MID](#) = 150
- constexpr int32\_t [GALILEO\\_CNAV\\_SYMBOLS\\_PER\\_PAGE](#) = 1000
- constexpr int32\_t [GALILEO\\_CNAV\\_PREAMBLE\\_PERIOD\\_SYMBOLS](#) = 1000
- constexpr int32\_t [GALILEO\\_CNAV\\_PAGE\\_MS](#) = 1
- constexpr int32\_t [GALILEO\\_CNAV\\_INTERLEAVER\\_ROWS](#) = 8
- constexpr int32\_t [GALILEO\\_CNAV\\_INTERLEAVER\\_COLS](#) = 123
- constexpr int32\_t [GALILEO\\_CNAV\\_TELEMETRY\\_RATE\\_BITS\\_SECOND](#) = 1000
- constexpr int32\_t [GALILEO\\_CNAV\\_HAS\\_PAGE\\_DATA\\_BITS](#) = 448
- constexpr int32\_t [GALILEO\\_CNAV\\_PAGE\\_RESERVED\\_BITS](#) = 14
- constexpr int32\_t [GALILEO\\_CNAV\\_BYTES\\_FOR\\_CRC](#) = 58
- constexpr int32\_t [GALILEO\\_CNAV\\_CRC\\_LENGTH](#) = 24
- constexpr int32\_t [GALILEO\\_CNAV\\_MESSAGE\\_BITS\\_PER\\_PAGE](#) = 424
- constexpr int32\_t [GALILEO\\_CNAV\\_PAGE\\_HEADER\\_BITS](#) = 24
- constexpr int32\_t [GALILEO\\_CNAV\\_PREAMBLE\\_LENGTH\\_BITS](#) = 16
- constexpr int32\_t [GALILEO\\_CNAV\\_MAX\\_NUMBER\\_SYMBOLS\\_ENCODED\\_BLOCK](#) = 255
- constexpr int32\_t [GALILEO\\_CNAV\\_MT1\\_HEADER\\_BITS](#) = 32
- constexpr int32\_t [GALILEO\\_CNAV\\_OCTETS\\_IN\\_SUBPAGE](#) = 53
- constexpr int32\_t [GALILEO\\_CNAV\\_INFORMATION\\_VECTOR\\_LENGTH](#) = 32
- constexpr int32\_t [GALILEO\\_CNAV\\_BITS\\_FOR\\_CRC](#) = [GALILEO\\_CNAV\\_HAS\\_PAGE\\_DATA\\_BITS](#) + [GALILEO\\_CNAV\\_PAGE\\_RESERVED\\_BITS](#)
- constexpr int32\_t [HAS\\_MSG\\_NUMBER\\_MASK\\_IDS](#) = 32
- constexpr int32\_t [HAS\\_MSG\\_NUMBER\\_GNSS\\_IDS](#) = 16
- constexpr int32\_t [HAS\\_MSG\\_NUMBER\\_MESSAGE\\_IDS](#) = 32
- constexpr int32\_t [HAS\\_MSG\\_NUMBER\\_SATELLITE\\_IDS](#) = 40
- constexpr int32\_t [HAS\\_MSG\\_NUMBER\\_SIGNAL\\_MASKS](#) = 16
- constexpr float [HAS\\_MSG\\_DELTA\\_RADIAL\\_SCALE\\_FACTOR](#) = 0.0025
- constexpr float [HAS\\_MSG\\_DELTA\\_IN\\_TRACK\\_SCALE\\_FACTOR](#) = 0.008
- constexpr float [HAS\\_MSG\\_DELTA\\_CROSS\\_TRACK\\_SCALE\\_FACTOR](#) = 0.008
- constexpr float [HAS\\_MSG\\_DELTA\\_CLOCK\\_SCALE\\_FACTOR](#) = 0.0025
- constexpr float [HAS\\_MSG\\_CODE\\_BIAS\\_SCALE\\_FACTOR](#) = 0.02
- constexpr float [HAS\\_MSG\\_PHASE\\_BIAS\\_SCALE\\_FACTOR](#) = 0.01
- constexpr uint16\_t [HAS\\_MSG\\_NUMBER\\_MAX\\_TOH](#) = 3599

- constexpr uint8\_t [HAS\\_MSG\\_GPS\\_SYSTEM](#) = 0
- constexpr uint8\_t [HAS\\_MSG\\_GALILEO\\_SYSTEM](#) = 2
- constexpr uint8\_t [HAS\\_MSG\\_WRONG\\_SYSTEM](#) = 255
- constexpr char [GALILEO\\_CNAV\\_PREAMBLE](#) [17] = "1011011101110000"
- const std::unordered\_map< uint8\_t, std::unordered\_map< uint8\_t, std::string > > [HAS\\_SIGNAL\\_INDEX\\_TABLE](#)
- const std::unordered\_map< uint8\_t, uint16\_t > [HAS\\_VALIDITY\\_INTERVALS](#)

### 11.695.1 Detailed Description

Galileo CNAV message constants. Data from: Galileo High Accuracy Service Signal-In-Space Interface Control Document (HAS SIS ICD) Issue 1.0, May 2022.

Author

Carles Fernandez-Prades, 2020-2022. cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [Galileo\\_CNAV.h](#).

---

## 11.696 Galileo\_CNAV.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file Galileo_CNAV.h
00003  * \brief Galileo CNAV message constants. Data from:
00004  * Galileo High Accuracy Service Signal-In-Space Interface Control Document
00005  * (HAS SIS ICD) Issue 1.0, May 2022
00006  * \author Carles Fernandez-Prades, 2020-2022. cfernandez(at)cttc.es
00007  *
00008  *
00009  * -----
00010  *
00011  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00012  * This file is part of GNSS-SDR.
00013  *
00014  * Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors)
00015  * SPDX-License-Identifier: GPL-3.0-or-later
00016  *
00017  * -----
00018  */
00019
00020 #ifndef GNSS_SDR_GALILEO_CNAV_H
00021 #define GNSS_SDR_GALILEO_CNAV_H
00022
00023 #include <cstdint>
00024 #include <string>
00025 #include <unordered_map>
00026 #include <utility>
00027
00028 /** \addtogroup Core
00029  * \{ */
00030 /** \addtogroup System_Parameters
00031  * \{ */
00032
00033 // Galileo HAS message field lengths
00034
00035 constexpr size_t HAS_MSG_NSYS_LENGTH = 4; // HAS SIS ICD 1.0 Table 15
00036 constexpr size_t HAS_MSG_ID_MASK_LENGTH = 4; // HAS SIS ICD 1.0 Table 16
00037 constexpr size_t HAS_MSG_SATELLITE_MASK_LENGTH = 40; // HAS SIS ICD 1.0 Table 16
00038 constexpr size_t HAS_MSG_SIGNAL_MASK_LENGTH = 16; // HAS SIS ICD 1.0 Table 16
00039 constexpr size_t HAS_MSG_NAV_MESSAGE_LENGTH = 3; // HAS SIS ICD 1.0 Table 16
00040 constexpr size_t HAS_MSG_MASK_RESERVED_LENGTH = 6; // HAS SIS ICD 1.0 Table 15
00041 constexpr size_t HAS_MSG_VALIDITY_INDEX_LENGTH = 4; // HAS SIS ICD 1.0 Table 22
00042 constexpr size_t HAS_MSG_IOD_GPS_LENGTH = 8; // HAS SIS ICD 1.0 Table 26
00043 constexpr size_t HAS_MSG_IOD_GAL_LENGTH = 10; // HAS SIS ICD 1.0 Table 26
00044 constexpr size_t HAS_MSG_DELTA_RADIAL_LENGTH = 13; // HAS SIS ICD 1.0 Table 25
00045 constexpr size_t HAS_MSG_DELTA_IN_TRACK_LENGTH = 12; // HAS SIS ICD 1.0 Table 25
00046 constexpr size_t HAS_MSG_DELTA_CROSS_TRACK_LENGTH = 12; // HAS SIS ICD 1.0 Table 25
00047 constexpr size_t HAS_MSG_DELTA_CLOCK_MULTIPLIER_LENGTH = 2; // HAS SIS ICD 1.0 Table 28
00048 constexpr size_t HAS_MSG_DELTA_CLOCK_CORRECTION_LENGTH = 13; // HAS SIS ICD 1.0 Table 31
00049 constexpr size_t HAS_MSG_NSYS_SUB_LENGTH = 4; // HAS SIS ICD 1.0 Table 32
00050 constexpr size_t HAS_MSG_ID_CLOCK_SUBSET_LENGTH = 4; // HAS SIS ICD 1.0 Table 32
00051 constexpr size_t HAS_MSG_DELTA_CLOCK_MULTIPLIER_SUBSET_LENGTH = 2; // HAS SIS ICD 1.0 Table 33
00052 constexpr size_t HAS_MSG_DELTA_CLOCK_CORRECTION_SUBSET_LENGTH = 13; // HAS SIS ICD 1.0 Table 34
00053 constexpr size_t HAS_MSG_CODE_BIAS_LENGTH = 11; // HAS SIS ICD 1.0 Table 37
00054 constexpr size_t HAS_MSG_PHASE_BIAS_LENGTH = 11; // HAS SIS ICD 1.0 Table 40

```

```

00055 constexpr size_t HAS_MSG_PHASE_DISCONTINUITY_INDICATOR_LENGTH = 2; // HAS SIS ICD 1.0 Table 40
00056
00057 constexpr uint64_t MAX_SECONDS_REMEMBERING_MID = 150; // HAS SIS ICD 1.0 Section 6.4.1 HAS Message
Completion Time-out
00058
00059 // Galileo CNAV message structure
00060 constexpr int32_t GALILEO_CNAV_SYMBOLS_PER_PAGE = 1000; // Total number of symbols per HAS page
including the sync pattern
00061 constexpr int32_t GALILEO_CNAV_PREAMBLE_PERIOD_SYMBOLS = 1000;
00062 constexpr int32_t GALILEO_CNAV_PAGE_MS = 1; // Duration in ms of a CNAV
page
00063 constexpr int32_t GALILEO_CNAV_INTERLEAVER_ROWS = 8; // HAS SIS ICD 1.0 Table 4
00064 constexpr int32_t GALILEO_CNAV_INTERLEAVER_COLS = 123; // HAS SIS ICD 1.0 Table 4
00065 constexpr int32_t GALILEO_CNAV_TELEMETRY_RATE_BITS_SECOND = 1000; // bps
00066 constexpr int32_t GALILEO_CNAV_HAS_PAGE_DATA_BITS = 448; // HAS SIS ICD 1.0 Table 5
00067 constexpr int32_t GALILEO_CNAV_PAGE_RESERVED_BITS = 14; // HAS SIS ICD 1.0 Table 5
00068 constexpr int32_t GALILEO_CNAV_BYTES_FOR_CRC = 58; // ceil(462 / 8)
00069 constexpr int32_t GALILEO_CNAV_CRC_LENGTH = 24; // HAS SIS ICD 1.0 Table 5
00070 constexpr int32_t GALILEO_CNAV_MESSAGE_BITS_PER_PAGE = 424; // HAS SIS ICD 1.0 Table 6
00071 constexpr int32_t GALILEO_CNAV_PAGE_HEADER_BITS = 24; // HAS SIS ICD 1.0 Table 6
00072 constexpr int32_t GALILEO_CNAV_PREAMBLE_LENGTH_BITS = 16; // HAS SIS ICD 1.0 Table 5
00073 constexpr int32_t GALILEO_CNAV_MAX_NUMBER_SYMBOLS_ENCODED_BLOCK = 255; // HAS SIS ICD 1.0 Section 6.2
Reed-Solomon Code
00074 constexpr int32_t GALILEO_CNAV_MT1_HEADER_BITS = 32; // HAS SIS ICD 1.0 Table 11
00075 constexpr int32_t GALILEO_CNAV_OCTETS_IN_SUBPAGE = 53; // HAS SIS ICD 1.0 Section 6.3
HAS Encoding and Transmission
00076 constexpr int32_t GALILEO_CNAV_INFORMATION_VECTOR_LENGTH = 32; // HAS SIS ICD 1.0 Section 6.2
Reed-Solomon Code
00077
00078 constexpr int32_t GALILEO_CNAV_BITS_FOR_CRC = GALILEO_CNAV_HAS_PAGE_DATA_BITS +
GALILEO_CNAV_PAGE_RESERVED_BITS; // 462
00079
00080 constexpr int32_t HAS_MSG_NUMBER_MASK_IDS = 32; // HAS SIS ICD 1.0 Table 13
00081 constexpr int32_t HAS_MSG_NUMBER_GNSS_IDS = 16; // HAS SIS ICD 1.0 Table 18
00082 constexpr int32_t HAS_MSG_NUMBER_MESSAGE_IDS = 32; // HAS SIS ICD 1.0 Table 8
00083 constexpr int32_t HAS_MSG_NUMBER_SATELLITE_IDS = 40; // HAS SIS ICD 1.0 Table 19
00084 constexpr int32_t HAS_MSG_NUMBER_SIGNAL_MASKS = 16; // HAS SIS ICD 1.0 Table 20
00085
00086 constexpr float HAS_MSG_DELTA_RADIAL_SCALE_FACTOR = 0.0025; // HAS SIS ICD 1.0 Table 25
00087 constexpr float HAS_MSG_DELTA_IN_TRACK_SCALE_FACTOR = 0.008; // HAS SIS ICD 1.0 Table 25
00088 constexpr float HAS_MSG_DELTA_CROSS_TRACK_SCALE_FACTOR = 0.008; // HAS SIS ICD 1.0 Table 25
00089 constexpr float HAS_MSG_DELTA_CLOCK_SCALE_FACTOR = 0.0025; // HAS SIS ICD 1.0 Table 31
00090 constexpr float HAS_MSG_CODE_BIAS_SCALE_FACTOR = 0.02; // HAS SIS ICD 1.0 Table 37
00091 constexpr float HAS_MSG_PHASE_BIAS_SCALE_FACTOR = 0.01; // HAS SIS ICD 1.0 Table 40
00092
00093 constexpr uint16_t HAS_MSG_NUMBER_MAX_TOH = 3599; // HAS SIS ICD 1.0 Table 13
00094
00095 constexpr uint8_t HAS_MSG_GPS_SYSTEM = 0; // HAS SIS ICD 1.0 Table 18
00096 constexpr uint8_t HAS_MSG_GALILEO_SYSTEM = 2; // HAS SIS ICD 1.0 Table 18
00097 constexpr uint8_t HAS_MSG_WRONG_SYSTEM = 255;
00098
00099 constexpr char GALILEO_CNAV_PREAMBLE[17] = "1011011101110000"; // HAS SIS ICD 1.0 Section 2.3.1
00100
00101 // HAS SIS ICD 1.0 Table 7
00102 const std::pair<int32_t, int32_t> GALILEO_HAS_STATUS({1, 2});
00103 const std::pair<int32_t, int32_t> GALILEO_HAS_RESERVED({3, 2});
00104 const std::pair<int32_t, int32_t> GALILEO_HAS_MESSAGE_TYPE({5, 2});
00105 const std::pair<int32_t, int32_t> GALILEO_HAS_MESSAGE_ID({7, 5});
00106 const std::pair<int32_t, int32_t> GALILEO_HAS_MESSAGE_SIZE({12, 5});
00107 const std::pair<int32_t, int32_t> GALILEO_HAS_MESSAGE_PAGE_ID({17, 8});
00108
00109 // HAS SIS ICD 1.0 Table 12
00110 const std::pair<int32_t, int32_t> GALILEO_MT1_HEADER_TOH({1, 12});
00111 const std::pair<int32_t, int32_t> GALILEO_MT1_HEADER_MASK_FLAG({13, 1});
00112 const std::pair<int32_t, int32_t> GALILEO_MT1_HEADER_ORBIT_CORRECTION_FLAG({14, 1});
00113 const std::pair<int32_t, int32_t> GALILEO_MT1_HEADER_CLOCK_FULLSET_FLAG({15, 1});
00114 const std::pair<int32_t, int32_t> GALILEO_MT1_HEADER_CLOCK_SUBSET_FLAG({16, 1});
00115 const std::pair<int32_t, int32_t> GALILEO_MT1_HEADER_CODE_BIAS_FLAG({17, 1});
00116 const std::pair<int32_t, int32_t> GALILEO_MT1_HEADER_PHASE_BIAS_FLAG({18, 1});
00117 const std::pair<int32_t, int32_t> GALILEO_MT1_HEADER_RESERVED({19, 4});
00118 const std::pair<int32_t, int32_t> GALILEO_MT1_HEADER_MASK_ID({23, 5});
00119 const std::pair<int32_t, int32_t> GALILEO_MT1_HEADER_IOD_SET_ID({28, 5});
00120
00121 // HAS SIS ICD v1.0 Table 20
00122 const std::unordered_map<uint8_t, std::unordered_map<uint8_t, std::string> HAS_SIGNAL_INDEX_TABLE = {
00123     {0, {
00124         {0, "L1 C/A"},
00125         {1, "Reserved"},
00126         {2, "Reserved"},
00127         {3, "L1C(D)"},
00128         {4, "L1C(P)"},
00129         {5, "L1C(D+P)"},
00130         {6, "L2 CM"},
00131         {7, "L2 CL"},
00132         {8, "L2 CM+CL"},
00133         {9, "L2 P"},
00134         {10, "Reserved"},

```

```

00135         {11, "L5 I"},
00136         {12, "L5 Q"},
00137         {13, "L5 I + L5 Q"},
00138         {14, "Reserved"},
00139         {15, "Reserved"},
00140     }},
00141     {2, {
00142         {0, "E1-B I/NAV OS"},
00143         {1, "E1-C"},
00144         {2, "E1-B + E1-C"},
00145         {3, "E5a-I F/NAV OS"},
00146         {4, "E5a-Q"},
00147         {5, "E5a-I+E5a-Q"},
00148         {6, "E5b-I I/NAV OS"},
00149         {7, "E5b-Q"},
00150         {8, "E5b-I+E5b-Q"},
00151         {9, "E5-I"},
00152         {10, "E5-Q"},
00153         {11, "E5-I + E5-Q"},
00154         {12, "E6-B C/NAV HAS"},
00155         {13, "E6-C"},
00156         {14, "E6-B + E6-C"},
00157         {15, "Reserved"},
00158     }}};
00159
00160 // HAS SIS ICD v1.0 Table 23
00161 const std::unordered_map<uint8_t, uint16_t> HAS_VALIDITY_INTERVALS = {
00162     {0, 5},
00163     {1, 10},
00164     {2, 15},
00165     {3, 20},
00166     {4, 30},
00167     {5, 60},
00168     {6, 90},
00169     {7, 120},
00170     {8, 180},
00171     {9, 240},
00172     {10, 300},
00173     {11, 600},
00174     {12, 900},
00175     {13, 1800},
00176     {14, 3600}};
00177
00178
00179 /** \} */
00180 /** \} */
00181 #endif // GNSS_SDR_GALILEO_CNAV_H

```

## 11.697 galileo\_cnav\_message.h File Reference

Implementation of a Galileo CNAV Data message as described in Galileo High Accuracy Service Signal-In-Space Interface Control Document (HAS SIS ICD) Issue 1.0, May 2022.

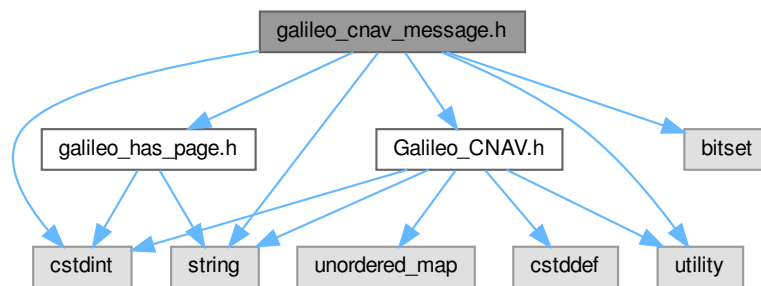
```

#include "Galileo_CNAV.h"
#include "galileo_has_page.h"
#include <bitset>
#include <cstdint>
#include <string>
#include <utility>

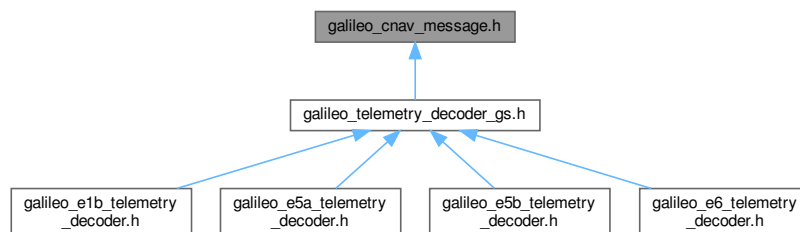
```



Include dependency graph for galileo\_cnav\_message.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Galileo\\_Cnav\\_Message](#)

*This class handles the Galileo CNAV Data message, as described in the Galileo High Accuracy Service Signal-In-Space Interface Control Document (HAS SIS ICD) Issue 1.0, May 2022.*

## 11.697.1 Detailed Description

Implementation of a Galileo CNAV Data message as described in Galileo High Accuracy Service Signal-In-Space Interface Control Document (HAS SIS ICD) Issue 1.0, May 2022.

### Author

Carles Fernandez-Prades, 2020-2022 cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [galileo\\_cnav\\_message.h](#).

---

## 11.698 galileo\_cnav\_message.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file galileo_cnav_message.h
00003  * \brief Implementation of a Galileo CNAV Data message as described in
00004  * Galileo High Accuracy Service Signal-In-Space Interface Control Document
00005  * (HAS SIS ICD) Issue 1.0, May 2022
00006  * \author Carles Fernandez-Prades, 2020-2022 cfernandez(at)cttc.es

```

```

00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019 #ifndef GNSS_SDR_GALILEO_CNAV_MESSAGE_H
00020 #define GNSS_SDR_GALILEO_CNAV_MESSAGE_H
00021
00022 #include "Galileo_CNAV.h"
00023 #include "galileo_has_page.h"
00024 #include <bitset>
00025 #include <cstdint>
00026 #include <string>
00027 #include <utility>
00028
00029 /** \addtogroup Core
00030  * \{ */
00031 /** \addtogroup System_Parameters
00032  * \{ */
00033
00034
00035 /*!
00036  * \brief This class handles the Galileo CNAV Data message, as described in the
00037  * Galileo High Accuracy Service Signal-In-Space Interface Control Document
00038  * (HAS SIS ICD) Issue 1.0, May 2022
00039  */
00040 class Galileo_Cnav_Message
00041 {
00042 public:
00043     Galileo_Cnav_Message() = default;
00044
00045     void read_HAS_page(const std::string& page_string);
00046
00047     inline bool is_HAS_in_test_mode() const
00048     {
00049         return d_test_mode;
00050     }
00051
00052     inline bool is_HAS_page_dummy() const
00053     {
00054         return d_page_dummy;
00055     }
00056
00057     inline bool have_new_HAS_page() const
00058     {
00059         return d_new_HAS_page;
00060     }
00061
00062     inline Galileo_HAS_page get_HAS_encoded_page() const
00063     {
00064         return has_page;
00065     }
00066
00067     inline bool get_flag_CRC_test() const
00068     {
00069         return d_flag_CRC_test;
00070     }
00071
00072     inline void set_time_stamp(uint64_t time_stamp)
00073     {
00074         has_page.time_stamp = time_stamp;
00075     }
00076
00077     inline void set_tow(uint32_t tow)
00078     {
00079         has_page.tow = tow;
00080     }
00081 private:
00082     uint8_t read_has_page_header_parameter(const std::bitset<GALILEO_CNAV_PAGE_HEADER_BITS>& bits,
00083     const std::pair<int32_t, int32_t>& parameter) const;
00084     bool CRC_test(const std::bitset<GALILEO_CNAV_BITS_FOR_CRC>& bits, uint32_t checksum) const;
00085     void read_HAS_page_header(const std::string& page_string);
00086
00087     Galileo_HAS_page has_page{};
00088
00089     uint8_t d_has_page_status{};
00090     uint8_t d_has_reserved{};
00091     uint8_t d_received_message_page_id{};
00092     uint8_t d_received_message_type{};

```

```

00093     uint8_t d_received_message_id{};
00094     uint8_t d_received_message_size{};
00095
00096     bool d_test_mode{};
00097     bool d_flag_CRC_test{};
00098     bool d_page_dummy{};
00099     bool d_new_HAS_page{};
00100 };
00101
00102
00103 /** \} */
00104 /** \} */
00105 #endif // GNSS_SDR_GALILEO_CNAV_MESSAGE_H

```

## 11.699 Galileo\_E1.h File Reference

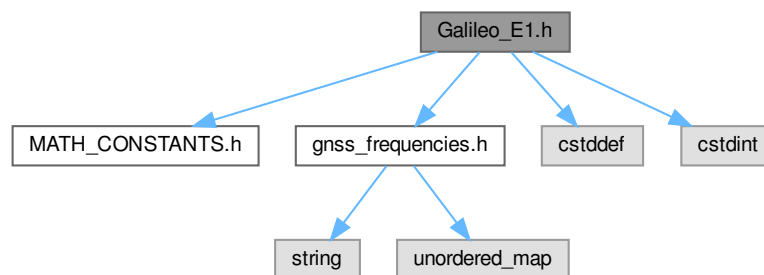
Defines system parameters for Galileo E1 signal and NAV data.

```

#include "MATH_CONSTANTS.h"
#include "gnss_frequencies.h"
#include <cstdint>
#include <cstdint>

```

Include dependency graph for Galileo\_E1.h:



### Variables

- constexpr double [GALILEO\\_E1\\_FREQ\\_HZ](#) = [FREQ1](#)  
*Galileo E1 carrier frequency [Hz].*
- constexpr double [GALILEO\\_E1\\_CODE\\_CHIP\\_RATE\\_CPS](#) = 1.023e6  
*Galileo E1 code rate [chips/s].*
- constexpr double [GALILEO\\_E1\\_CODE\\_PERIOD\\_S](#) = 0.004  
*Galileo E1 code period [s].*
- constexpr double [GALILEO\\_E1\\_SUB\\_CARRIER\\_A\\_RATE\\_HZ](#) = 1.023e6  
*Galileo E1 sub-carrier 'a' rate [Hz].*
- constexpr double [GALILEO\\_E1\\_SUB\\_CARRIER\\_B\\_RATE\\_HZ](#) = 6.138e6  
*Galileo E1 sub-carrier 'b' rate [Hz].*
- constexpr double [GALILEO\\_E1\\_B\\_CODE\\_LENGTH\\_CHIPS](#) = 4092.0  
*Galileo E1-B code length [chips].*
- constexpr double [GALILEO\\_E1\\_B\\_SYMBOL\\_RATE\\_BPS](#) = 250.0  
*Galileo E1-B symbol rate [bits/second].*
- constexpr uint32\_t [GALILEO\\_E1\\_CODE\\_PERIOD\\_MS](#) = 4  
*Galileo E1 code period [ms].*
- constexpr int32\_t [GALILEO\\_E1\\_B\\_SAMPLES\\_PER\\_SYMBOL](#) = 1  
*(Galileo\_E1\_CODE\_CHIP\_RATE\_HZ / Galileo\_E1\_B\_CODE\_LENGTH\_CHIPS) / Galileo\_E1\_B\_SYMBOL\_RATE\_BPS*

- `constexpr int32_t GALILEO_E1_C_SECONDARY_CODE_LENGTH = 25`  
*Galileo E1-C secondary code length [chips].*
- `constexpr int32_t GALILEO_E1_NUMBER_OF_CODES = 50`
- `constexpr uint32_t GALILEO_E1_OPT_ACQ_FS_SPS = 2000000`  
*Sampling frequency that maximizes the acquisition SNR while using a non-multiple of chip rate.*
- `constexpr int32_t GALILEO_E1_HISTORY_DEEP = 100`  
*Observable history length for interpotation.*
- `constexpr char GALILEO_E1_C_SECONDARY_CODE [26] = "0011100000001010110110010"`
- `constexpr size_t GALILEO_E1_B_PRIMARY_CODE_STR_LENGTH = 1023`
- `constexpr char GALILEO_E1_B_PRIMARY_CODE [GALILEO_E1_NUMBER_OF_CODES][1024]`
- `constexpr size_t GALILEO_E1_C_PRIMARY_CODE_STR_LENGTH = 1023`
- `constexpr char GALILEO_E1_C_PRIMARY_CODE [GALILEO_E1_NUMBER_OF_CODES][1024]`

### 11.699.1 Detailed Description

Defines system parameters for Galileo E1 signal and NAV data.

#### Author

Luis Esteve, 2012. luis(at)epsilon-formacion.com

Mara Branzanti 2013. mara.branzanti(at)gmail.com

Javier Arribas 2013. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [Galileo\\_E1.h](#).

---

## 11.700 Galileo\_E1.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file Galileo_E1.h
00003  * \brief Defines system parameters for Galileo E1 signal and NAV data
00004  * \author Luis Esteve, 2012. luis(at)epsilon-formacion.com
00005  * \author Mara Branzanti 2013. mara.branzanti(at)gmail.com
00006  * \author Javier Arribas 2013. jarribas(at)cttc.es
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019 #ifndef GNSS_SDR_GALILEO_E1_H
00020 #define GNSS_SDR_GALILEO_E1_H
00021
00022 #include "MATH_CONSTANTS.h"
00023 #include "gnss_frequencies.h"
00024 #include <cstdint> // for size_t
00025 #include <cstdint>
00026
00027 /** \addtogroup Core
00028  * \{ */
00029 /** \addtogroup System_Parameters
00030  * \{ */
00031
00032
00033 // carrier and code frequencies
00034 constexpr double GALILEO_E1_FREQ_HZ = FREQ1; //!< Galileo E1 carrier frequency [Hz]
00035 constexpr double GALILEO_E1_CODE_CHIP_RATE_CPS = 1.023e6; //!< Galileo E1 code rate [chips/s]
00036 constexpr double GALILEO_E1_CODE_PERIOD_S = 0.004; //!< Galileo E1 code period [s]
00037 constexpr double GALILEO_E1_SUB_CARRIER_A_RATE_HZ = 1.023e6; //!< Galileo E1 sub-carrier 'a' rate
00038 // [Hz]
00039 constexpr double GALILEO_E1_SUB_CARRIER_B_RATE_HZ = 6.138e6; //!< Galileo E1 sub-carrier 'b' rate
00040 // [Hz]
00041 constexpr double GALILEO_E1_B_CODE_LENGTH_CHIPS = 4092.0; //!< Galileo E1-B code length [chips]
```

```
00040 constexpr double GALILEO_E1_B_SYMBOL_RATE_BPS = 250.0;          //!< Galileo E1-B symbol rate
[bits/second]
00041 constexpr uint32_t GALILEO_E1_CODE_PERIOD_MS = 4;              //!< Galileo E1 code period [ms]
00042 constexpr int32_t GALILEO_E1_B_SAMPLES_PER_SYMBOL = 1;          //!< (Galileo_E1_CODE_CHIP_RATE_HZ /
    Galileo_E1_B_CODE_LENGTH_CHIPS) / Galileo_E1_B_SYMBOL_RATE_BPS
00043 constexpr int32_t GALILEO_E1_C_SECONDARY_CODE_LENGTH = 25;      //!< Galileo E1-C secondary code length
[chips]
00044 constexpr int32_t GALILEO_E1_NUMBER_OF_CODES = 50;
00045
00046
00047 // optimum parameters
00048 constexpr uint32_t GALILEO_E1_OPT_ACQ_FS_SPS = 2000000;          //!< Sampling frequency that maximizes the
    acquisition SNR while using a non-multiple of chip rate
00049
00050 constexpr int32_t GALILEO_E1_HISTORY_DEEP = 100;                 //!< Observable history length for interpotation
00051
00052 constexpr char GALILEO_E1_C_SECONDARY_CODE[26] = "0011100000001010110110010";
00053
00054 // Galileo E1 primary codes
00055 constexpr size_t GALILEO_E1_B_PRIMARY_CODE_STR_LENGTH = 1023;
00056 constexpr char GALILEO_E1_B_PRIMARY_CODE[GALILEO_E1_NUMBER_OF_CODES][1024] = {
00057     "F5D710130573541B9DBD4FD9E9B20A0D59D144C54BC7935539D2E75810FB51E494093A0A19"
00058     "DD79C70C5A98E5657AA578097777E86BCC4651CC72F2F974DC766E07AEA3D0B557EF42FF57"
00059     "E6A58E805358CE9257669133B18F80FDBDFB38C5524C7FB1DE079842482990DF58F72321D9"
00060     "201F8979EAB159B2679C9E95AA6D53456C0DF75C2B4316D1E2309216882854253A1FA60CA2"
00061     "C94ECE013E2A8C943341E7D9E5A8464B3AD407E0AE465C3E3DD1BE60A8C3D50F831536401E"
00062     "776BE02A6042FC4A27AF653F0CFC4D4013F115310788D68CAEAD3ECC5330587EB3C22A1"
00063     "459FC8E6FCC9C9DE849A5205E70C6D66D125814D698DD0EEBFEEAE52CC65C5C84EEDF207379"
00064     "000E169D318426516AC5D1C31F2E18A65E07AE6E33FDD724B13098B3A444688389EFBBB5EE"
00065     "AB588742BB083B679D42FB26FF77919EAB21DE0389D9997498F967AE05AF0F4C7E177416E1"
00066     "8C4D5E6987ED3590690AD127D872F14A8F4903A12329732A9768F82F295BEE391879293E3A"
00067     "97D51435A7F03ED7FBE275F102A83202DC3DE94AF4C712E9D006D182693E9632933E6EB773"
00068     "880CF147B922E74539E4582F79E39723B4C80E42EDCE4C08A8D02221BAE6D17734817D5B53"
00069     "1C0D3C1AE723911F3FFF6AAC02E97FEA69E376AF4761E6451CA61FDB2F9187642EFC63A09"
00070     "AAB680770C1593EEDD4FF4293BFFD6DD2C3367E85B14A654C834B6699421A", // PRN 01
00071     "96B856A629F581D1344FEF597835FE60434625D077ECF0D95FBE1155EA0431979E5AFF544A"
00072     "F591A332FDAEF98AB1EDD847A73F3AF15AAEE7E9A05C9D82C59EC325EF4CF264B8ADF2A8E8"
00073     "BA459354CB4B415CC50BF239ADBC31B3A9C87B0843CF3B9E6D646BA43F866276B053826F3A"
00074     "234CC5E2BFB9F8F195B382E75EAA63F58A06B3F82A3B5C77C1800FD9498F803E524435B32"
00075     "1210BB84690BED0BBBE16D363B3A90656A73720E27008852FB7DACC8284411B17728D9527"
00076     "C560859084A395A6F11A96AD9DB6B43E00642B000ED12BFD967868EAB1108552CD4FC89FBC"
00077     "408ACE7678C381EC91DD000319124EB5D5EF52C4CAC9AADEE2FA045C16CE492D7F43743CA7"
00078     "7924C78696FCBF2F9F7F36D8E623752200C6FCBB71ABBB6877F3C5D6E6740AB0389458A6B"
00079     "66440858B2D38244E85364FE2714211DEA9E6196252815BB704A20BFE556AC474F899894"
00080     "E40CABBE21A6400B87BDFCF937D12B2821D59298AF4AD378F0F42BD8C41693B8D993CF37C"
00081     "B8478F3BB5D33AD2A9FA24AD7B8FA895FDBC04964192F7BA3FF74E0E3A435B5DFE042E3115"
00082     "CACF29624C0645E9C917534A2EBC1F5665E4E1B1BC56208DBC8A27CCB6474D5D0E20CA407"
00083     "2C96E5ACEA41BDA3770DF3B681F2B318F6F8E1CB17C2857350FB6009AED665E13B278D0792"
00084     "17F73FAC7A8A48048DB0FB8A8A5007CDDC9A7B2DA8257C99F1CB605A18204", // PRN 02
00085     "E57DE19A3E4A8C122FCB1DD6584B3D2DAE364D800F9C5A9E957B38F624CBD3ACC58FA3ED07"
00086     "0B5E44857FCB813FBC0BB83B5D157C6C562422E5963CC4DD753C45B0264F8E136A0F1774D7"
00087     "7A543E44D51EF8C6B9408B6E3B5CEE1347A94F13ECDC94DC764976E5A50B4CB0AE7557553B"
00088     "47EDFE03EC2CD32EA8D125A341E1EDFC77E75330D6E7B23DC838EBCE7E5567F5B8C80C3D15"
00089     "E77404BE10F0BBE0C69626A814AF91334199864FC77E0FF548DC2A6FA6A71C3C0561F2B085"
00090     "CC05E8512E27B9DBA60B93D114B87935776C8E9A67905C429D48BF3AB1B0A56FAFBFD59C8"
00091     "D8C8A9E5918BFF273FC5E8664FF2B90314BDBFDAD5AB8C22A0E45C104ECE75EA43FE9BDCE3"
00092     "06A5A28AE4164628163D249D8056005F1A9009511808CC8620F81768153436F741667A8E271D"
00093     "D986C7A1E5046FCC74C7CEBBF9A1296D6CF0B2FF85BE412D87214BB368DFF462AD649D7324"
00094     "A117252311C664D33E4DAFBD830FBCEB6EFBDD7391D4BADA7A775FD1949D981F619655DB3C"
00095     "22BAC34E5AE41222905C0C7E80D6EA28471EC0468756531C09A471EDBE200472E78F1701FE"
00096     "E96E5769A9893C0F11E7906B064442E06E21ED8B0D70AF288690C532AD03B373E1E0085F6"
00097     "2F7AAA658B569C5184E3DDC40ECAA88B887118601691892F9F55E2DE79E49DFF11D434C2BA"
00098     "3AA6447522A7C99DC215CAD2ED0114ED62CBDAE9D315E48AE14D2014B7F8E", // PRN 03
00099     "C0FC4C72A12023BA7093C86775DF3D2F42C7CEDE616876340BE4301361B9DC9DFF4F1DEC6A"
00100     "62E165927BDE4F809E969AAD085437496BB95904719820F4CA8ABBA0B84C34B06DD7E268BA"
00101     "10E386FA7DB9FCFCDAF2B6AFBA46A8A229153B4E11582FBA7F28F0A0F9DE41830AB333506"
00102     "2C57D81DC361EDFE491939100FC827F36273760043D1C35B74E36C6C4DBE1D307847D55AC0"
00103     "7D8B212C2D6A632A8AB15BD0FAFFA43070644C7E50623195A3796AA8E8D6E4E964FA0E448"
00104     "8A500B9063FBFB1204A0E33C6CF2879AC2BA7C86CAB57E3E8A497836194E65C39B950F1"
00105     "AFC3B5E850A5EC39F4190D55351D16529CD52B36DF4A2DC68EE202BB758CF19C54B0E1461"
00106     "D547B5D06C2F9B0C9C2B15458C3140860E4C6F3FE4F417FDFCEDE00F71212EE137E6669E56"
00107     "9A7845470CA564F85CB4772808D65D2B48D409B709BD7AC5F7E28AA804CE9DAC3ABB5A5B76"
00108     "8C6A184B5A974E933F2C1772FF64AB26BA2D5A165744E314EFB2238AC485A8B82723DAE88"
00109     "65478EAA261F35DD4D98A9C07ACB0822AFF1AD3E739CB214CE737196FEF2DD0B0D45BAC42"
00110     "3935670BCF71C2EC04CCB98943786173C309E75A02BB78A788A5E6F8A8F407E57B8403841A"
00111     "9E1FCB3A7AB80D1F6529770E52C173E2C47EDED4400D5E665E325ED845C9E8D0E66FDA16B1"
00112     "7D61EDBB336F22688C3F0FB040A55F33B65FA9F3D45F5B22C445CBF9DEB22", // PRN 04
00113     "EA9596357B343DFC31D5875CC0E94117A33651472E476D3892D8112EB6CB6E0151D409C5A5"
00114     "14DCDA38A773C58F18B590EF9017B6EDF0192AB7EB29DD6E1E7E7390C13E9B10209D5775F3"
00115     "B066F7B2DBB7307FB44F726DD2F368A5FD8E75BA7248762E1EC7E4589DF1A353A16D6B3CAC"
00116     "1C9ACDB89890ED2C4F44AFEC763DB51D102230C37E1ED0943CD6F4176B2F5C19119588911"
00117     "ACF81A7A29320AD579C1BFAED1A70DEE1B87037138ADE411E0BB92F5B3148DFA11F2F84CA6"
00118     "C019124B922837503AA9823A97E443A66378D5CB3130A7EC9B05670E85D095D5E6F603092C"
00119     "632E51FD9013FE7FB9F08448FD09F1219A4744CDAF82BF9C60039C8185C7E9559FCE301C6D"
00120     "3F46A2E514AD44D3889C8CB4ED7439BF47019194F2644363770F8BBDOAE92B6F5F43CBBB5"
00121     "03A885239DA636903D4C264B3FF09AB77E3FDBA7EFC63E0792B6D5183759E57D8A694CDB13"
00122     "3B4A9E301CEE9B78050AD9A9E410091AD29E389829E2F24BE1E3B24F4540C4A6533EBA72E"
```

00123 "8AD540BAAE43A0CB82F971F3A51DD77FE9E1956E2EE7553E050A1D10B99552DD5B68F2E28"  
00124 "59712835BD2AD6B08881753B4833FB04740E3364D2CD4921B939393E7EA91B854FA1E5A8EE"  
00125 "79FF0A83F111F78435481D462E0E1CBC0C921D190A435A1BA755E4B7021244FC5E3F0630F2"  
00126 "A1F439C02AE619393E5624834B05ED7DEDE5F0AFC7A40899424E75D4EE792", // PRN 05  
00127 "90E92279CD4F60D98F6E8FCB3E9263DB60FAB146A835AAC2E96B3BE3FF07119032DEE0521C"  
00128 "731117E90C2943B389DD6B65C5E21C34F86F5A7ADE04072DFD1479EA36528D340736B0FED4"  
00129 "F6207BE9F6CFC971D5EA11781AC2DA25DBEEB6B903EF8BB0AC0CD2E29F94B8CB67874A7B74"  
00130 "41045758E09EA061181A50E0AB7BCCF801554E0644780BC137436E3FB7784C182856A790D6"  
00131 "943BB53DB40D13D6A2F7B83A5C521073883B90FB8DB1C0F954D132943C09156A09984B8220"  
00132 "79FB8FD09BC07C1D6336C7CEAE8CC3162760B9838CA6A38FD0044FDF099E416D57BF9F33A5"  
00133 "51043F34EBF9BAA90901E62D2D981065F977852072F692535DDE24EE8946387B4E5B0FEFEB"  
00134 "D75552C1FC325A608A78079A9AC864F2F30010A3304CB16A26AF98D9BFD3B8D128541190B6"  
00135 "BBEE275A6F53B9BC5108306985ECBB983B56E34F18B48A12AEAB88271F4F780CFDFA83E05E"  
00136 "35C12464F4350597CCAE9B4498F5A5454DCC3218D3336763674934ADCCBCB5EA52891EB240C"  
00137 "362248226DE64899BE30735F6495E94AA61ABEF62B803C57FDD045B724ED1966B6E7DFDFCA"  
00138 "5B36F7B0FACEDAC62DE8E10B12DFC84B1A9CEB407BDE63CDB5208ABBE5E066AAF262187E94"  
00139 "502B1701B2CC8681CB616773DA2B7AF49443CFF528F45DD7F25959836771908C2519171CAE"  
00140 "D2BCDCFCFA46301E7D99A5AF7199155772E92BAD85F35EDB656F0999EE828", // PRN 06  
00141 "A91F570102961D62CA6CB55144AFCCEAF3910F3336DCB029CDCBA164ADA72732771B6ECD1C"  
00142 "58E49F468A2BFD23E1B996DABABBAF5AB3A4C74926187B5833006F8BEF7F9CD0F05A2A0B9B"  
00143 "D9073C4C3976E860CE7BF81634CF0B31C3DD806A6A0C15BC552B83A86789CC675A6D137B"  
00144 "E27BC86DF68FEC5D268119EB9E965260FE1F5C56AEF60A8622CDA8C42F24CBA7F5B07A7416"  
00145 "91727723314AFD3ECD10F74BEE7B22DC760EFA7F935FC9963411353782547FAEED32E69A4"  
00146 "FB5756C1A73CCDFFEDE50F4B2D9B5D2ED5C59C9A52D80CD27B989B8DA14C569E763C08FD4"  
00147 "2358CD064B2DE0526607C9536D75E1617EC80615EF5EE2314FAC29907B61B61F8696CB80B1"  
00148 "4B3A0148EEBC825C91150A08A23FC7B38B5982AA02A18BF6E91B3A1F2EEF360F682A34AB36"  
00149 "CAFCAD556841073F7ED7219910F7BC2F07CE45E98F77F50475DF9EDFE2DC9E3D7280193D61AB50"  
00150 "76A14887E9D9193C3B83C5773BDECA067CA1BC3D4561C3A8B4E30072A6269B529760CA1B5F"  
00151 "E9D3DB2B5D1202CEB18E9E2E80FAFF47108168D3C7EB3C940B1A35A1D1B968A5A9DC0686D"  
00152 "D8336E49FC8240F28071600FF995B9E33169DCFCFCB58E75C94D82F843C60A7118F0D7B4006"  
00153 "4A8A4176C5158E86AF0BE4C1D5D73D1C051132A85CC0628486AFD660502A515D6353B674B1"  
00154 "D4E61750C13E8A3AD48FE1F89F201C288A8F443867C2BAC23C706EE7A2D2C", // PRN 07  
00155 "C6E00978E351164532EEA256ECBE0D4F8FCE02A276BD19666DE93936F7A242FC4C7E879791"  
00156 "314B043ABF1D5F9B0036ED22AA92028C800C4D62BD6640431170EA77311865074D670AF284"  
00157 "7AA47CB94584A793FA82F51574BD7C62BF14386F14A3D7DBD129FDE64EAD67EB35D5E13FF2"  
00158 "14D7D163B770DA477A62D02D88C0FC3FA5EC306EB7F85539105FA2CE5F53D182E58FBBC1C"  
00159 "57FCBCD2D2F7FC8A067D6FA0BC834DAB8F370B0971BF6D068CD4D3A32C11C6598DEBBAAE04"  
00160 "6528C5EF76288CC84D003847069FA18743A809A004431E83924B8FDF0AC78699B905ACCCFF"  
00161 "82E83FADFEC6848DF64042FC9438B261B73F0541498ACAD67D702AB631BECEFF8680D33CE8F"  
00162 "4F0CE29B95132591A350DD68B36734B97D4B3E84A76497F702312F2A8370DCF26A7C3CE8EB9"  
00163 "1DD8699C48F551750712683E039708371A6CAC3457C0FA70BB3A036C6E0BEF24E6B20BA55"  
00164 "65B351C2EFD56BD9455FF7728BE07A097208E73DE4CD0CB4E215B4642365123CDEA419B284"  
00165 "59D50E864B762554E7C1D7CAF73DA7D40DEF5D824A2FE1A6CA473B07370932A8A50441DEE"  
00166 "3C9A60DB68E27A9D3E9C8229B44E5B434C6D18A8CADB6D17BC4614DEBEAD670C73132CE2F9"  
00167 "99C8716D1098C69277E8ECAC546FE8002E5182E25F31A354DF112E97F8733DD20893B430CD"  
00168 "7130E6ED4A0FE4D6C2E4FA79001E42EBC9F36E5DFD3E0BE35A64B89745E", // PRN 08  
00169 "821BBB3FB91E50253A9E71AC379ED57AEF394C2CC59587B2D0337CE74002EEAD17AB5D504B"  
00170 "CA68BDAE9061C3DBAE2985EBE292B9BEC9D3542015225F44ED3C2C3FFB036A515BF33DA169"  
00171 "0F3438FD225A5034106C5F4BCC43301EEC2245D73F63038E2A7D9B8CF95A9FD813FFA071FF"  
00172 "DE423E0CE737969578BEB909764A8D6DAA9E15A4FA0867831652C0F6E9AAA39A63F0AEFF62"  
00173 "A433476CC7380460ECFB8B7F3B2FE8C4C42A3EF1CDB808FC9747FB4F044B3B47A4EDFC9450"  
00174 "3ABB72C55399B2F79EE5FEDA270D6358B27F8466969DE4A5F2E6A5F2C4CF0813C09F468DC9"  
00175 "7FC0E5DD057A8A0355767B698F8A79BF0350C4200413A15E6591DE70A1B502E19FF515C3DF"  
00176 "36935974A4764895B9E3CA2626BD39B7ADB780AAF7E2E914E804CA923089A51F3876649C73"  
00177 "CA3C2623A8C95D11EF4B3F941E9772EBA1F47212C666F03F01509FF699F74EDE27182B6E98"  
00178 "AF49D1BAACB413A28A8C34D6E8AA3553DA3962B27B041495F269328B6BFB4A385CBB118953"  
00179 "F3F009920EC48590003290DD60AC89177BB8C4BF753CE723AEC392B8D9E5E9E4113DD062"  
00180 "F294A77B6EA9A0477E697C04C787CE78A92C704409D37D37B6B392128698D0D8D4CA101EB3"  
00181 "8B92F467F0D86EFD8759A14162CAB55F8C457E82392790A5BDDC8DD2663944F880C95EC02F"  
00182 "E5363B064623994EE5D4396C0E44DE2A3D225830BA6160270BCD110A942B0", // PRN 09  
00183 "92A0DEABA9875D4AFAF99A24C1D5F10EBBE6DEF9CAE5B0C85B2A0417C1CC5D1A5F71CD8F8A"  
00184 "4B013C3F012C0A19EE4A23106CAB8662C5A2A93A971D0B6E487FC05BAF5C355A9520C91485"  
00185 "84CFED3EDD0F38696E161E64378C831C586D9178A0CE289A67F33AE68C02A3CD138FA09DF1"  
00186 "CAD01EFADF8CBF6F5407B79B18D09C82804736752D08A1FE09EB35F544E9F797EA36DB493B"  
00187 "A947AA82513EB1615A356B5AA4308B0B4183E070EB494D628159D24BC3B110AB0CCB2E9E"  
00188 "73B5B7EB561787621E72D99F1FB78565917B28464A5F29DD8D6F98B6ED70304A44B0ACD97"  
00189 "F15049E009E8533FDB0B6DB2F2582E6BBF81D7B0EADC8F402508F6B8531AD13FD1C55978A8"  
00190 "A70DF4E053DD475132D348AE27581370EC14A3E0F96E0D70DA4946DEEC0760011404FDC5B4"  
00191 "36CA7419D05895F5E0EAEBC88C74947733BE9919F18CE702887A6C4DF7C19279B82F6B6460"  
00192 "90822DAD9CD9C7653F6B931A337A28F7A4A01DE0CC0744F22961045F8EF8D4B30B07E5EDF5F"  
00193 "A944EDCFB9841A9088AE82444FCB6E90B0E9C567A80E8C42EC713D78132F37AD1D2592C31C"  
00194 "93D2EAEFF38AD94E5C0D94F949F47B88B03BC1EA4E5EC9C7D9DF19ED208B8E44FFDEB0B625"  
00195 "F633C7DB1C826AA9E1C1309E5B14A0DDDB79714DFDCB52221CEAD7E8A140DF7806F1271564"  
00196 "78AFBEE922B8ECF322D66B48BEC434299BBB36B3BD9030467B7F2EBBDF358", // PRN 10  
00197 "AFA7FBAC93326D0C36A388831B99DF4D527BCE7C9070F7B46B5FFCDEB07384801AE5F86A89"  
00198 "934DE23DFE2C1AD117797D4FA1BBA6175823B41166DBE9D126F17B3761E2C352AB396A5A9C"  
00199 "CEA42A5E9EA1BE3497C0A5BA9121DB97F64159AAC78E62D7DEFF3BF4CF73F8CFBE045C9D39"  
00200 "E41D5D208DCC4B47CA27E900C3CD8FD1408DC5E0F5114F2FE65817D37CD1452C4967ACAA21"  
00201 "19FB8D60E5E2FD8A8E2D0DAADD88B94D40435C095568AE6394D3B97C835BA868A83083316C4"  
00202 "9C75D36FDD85165BE74A4F2B2D21295EBCE085D9C4A4758FDD9CF71B97FDF34B7B63A5E96"  
00203 "91DBDAB834D87D5B52CA9A53032FFE821398616EA92625C2DB633E37911987083A3B49A86F"  
00204 "C562FB1264A75643A5FB6E97162E16ACCE353227FE61A859E094C2359BC4645946AD12AE5C"  
00205 "39C70F59EA7B597A9B3372C23AA578146781A61163C92816627DD9C4BF1788087821F9F5D4"  
00206 "1B75A0F251B06BBD3E29ABD41E72A1D48323D24E2AD6F11C2D49678CC04FCF6B0EFD33BE6D"  
00207 "DCD444F5CA02F8158112631F782CA7B0C5F3607ED807495BF8E82CE5A51A922FE28C8168D9"  
00208 "844859E7A3EE3038C5D1D4BB4813406C340894DF46406836739E31D01082BC84489592DA0E"  
00209 "985630CEC40702A36DDC301B3AE1E8101786FEDBF752F9E175287C239C18FC25795BCB479D"



```
00210 "EF59C58C373313C02A1BC5F16355E2B50EFB58855670868728B902653ED80", // PRN 11
00211 "943CAEB680AA3B630755DF32F406F403D7AF5E48A710274D3887A7AAC8EA6744B889F2E0CD"
00212 "2033DECOB434A9591254A0AA68C5C9BF1D35765E86B437497D84E5DCBCC0C580CE9BC50"
00213 "EC6382AD74DB02C2C233B7BB07517D48056226C505ABF2DD244F6BBAA23313D570558B065E"
00214 "4232776807BEFDB53DC465DA038E3B216D990EE951B3E13D3C1CD559998F77BCDCD2B9522B"
00215 "F61DC5E12C912EAEF574AFD69C251F9B2532501AB9F4B3B2223D0F8920BD562B0D358A14AB"
00216 "0D196DF6337D1C96CDB47AFEC6F81DED4B5773864DA32FCCD06B9AC53C122B2C6327E6E5EF"
00217 "E227DE4893FF15BBB2257FAEA836E99676EE32BF6FC14D4F56EA191B8A3870374A0867C49E"
00218 "B0015D1C6D07B87A36BFDD1DCEF20EA7B80D997CBE2D83EB5630F2EE6F73B0D50700C89E4F"
00219 "32438F5541360683DF11DA6E7A3C1E7DB2A87800D9245BF04278C990A8DC9CD86DEF39CB6"
00220 "D4BC00FF13BBE132F9D86681A8913BE787CFC69C35304824788716D52DC74CEA399E06DE62"
00221 "41780447C74DA8E947134D8B2FAA9648D6D5F34C9D60AE5973B5BB0187796D589C8FDD767"
00222 "5671F28C04AC1038D09251980683CB712F694D7C5B0D5B1DE86CD10EAC4EA04A55BA8803D7"
00223 "8249BEF516D38067890105A2321E72879FA267A8B4F0455A81F17CFD3E5DDC55E5D4FE00F"
00224 "83E18626C676DAF00E6AAFC23D209DEE0B0FC6C2AE4DE161D13017ADB5D8", // PRN 12
00225 "E5E70E7837D094416558C044D758383EDF5755C80921218ABE76E51FB93249E211A38FE6D0"
00226 "7A7DFD2263E6E3D8DA0F921A06A606B804DE7AC3FD097E5F96EFCF0544D623FD6F43FB88C"
00227 "EA7C341E901CD47A7E24AB141E998FE41CA87CD6CE8C1870D9ABB6503BF7E8B659084BAF22"
00228 "37DFC94F35C9884C7F44B87120BFCB298696E613C1656AC4899781A94869EC603B4D386653"
00229 "37CA8593AAC83AD8ECE10302E4B4694237E96CCD3AD9CD5F8EC039A1D1A4210716371404C"
00230 "5C3FF375CB3A3C559B1C1A239F5E442C8EB033501BB290434BE73489F716965393989422CF"
00231 "4D57E5B4F3C76AF3C5E8999E61805134B9D7C40BFB59D0D0FD30F98567E66D6148D6AA64F7"
00232 "4A22C50AE49D681ECC6BB5A002ABF38FF2E2436766B86BDE7D95DD6E02AB0FF06E7BC22CE"
00233 "C98D55AA2BC4D7B91C36B2FF9F525A74423498D548318509320FCBCA582A6C2996AF65384"
00234 "22FF0DF060C0BC7356B0850A139AC3914338127B786F4BC58CEB6064DA881376A147DFF53C"
00235 "6700BD13316A5874A75D7B9713DF54FBB393BAFAAD7F7B0710C049A0B6A8B76A9956BF6185"
00236 "BA39D9C347D1CD47A7E24AB141E998FE41CA87CD6CE8C1870D9ABB6503BF7E8B659084BAF22"
00237 "56ABD7F851B65B220A72BA48FAD0436332E4EE7EDC554B7D75481EE05C3D3453D760E9099D"
00238 "D27B324DD84C0C0C4DEC4C674D25284B16410F959FBD09D9DF09CE875601E", // PRN 13
00239 "BFDB8C2ACB4FCBD5A90C5967EB2FED597A02607F426002128AF4B38942C85AF4472B3CBF3B"
00240 "183F240E049B251713740A31117F108936631FDF0F11C5F79325BD6677A2C2B242965AEFC14"
00241 "7D93358730AA782491209CBE600976F56030753CC979C240A196647CD9EAB1DD0380E59BC7"
00242 "905EF740C3411AD9DD72027D0D3DD6DEB0F5F3C18F6D6F7BC59B758E7E262937B4599B3856"
00243 "7C147ED2689BA2CF23736CAF55B6925827E2B70E47D3813C94C85298BD6B49C97B5D0221BE"
00244 "9E3164B6FA3D95AECF53AF170966090F19A69E75F188BD2556B4E8FA7DC4AC6C34F54297C0"
00245 "6C2A96DD1C45B42E6175B5E8784568F7FEF0B6C124C5019CB577B374941E8515CCFC21F46D"
00246 "188BDD2C2284C688879A5BEC50CCB97FAEE1F75580577498D509D3DE161BE216C873B29E17"
00247 "8CE17DCAACC5E9E224D05ECC842FBEAB82A75AAA20769FD81131CFB69D5E35409273CA106F"
00248 "FB27F63FF997CB500F161F6DD3A8BFA5719F004EC17860152D3290951678A131E4F3D3AB34"
00249 "CFCFAB2967ED9D8F1BB987950306BD28751D2AEAB05F071B08574EFC A01E5386E04F727BF4"
00250 "13A8279E9392EFB64D9AEE00877C76C81EBC861E2B484A2D35E592A131726CAE61BC010B95"
00251 "4721A82C968CC6F384D9BBB99B4E87846D10B94EE31F64846A5834DF73A67A267B894B1C06"
00252 "242D750F15F3E1E850A11CB2E2B16155008F91493AB3BC77CF9BE56F9DB20", // PRN 14
00253 "D64F3D1CB54CDB9143D9E701BD313779C09DA064D9A85674CCB53B0C5B4446C122098961D5"
00254 "EFFD6A8D537486D5EB26B5E18FFBF8E6EF16C2DD2C02EC7C07DB15CE33015A636E225F744"
00255 "C963BF0653A89A48F1AF04819E273A3AE1F5538AD574D553C5A0DEF47B552957037BCA9219"
00256 "70C76DDEF74BA083ED55363760A6780612C075964B083B4F674EA0012FD1DF09F0445CE75A"
00257 "698852086868AD8241E3B319FA8D2D86DE6E7631DF1AEB571F9676323E0627307F6D8F5"
00258 "69536A758DB5EDAADF80F4335E3AFCAD07F70AAD5CD08CCA1E71B84D4D97931F924AC0010"
00259 "C0811972ACAA44148B9FFF7917E653BB31E9CDFC72595066C662CDB9BBC96152D46BF4E8C15"
00260 "A8D34809CB49D79871BDF0B63FA294F2D667624F6E0210CD40C92F1C033C3D8BF089EF85C4"
00261 "F571CA727C71B23128A9B0FFD70CEA93C316FC4D69D79B089107F292E03425B2552AF5AA18"
00262 "FDB9AF86EA1972B66B1276B09119437E4DFB8F8E3972D91A93816EBD7D8D715CB47EFA7429"
00263 "38B0B49FA27A291B0DEA1DF0B8F87832103F45A9993689618E15FF65C6995F57C2C54B80"
00264 "02DEFF54B0EB3131EE7D61030C33B5502C49CF398FEC4B7615D16FCEA3E8EA12BFB311D426"
00265 "331A06606CA5A066707C4AF8D1048F1CA6065FBE506D06C6C00D5D250E22726551867A681"
00266 "FG05155FCBDE24DAD115BDA98AFE08B12A1F32E7C2ADA801FFB78BA05726", // PRN 15
00267 "9D6AD9889EA02FC9A58949290975DB0F512EB37C8156CC9F1242B9E45F22CC1D6ED1CBCB6C"
00268 "B245811CE729261641FDF7A8F389BAFD7311B8BD689E02409F6E8C5202F466349EA466E539"
00269 "8259C8CB126D9600D89697A07A6900FE8D95951903DAA3419839C2D9E35E9F4EABC04C9006"
00270 "EA585F544C7163A33D7E78DE28256B7B8978FE018CB529F7F79BBF66DC4F0DECE80AE3C2CD"
00271 "479D78C4480E4DE2F06C70E5FEBDFB4ECAEDC2E7BD891AD6C91A7C2446F1B13B340B716078"
00272 "2F6CC5B45F9787CF1B0985202DDF02EC552A6DC41325FD8D31A4316C13C56F7157134F66E1"
00273 "D103CC3AA7EB951C92094EB4409E6E7BC494434FAD80999D46D824A5A57390599052025F7D"
00274 "A4838F7D16A8DACDAFA06D175546FADD1E3F7975265230F6C01B9C1FB1B7AB1F2FDD43A577"
00275 "8E3C88FBEA70575CA26D94D249670E4D9FF28EC67D15829776D7BC6754D2ABB01554E5FF0"
00276 "C3FAD8A1CB546E8AD5E5314103D086D14ABD30EA95DDC591C1D96C1CC3F60FD18D216B671"
00277 "81BF6324AC09A97C0C45E50EE8380ED42FE0430639373E7760C708248EE7D74830E9594114"
00278 "879748883F247D056B2BA94A0FC54CECF6F5C6AB4DCB7FC8C224F40D886427504233DDBED"
00279 "CE160DEFFDFD69EE2B75746D9CF71676DC453FD01C315ACA96373ED387B040BDEBA7FF3CE0"
00280 "OD915F90AE6E1796971F8052160154E8986913AD7BA291188EC49A60BE27C", // PRN 16
00281 "B5184F7D580935ACFF18201CE8B5D54CD0A1CACF102FBC8AADF391C4CA5807BAEEF4E5E47F"
00282 "7459E74485E48E0C42D27CADE6970714FD97C08F9592FDD387C859FC12C1CCCF3EBF510D6"
00283 "6FBD8C448C2A5322CC5887F9A455D48ECA362C690F24833C3B032A047D12BDA2ADC6824A1F"
00284 "6EA9320BED27968E9CFBDECE60D041EF538F1740C0519003FAA89CD4224293167E05344998F"
00285 "D396EEF618E8F547990BC06A8B7D0DF6FAC13284601AB7191CEB813C46C45CE7B3FC09EDF"
00286 "08DAFE136BFBDD636CE7E4BCBB16C5DA68AC71A1298FD27363349A261C2F2CA8CB799E860"
00287 "4ADF70092BDBD6A04CB80568776A537AD1711891B251C74E42FCB095B23EEF70F167E8B485"
00288 "6BB7F92E3A43C79FF4437262DD70BAF9B16CBF5F10D1AD7559AB0F8CEE1B9FAD058E84FCC3"
00289 "42D9F0D9FBE4207D40E281416506242CA1B8DAB28DE88D2D00BA21AA7FDDC25940CB29F028"
00290 "11F8DC6850A6A87D72CA9F347A673649FB4A254B1204CC1261E7D512BFE7B0D0091AD5CB0F"
00291 "BBB765BF5AFDAB0701941DA54832FE8253BC0CF61924BCA2CA231A196C7C32A350AC9A5F"
00292 "A2884D8571FEEEDB7D29632E71898BB6B5E4E0104F73AA6A9C6B8CDA816872805D75ECA64"
00293 "F961641077B259C9D39E2F3CCD9FCFB1E6B6E2692EA34336A967E587F32E49B961B9131119"
00294 "CA204D11874B4EBEC6C04DDB5B82D5B741D3CEDC03A56A2017B3D2C4FBB4D", // PRN 17
00295 "FD26B78ABE21CDD6AF8C349F6DF8FF8B96BC8246A672A16E45B50AB7D992570EC45A534"
00296 "B77F204039FE200D4C5E7C78FE24941F578097B216177D8AD4E1844B2E52D843256D0BE850"
```

00297 "4CF2D5B639E2CD501A6FE39B8AA7DB7DEA924B38692E43195DB7E5F25E25152DF0FB7E0D4E"  
00298 "F63F99CD95F699E16576702B651C29583645070011B2A1F88C947BAE7C94D48EB07A132DB3"  
00299 "8D4FE2B77EAFB31AFB442710BDOAE4E6102DA69A454517B6F148D97DBFBAC7305979B5D74"  
00300 "D7D7568A0CA56CA89F23D8330261025CC741F9D7A4BDB356B544C68C89CCC2C125F5C71E18"  
00301 "C4EA102343AE444F6FC695810E6F28C86BF53F4C8B8AAE46DF6006B1679EBEA790266D4D0"  
00302 "2A2095074ADA634EE60C7070285C316E1F191BC5A88B80D673F144D65B870A65FC93D8B48B"  
00303 "29B80FD58F9FE95F5994878308CAC5394781E4D5A3F5EA2A8ED834EE5BD31D2058C843F22E"  
00304 "B778C4C25144193DAA65F9B57AEC4A344713E9EDF913F3CD29196B42E71BB182AC3B1A60AF"  
00305 "DBF1112A86A20BFC1D28D3E0DBBBAF38E8F12651C207C951654FE8C4CECB6C6F93EC46456D"  
00306 "AFFD7320DEC8D08F2F712CEB4D82407D61CC47B333F69310C06EE1FB5ED84F83945F05D4A8"  
00307 "7CF5A68D78B5536880DE3443E804040E599BC5837E22150C93CC1E5E711F9B889C78C6FF88"  
00308 "2D80857EF41ABC5F12E99105E6C894EC0B796E0A645780341CBD039E8C6EE", // PRN 18  
00309 "ABA759AE16B9D8778FAC203FADF48015331D6499B8CD74BD71ABEBD3E53ED90625E3057EA4"  
00310 "7BE587600F308D38743A686EF6FA189A4D86E4A35EB798FD2307345FBD10FA701265F64176"  
00311 "03365FCC4CE7635924428167115BA372294C27A23CE6C27C506603C5A6618A2B3344BAC50A"  
00312 "B7FDC29D36CBDFCE0D48D088EFD8EA1DE492C543093C30AB7694627C01B334CE3368AEB4B"  
00313 "B3267EBB1096450BDFC2571977D7EF78D6E288FCE0388A041838EC2031248F5FD659C70180"  
00314 "634A1DC7196CD89111C75B51C50F854CEC63DEBF9FFE1AB9406735EC3187276DE7CA2FAD50A"  
00315 "87027956C93B8E84B7C0C3A9C3F7E82B3DB35EB6D2CEBDFE0708FEDD764C839954F2CC9044"  
00316 "B652D0A0D128BD6B9D3DD9740CAE39AA52597FFC1227FAD8B78EAFFC31BE94A632A1AA7A60"  
00317 "AA5A9E09D0A2B6CF6DBDFDC50DF6EBE1D9949619FE9B2302248D6C801DD2D6C01FF8206A93"  
00318 "C0AD22C6990C4EECA7D4BDF36C3246A5D2D2B3982C608E6AD6BDD85C92682EBDC9E4117F8B"  
00319 "7F841239C2A5AD7977E11E4E9CA73A55859EADF7C9C2F1B28A6B4AC7202019230063331FC5"  
00320 "586756CEA1F8478173A0A4964D00C1AC099590152125A4D01592C54DC2555E1BA34C7AC039"  
00321 "394D1979AEA2BF7B2B2A8CB9D62E89132CE9E3B325F023AC6E8117CE57AD4B271EFB0C172F"  
00322 "BFF8FA6A17A490B67CA7B15F865A8AEFF37651A622390E82AFD418C7AFD48", // PRN 19  
00323 "CEA29601B96AD3A83164692200BFFF02C014A9136D9A151A0E61A51F9FC2EC0C3A8F4C83E"  
00324 "64BDE569A33B4CD653C1345B7CBEA3B3AC0411B6145727B1DBF6066ABCE9DAA8B0DE58ADC2"  
00325 "510C02C2619A542A139FA3EF7A03AD3467345D9573C107A13E7FCD43C0D51DB5EC1A09D409"  
00326 "DA75462F9C71F0C9E36C2742C279C910F07CFC5CF7F98AD48D67232A2DF29A66B782095573"  
00327 "57A4BC91922D4195DA9533CD3501F388AF6EE2BB3AD08BC7D5301505998F5B9BF7824D066"  
00328 "DCBDC61CA588DCCF0EBDE4A96632DBA22CA0D770C61A1DD66EDA882D02C5FA284798E12296"  
00329 "E89C4590D315EFD8A816FD869DF869A65DD8BA4E0B13C441EEB052EF3D0FD436E4AC6BEFC"  
00330 "749E0CFC4C7B15599D5514E136ABD134BA638A02E9EC1FE66CC9ACBCE5082C8734196BADC21"  
00331 "F4DA7621D9FA725362C41127836A26CB44CB3851D53C599B94A5E67862665D7092C43D9B4"  
00332 "AD3FE20B8AFACCEDE920F440F3BF5552CFADF0A47D7E0A9CEA18D497282D44778FB7D5072"  
00333 "832C0B77C4C51F4DCFD7AC07DC7A9863DB8A38F1C003CB852F6119BE801AD12B8BC7393B00"  
00334 "640F125C734447DB2FD8B02F7F7FC7A23B84FB80F9CC08E3EF888634FFB6F51ECE9B20A89"  
00335 "941FBF2B49314DBDD67CB7A1B58D8D629FA327AF2CBB47B5419A0A8CB807D30152FA560690"  
00336 "DBAC49D6B043D5BC9D51E82C3B1CF4ED69E997050C65197F3D93E21CBE91E", // PRN 20  
00337 "D358BFC8C6AD1DC94E71D1F5D05589424275875AF8CDA2ABCC6404D6FCB7A2E0A74C680248"  
00338 "27E02621C10D5F8149FBA373AE32DFFF725CF386C3D7A04E3FE10B6F1A6F4782B4823242F"  
00339 "29672E847CCE760BA005D6852A3459E7576A254B10A9A78A9F8112BEA39BA65898CFED1179"  
00340 "D68211D98E6950ED06399E39433ACD898E2F6C87F5FB9D99518EF36429D447B0EF0C5B7D83"  
00341 "4AC8A388578BDF60D4B1FB5A00CEE7D1D613BB9B99E36DC9636E70A543BA6BF0B3A448DBDF8"  
00342 "0469494239D4B7C4979D82E80C8EF36EA67560C86665D458040CE31BA009BCDC30CCBAC50"  
00343 "259E4485E570F190613CB010563F6BD242CF1CF73F6A6844AB8350D23BBC3D1361E73DC9E94"  
00344 "AF83697BB817BA366C9855A754EFC2F007D99A964125682E6F5CF7FBBF687D221B5A0FB844"  
00345 "477A2F87D5370F4469F76073A93AEF7812275FD4F70B2040C12A83ADE5E5D862684D119DCA"  
00346 "0F75AE2B56C794968A68566291B731579A1055A84F083B3072B7BD5AC9D520F64F0829B592"  
00347 "875613BDD81C11622B331289C98501B01EE1D813C0E97CF36878260F80BF88071D258B9DE0"  
00348 "2F3F90B4C12BB56CBC731550B5EFDE6D97A1283EEFE61CD6E5DF312D0F0153A32DD65B143E"  
00349 "C6A3F2B64E2B8FFB47EAE46BD92A6EB9ACBDD11A2D730D027A3EDEADBA5965198FD59BBC85"  
00350 "74B680B96AD48586E5B17625251BF4374E28C6AB956C6818183FDC119499E", // PRN 21  
00351 "FE6943323386067B0EACF1F47BD3AAD9783FA30F684110D1152459233896479D08A976B853"  
00352 "E4B7B52A34511239961048B7C1B9009095327C86F2EA291FAC1734ED2596EF19D04528F3D8"  
00353 "F2A343A0AC19D6A70A37DB6DC034BA0053B57ACB9E7C00ED9BD6AC11339EA169D9D54E673"  
00354 "9B051AF40EE79A1034D6294261E1AFFCD61B9CA5016C56B2D1172D9B2A7283E4EE0A06C814"  
00355 "9E5A2DAA263A5D2429C2B1FCE75C41887DD02E056EF8724645FEC6FE7FC1EF180529B1E894"  
00356 "773CF3E2E1D938EFE9CD824D91454116797F5A84746537FED5F0EBF0583C8508EA07458498"  
00357 "9954EBC4F215B3D515687BCDD5DFDAB981435B07038E0CB869A8C34F916FC67773191679"  
00358 "C60A15A0A399E224D0B0168439386C0AE8F5EF77185AC847A66D934CB0ED6A3467C3B386B"  
00359 "A7F115877F36B49E111DE49E409468F343A98974F4EF1EEEDD282F73013EC2727518DB46C6"  
00360 "751A58AE3E0D5F9D2B966D4465BC5595BC31B2712AE1E1BF9915CC0E02CA7240EBB9A045F9"  
00361 "59E77DFDCADAB6248D58B47BBEF3C775DEFD629A2EED1501A21ADCA470B1AD3084924FABC"  
00362 "DAB6B12FA620B2329AE8F1BCD7CC39FEC62587E58C8AAAC15935D45261E3AFEB60016AFA"  
00363 "0902DB98DCFE586513FF70EF4E3F47773635D475754A158FACC9C470921FB0186BD6EEDFC"  
00364 "BEE9C803118851F82CACBF8C0A544B0562E27286CEA5FBAF83AA5C1F97A", // PRN 22  
00365 "C7386F9FF39FDDBFEB223AD8B856EA2E7F3AFED197A61F183FF7DF2FD6DE208E71E6E1063"  
00366 "FB3774B696913524F7488EFC2CA54E8B653EF5BCB7A8F4994E312DCEE99A316C2ABF3FDF85"  
00367 "B8FA9BBD4366ABB7B3D3D433C14710A95EBB3D0FCDAD237A443D62A8361DA78ACA781CEC0"  
00368 "4542D01DE7B6C6D14CDD4EA709264251D46C42AAF404094286DA5BFF8E81FA2F8C54B17282"  
00369 "1054F4CED82287F29EA3D3AA798C9CF5C5A909B9FBA641A8D9E310248B0F9A1375CE4DAA98"  
00370 "EB62286BA94DFC58B877A73D017B17AFD7F1F58D3D2CAD3B7AF2F06699B08B88FB4EB70D2"  
00371 "511190158BB4928ED1735C94400980144EF9ED06E06074E2F29325C1AA316A46E8E617B3CE"  
00372 "916CFCF05A389052DE120498341EE26A27A3D757AAE763046B8CBC841350292F06AFF97C97"  
00373 "07CE5561F5C119E2F76C137094F62573EB80DC13862797C3319158DD465FBC033CAD81FBF"  
00374 "BBB54D9467599D751B9980A9AE8BFC6715C5EA74859E6A10DB369D5DF83A92655A9A590822"  
00375 "B33B36F55DE563005B886EB324CEC4160F0D18938E9FE41D39234C29E13B814DDCD13CA64"  
00376 "50774800924B0848735C5DE076F66EDC973FC83B13938811CD9887371470AC5DD985481185"  
00377 "F1191EA8C1D3A7DC65E1E82E2318D0FF0C9AF65EA1515DDC536C5A8BD0AF481789838DA54A"  
00378 "39BA56D014E12242600AC78D28ADAC3FFD3600E896445868064D1D2ACF22E", // PRN 23  
00379 "BF5202D3599D2DDAAE5F526B6B6AC469D4BA0D0BA5D79B1DB89173320F0EB68F5D9DA495AA"  
00380 "0981F8022426F68519B548B19B5F8CF068A6CA1442AF77C83B7D8649DC281BF438F9576F7A"  
00381 "719A902A8609BCE9AE9C14B98859B282010A5DC90DC612AFEDF44E0E9E7666A461AE50C2"  
00382 "656BC03648B826CA9C3C7C53B30976335B097C19390716A41FD437A2098BCFA2B29751EA"  
00383 "E5BDBB8192024C20136D2542FD89FB8F2F94C08F765109279BC4E511787496233F15F52D7C"



00384 "3BC3E98A6DC39AFA1818B9533EDE72FDAF021E2C9B7D6C74E49B849F372B1A131F4C532DBE"  
00385 "3B63635E0E1334C87DD6F3D73883D2B43E87CF19E40D6B404E581E807E6EC1A94F5261C7F"  
00386 "7EFD4CF043C90A1A7E97465022ABAA1DC21588FD285E7158FD9B67EC5FE7C9E84029E61E0"  
00387 "45EB5227E4726154F4F057FA337BB20DDA25D11632A7995B810764084EBDE01AF07372EA82"  
00388 "7BFAFE0434401FCFE05CE8FE3C20C01ACF4E9B8EAF4D50C73D5C42A95526DC8313DBCA6ECE"  
00389 "ACBA457D967356A1CC0AAE23FD6261A8943E8FB84CCEC676601A4B302A9CACDEC8998EDC84"  
00390 "7A53B3CB0E12C8B4A7897D5680CB14A3D11BDBF4826C3938EBEEFA0075B6494CC7114D3C0DD"  
00391 "A2F5F783CF23AD2D2545C899867C1115BF4A4F559F63E68098955550BFA1EF7771598EF86A"  
00392 "08C0C634B291674BB77615121BF0838DA96D6E7C53BFE6A58A382FD9721CC", // PRN 24  
00393 "BF8903A3918B3FDC06CAB4EF675F7BE3962CD7E3C6ED643386EE533C3B24A3D94D2EA2CFB8"  
00394 "3F0A346FF2875DB07BA647492D47A807E7FD9717CF12BC97B3C1BE1361E598850B39D50CF7"  
00395 "BE700507863BC4BBF26620FAC1D97128049BD96C5E09DC8FF3F62655D660FE66D31AB0B0F"  
00396 "6D4F8420E3D2E633C571D7FE2AF1CB4E3BEE95E092B00EFD2796A3DEF376F75B7EFCBB1413"  
00397 "37D81AE52939D87956C41B1E42C1CCA4317D31AB4F53DC9502A3DC774E05E1ED5008DC931D"  
00398 "DDB98DFA69960A6ACD45B60895C4FBA2BD8E8BC7DB8C821697558B1E0A3111F156738409FD"  
00399 "180C5A4A33B24C5EE4991B84133CE9AC089724D62DA9D9827A2A04FC103652F216A0895E78"  
00400 "A96086270814C2699F475CEFD6359428D8C505BBE8C1A96D2793802219144CA6B3EDB45592"  
00401 "9B39A3E9F3AB74D685608CE3F301FE38202ADFEEF529CCFF46AF36DC24956A7CD07CEBA55AA"  
00402 "4C89F7913A8A4B844FD8F152C8A823CB9888E3BFEA97D7E4AAFA07125DA4F51D974A5DAFF0"  
00403 "045BCE5B868177A91BD932963451EE2673A85AA8B7D493BDF25BCC2F64AEC3150D8C40C835"  
00404 "AB4F5D0B7F239DF099BD6FA9F5CB198B61018B1448035CCD34E7E7A2138F437490026050BB"  
00405 "E3CE2D4CF4F4F095CB97548E5731A338CB3903519D6B13A029727F047A7D00904A556C8837"  
00406 "45410360FC87877707A716D549ACD6A70A18F9EE0AA8A6EE2080608E10AC", // PRN 25  
00407 "F58CDE0EFE2356F429B0F2F9A7869A142A6173188DD75B570F1D1ECD282EA4FAD11370C5"  
00408 "B4CCF3C98535D27D73C011F11A84711F732441EAEACAB684F2F0D7FD4FC4070749574922A9"  
00409 "06E84B3350CDE5957DC388FDA23BF45F05951A393DA253EAF691940897B57ACE655E9630F0"  
00410 "881BFC354F37B6D59C566823F629A21C973324F7167BC39FBD2C121D2A849308D13DA1A289"  
00411 "4AFFBAE677B05677D60DE1D975A4F3D7EB3461B424B67B61025AAC257A69FF720CB9DAC007"  
00412 "C50C69A7ACDBBCE210BAD4DC2E629A039D987E9A037A5C344B5CAEDCD0A35F28677A41D55A"  
00413 "0E3C6E480CCB12B8F17062A983F4E651B4F7CB217FD06BE46747CD5418C80C81916465A4F56"  
00414 "60152B3E4781DA8040D4246F9BC47366BF663CF9DA3BB247D9238873CCDC6FC62D1D8F669E"  
00415 "FBA42527112FF4072262F7E65AEAC328871DDF47588A0A0DD13A4139F4145822A5917F624B"  
00416 "49664E08F2BE0B93C6E3B89061193FADA0FA9485F62CA87F3E68E204186EF1187642D65116"  
00417 "48EB59F7DE97E364223E17A30119BBC7F43E21E7DC3093F7505055ADAB4654194A77C1CCB6"  
00418 "1898840125455A275A8F071273D8C13934915D379CC603657D99CE4075C1F1DCAB60B6BD62"  
00419 "ABA1A10B5402A59706798002EF30ADE2F354E38CE0B57900FDAD31E7F684E53D097B4313B"  
00420 "B552EA66F6D337F29594470D3DC0BC6CD361831251004DD3C5357BC0BECFE", // PRN 26  
00421 "D9086F7C722AA317C64C00AF43C924DB5DAC97F8EE3ED2296252FC4756FCE6928BB009D448"  
00422 "B99BAB77411BBA52BA6F61AF1181CC7BBA94257593FA1BD26D52AD5014C3F1A1832FC4F74"  
00423 "45C8BBB77C8FD31C88F0C5D4736D49DCFBEEF2B8301E31185793BFF87CFD9E6F7E084D343"  
00424 "AB98BA3518A7A5F915BC0D76B01AF7DC1CE45F1C5280BD39D3E3D94D0A0286F8BD9FA9428"  
00425 "49664E08F2BE0B93C6E3B89061193FADA0FA9485F62CA87F3E68E204186EF1187642D65116"  
00426 "2E4D8E7DA049F462362D8C94539CAAD09AE4768C96ED6C2CAB8025EBB6901CBB26865E1F19"  
00427 "FA1B193D47ECE390B881233578950175C85B928582D5B439EEF2F56A8C7EA09278E4774105"  
00428 "1223AC182456C4FA04D025BDB33FA10C48C70EC91BC709E3CB0FA3E01DCE5FE5ECB9018130"  
00429 "A8DE5D0583EDD68EA2EF227A612748B2F785A30A01014BD479DEC2656C8AD88470F79DE0E1"  
00430 "432CAE448DD7049E5B7D4DF3C978F65E708CA3759AAB9D329C11FAD71204E1E92322E3EA1B"  
00431 "BDD9D034E2A32ACFA21CF490AA5E2E419197DBE990667BCF277ED61B264632F694392EF52"  
00432 "F0A27C38E478257AEC8D2542938BF0713EBE60779C95A0EEC8F32A5202A849CEE8CE0F9970"  
00433 "2F595AEA839531D4CFB5F5A6166B06EB64387552A1F9BC6BB97B9B99D19C3D2E1E8E9B305D"  
00434 "525E7413496E40FF50CF77D4D4E2D41B1D5929848FB2F1FDDA5A39DEA0546", // PRN 27  
00435 "AE4E3B30560A50DA55AB3E59FFF512844A2700D2D763D85D5C3FD8CFEACD4D023BD926D3E"  
00436 "F2E55EB1B3831F2276EB07E5C07B44FD7D79333699BED0804B678915FE0F092DA9A62F69CB"  
00437 "020DA21932F9FDF9AF332E1B400C6B7E7880508E840D62FBA07E827A23A2575AE68E15AC44"  
00438 "4A1CE35DF3C3FF7CA49DEF2966DF3BA89C8E90ED5E2421A6407F2EC51A3E92A3608FCBD6AD9"  
00439 "FF9E5C7817E79A0C09FE9014F7AC291448263E4346CBC4BAA6EABFB59B4526B654070084F5"  
00440 "2B864F976918DC6EA91B576956397CE55CCDDBE41F94E5DC366E775C86ADB1C807B66D086"  
00441 "96A2BEE45B90E8736469A371F05929D9D9FD34980DE08E00BDE2CD0EAB6AF2165D76519F8F"  
00442 "2D894AC70740D2372B37407BDA4D943EDF1CBD35CCE4D81340CC97751C568731C009DF6557"  
00443 "1F28B7F58106AE67279E83C3A0C130DE0C5B6C99117099548661D290C4CAF3BC60EF719E2F"  
00444 "7B210FCD4381C33904AFDF96DC3A6557B42B6EE895B4D604F5F8985F454C51E32B2C874E90"  
00445 "926CBC58D044D483D6D2A7C26C7AC4D190531F79993D07B2E830FEB99BFD00AE8C008DB1B"  
00446 "762F3F4A81D41295FDDA37F3056B1110D4F0CF385F9CC7E14C34F6752A2FB17F5CD3FC4AF"  
00447 "0D51E4A0AF7D28DB0D4D651156189209480054F8287266B1CB26C9E8CACA0BE5A69C69630"  
00448 "0025D16CF9DA29F9EC79838941459B7B8164AD95577A0C532EC2EDB35250", // PRN 28  
00449 "9CF0CC00B5788DD743A5F33D87E8FA5733B72EDBCD61AA4B8D0B81213DB52E7EF17AE90934"  
00450 "F5EC0711ADD19E881CC330F696179C1BA464FFE6D7B04EEC383A4106BE5892C5DD1BD719AB"  
00451 "3739A909A384FACA455E6AF96600AC6FF809788700DD2AB93DD228483759BD903EC002D4C1"  
00452 "278808B764F018E3B740EFD821A61F5BEA2948A653041FB31F6D5D0DE0A045DA366E44112C"  
00453 "820FD7FA966B2CCFD5A6816AF84DC0A3EEB8F9D2F0A912586F91D50B1AE3D930A680A8FB74"  
00454 "35B6875ED2E599B87598A7C20245296C4965E2E0CF372B6ED1219BA68CB646D3E73D52665A"  
00455 "AF2E3D1C4DE8D264578299B166FA0E148281C877FA9B14818759CBF7FF575307E80B739335"  
00456 "99D94EAD2FB1C0A8A30006330BF0AC1F1C0A4EE6B07F9F3381AD7E2E469E8DA9C2D22CFC0A2"  
00457 "08B5824D2F994AFC0268EFE206E0A9EB79BB51CA26FB49013B9A17017E0C08F9FFC6C319B"  
00458 "B1B5AE41771443BC670EEB91D7769F9890A9B80F52CB0167EAAF850FAF2A52B74ABB1792E7"  
00459 "CEFF68CD03BB01F244AC0CC0EF0731E3BDDCDAB89DF376973A7ED5D4264EE82C334671FCD3"  
00460 "9ECD62CF869493914F332767BBE461707166A9164776D29F5EC9291F505AF291254D7319A"  
00461 "A594B5F397D5BDF00BB840C4DDCB425F4325ED8AB77E57BECA3441B89414616671692EA88A"  
00462 "89D2690A4B5FE958F990BD8A4388A460FADD5A57EDF01865F85829195460", // PRN 29  
00463 "B85B6E754CC8F6805A8A19DA104418D9C134C8B0DBCDF5DAAF5A71BC047A73BEDBC192A453"  
00464 "674BC624959BB76E44C5B34244D4736ED3F0F3C9658FEC0DA5437E01E128795EDD7593D636"  
00465 "CD73FC1780B37A381502633CF2EFD0A0BBB494C1D0FC7F602DF8C282F55E382E81A92458E"  
00466 "B16B74835040D8A9C8F2DDF180A617B0592344B4373E1B526C9706B843B0CED4D25D7324C6"  
00467 "FDD0F33133C00443638E6249061C56A116CEC7822F4512AFAAE52CE8F94D8547F72612EA8C"  
00468 "7D160C65FA3BC92BE01493706EC4E5F203F0BF85C52F417BAF8AF490E50133505685CE63A"  
00469 "C5B173E07D8DABB2439C6DC18BA41B9CF37D02C92AB5C2F27EC83AB6B2DDCB7ABCEA30A95B"  
00470 "BC39E9FD0CBB28118823F7D0342F1EB7B45FA6BB3A50223D0D7B14E975E7658352BC9288B4"

00471 "8AF1346955F4551F2ECA47D423EFC63D20681057E5EF234D061A5E6E234ED01F3DF223A0E8"  
00472 "B4DEDDC552C7DC3ECF663D5011FC907EB4A7CF746AB9E07C2929B7427DFE9E00B0A1308881"  
00473 "912635A72EA99927F343EBAD32436A9B8EB1934AC29E79BB80AB3ED9F5CE39D1E43C251564"  
00474 "654365DA43FB8A0FBA27F2328D82445A1EAAED67B92716147E859064AC326A42DC7880DE82"  
00475 "FA782AFF9C59FBDCE088746F8CEDBA288BC8C2C4B458782CC9BE63A86168B671BE99A09F2"  
00476 "217B7BB2A78C88651C1BCE8A0B89316ABFE72B2272273AF570974D8EDEE4", // PRN 30  
00477 "DD40DD438251E401FC926CC6968393415D52521A5BB34D4272D6BC7B5431062B35112CA70"  
00478 "9C0680CBB18EE053AAD62B2391C9E9D580562541A453ED936CE8E88DFA61A88CA3BEE66CF"  
00479 "FF801785CCE863ED9C36A04D2DC8742A81CA55127B44314AB4E687ED921B4881CB363AFB3C"  
00480 "CE7EB774E3205D4591939ED7D3C0C508A31786421F49669E120F01D35D467B40F85F2454F1"  
00481 "3F591F3B830937421B5C8A6C20EA878971AEC941FD99CEA92FEE00E5DC2264987DBC549EFF"  
00482 "3E4A26AFCAD7421C4256D107A3E8908F67450960E4E41FD7E2E84F754BAC81C8F5F1D6F65"  
00483 "0DEB3E6EFF6059836643209E3880D7BDA701869208D8E4BC8D0614066414DB3F93D6EA1879"  
00484 "50285F55BB7A1B026EA4BFCAB4671B07704828D5CBF9730EFC99E68E91F1FE9664DFA73297"  
00485 "F2D6BD9497DE084982C9FF3730B6FC3EA2053B3F45DC7FB587BA19B3C6B7E780EA5F25B45B"  
00486 "B727174D4CD3B401FE1906360BF0B15DB13B62752F82EC62226AABC83C1C26376F8366BB84"  
00487 "9DDB65958AD969B25654DEF1841518993033AF47EABEE3CAA936F19E28A205F3CDD56CAC6"  
00488 "49DB6A90483AC63A24EA46D397508EEB5DA94E9C883EB0451D036E28CC303D52B1BB31FFF"  
00489 "582605F340D449508959ED1FE2FF0BD22FDF77F9680D6B5647D59E7E6A003AF0C6A95092F0"  
00490 "DE43D1252EA6DE00F288BCCE3ED9CE273DCB4F3BA7E8D17353B8ECA24F03A", // PRN 31  
00491 "FE32B1AFC36B4D415F3FDD4DF0E0274FBEFDA0042BB203A4F6627ED9E29F405379B2F2DDC0"  
00492 "F3B02A0CA70A9499F3CE82B87603FAA347B7052CB5D13D9DE84C114EF3B8F62418FB1F3E37"  
00493 "4B997127667FDBCA2E2F9DBC04ECA9D908CD37C62F08EEA6F44B3FDC149465AA8037D65A6"  
00494 "C8B9B83D5E9A40578E5EA3AE1209BA49E5E2AC615C59A2D71AC1605B98E39A5E66A890754"  
00495 "C7D1C07E06DE78632587BADAF7FAAB0A529AB791095DB0A708B691E9D81F2CEA8F07B05495"  
00496 "528B9F056F77A4C8209DB972FAADD9791BA59F47C06F241F50C0619FC04F8456339E0AF331"  
00497 "310FA4CCBBA05DC2795CA6B3ADD0174AE4B330AC0428320ACEAFF68F73ED11DC1BC9F0237"  
00498 "BDC75F7F48BE518EB3305CF2BB898B329716FC9ECF7E99B510B3309808735FD0A77B15731C"  
00499 "233998F9ECEF46E2CAA6E6EDC8D05B943ABD17027A80D636E535038FAE44D60AAEC540A637"  
00500 "2D62479192FA84D844520C6774CC589FEE16A3A5549495D968AABAABFE4DB94F5AE0C54E60"  
00501 "3D6DA5C3056769A064890533EA8EA1E5D1CD410CC8DD4B1D7E0F5F787232439AA4B3911C5D"  
00502 "C792ECB873E8105A1AA61C627BE57E809C6863073E1E19AD8B987DE97D88A817FB43ADBB77"  
00503 "51E36D1F0E7B70B3759D6EA8F2350D10AF38C331E22703B2B5103C908E1D35A8E814E45BAE"  
00504 "81DCA0530FC3525CD640548245C259738E749E195B006081A18C45475F906", // PRN 32  
00505 "961408BD52EA1C6A9F340D9109B2388CC358BBA2D35BB6AB672A9C16F3AF968BE4613BA6B1"  
00506 "3D115B896BE71CFEC4A4AFC0BF5D2BB1B5DC19405F40E60FDF361E6CC362CDC28B75B8C30B"  
00507 "468D3BBB77F3FC62869FBB635A3F7AEA63420CF1B80A4B3813B2240B83ECE999808E1394DA"  
00508 "2881DE2DE62C80EC163AD8D81495DBE2C7383B78E19AC506AED9F3BD1280A77F2D9C55600"  
00509 "BFA25876761391145D45F74253C6E14BAE16179884F0F0EEF8150A445BE1B5AA4FEDC2198"  
00510 "CC39763A3DB473C1CC4263CE2DD587447DDA6BD0A496E8DF60859CFCDADFED2EEC5B1E77E68"  
00511 "F449ECF129AA17395BA39392EB610DF45134571BBCBB26162C83FFE77D188160EBCF598FE6"  
00512 "F092881612A04BA9F4215D429BB521E737F6C3048B5D95B20AC37F79AA99A12CD046926822"  
00513 "8463C317A1F31E63E4754890F20B4516D179342A76201402BC0DB2AD091A70B4232473343F"  
00514 "E9E0002E59044C5F3B6E3D7368DB7E7F9F42E7A4A1942B1161552C5C84390197A54079F570"  
00515 "B57BE8EA8A30659FF5A61048142E4368D5542B968966E54DBF91D3970B9A3A278B951D6F914"  
00516 "246FA5CCCC20DE53713D4830AA86D758ADA7A8747FCD455B2C320FF9E5E7FE1CCCFD6F7928"  
00517 "884FC0B35F34118B4E7E6F6A5D6E3FB4E3DF90CEEBCFEF9D7D595A4C456C373C5356EF0DB0"  
00518 "E6F8DB1E413F80AE4C32532BAA39170AAE2FC606B7206C379C4155031AF918", // PRN 33  
00519 "E03A069F63220109A77232F2BAF29D345737AAFED2E5E8D5C846B937277DC88392DB2D5507"  
00520 "3CE063F5FFC5717BB89B481B8C86EC01808A85ADD78517CE12DF776CD3F02D948BA7952155"  
00521 "99CFCC09B4F56DD681A59C71D24946225171DC18E47BFA9068302BA428929790EE62306E"  
00522 "FC8C10E71F6F372105C9421A563A4DA704E01FFD71E46B28C6EE1B7CDF95BCA6A794E8FCFB"  
00523 "6BCA6AFBB67DED5C1267A12A4752176C3C1E6C2F665194C7F11C1CE6CC8481A5A966719B57"  
00524 "124CA33D8EC9862AE5485788106562A0EBD3980AFC4469BB1FD1653678192B3022E26CE87"  
00525 "88C68F202D506DF098D49EBAFF7741CA96A02BD68BFC4FDBE24D34770FBBF8EAD9647C911B"  
00526 "2E6AA705F0236301162CD2B41B88BDCFC2A79EC54698276126D6EC0213931609985224BF51"  
00527 "5AD3221FB5E8A1CAECED51DB4AFF612157ABA6640866DC6D3602C446B8C6EAEF359BC8703D"  
00528 "5EF9FE8EC7F5F2685C53924C6F1F71519E0AB589EEFB393A12C06B906402FE0A2FBD793272"  
00529 "BC101D3B57F10A52C172E505B74F2A00BD5D2F7C7EA2883908434C125969A62F1B09F42E57"  
00530 "8BE22C08E8BB11767D97C4C0CB2BBD98508AFD591EB0B0C846A95FB72159E561F9D87DB44"  
00531 "6E9A19AF378B6B4DCCB49FF8DB547B3040FFD0171B87245CF73A0747287B6FA1F1E4AE995"  
00532 "18E8C53DBEEA942AFEF75B69E4790D75CC8A5181C609BEFAB641E28C07082", // PRN 34  
00533 "D8D7DD9242E54C6DCDB2A717A6F33226A94D5794FA0BDC401F4ED842A5CFA4AAB462F70323"  
00534 "9F684DB9B95E5101A5DC6067C7062AED9259CDB2067BC815C157A7E150F8557C3A54DDDAE9"  
00535 "4E5C569A0F0C938A062B601F920EF4D9574BCA8E329123BD6FDC3B731361864CC139EBE3"  
00536 "C68FED6140FB127D9D1DC071DFA1552ABC6703494A632AA1314984A2D4B7A8BB32C555B79D"  
00537 "EE013A66745AE15E8E5E4E129E44A119203425F2ABF1F03CE9CB33C0BDC3285ADFAAF4D7B5"  
00538 "1EC31F02D1E654BB10F0CD97EC3E389CA34398166B4D5C9FCBC6E957FFEC9CEE4071F90EE"  
00539 "2926FACCC8FDE884CC6FF1F6EF1D33CC681FB44E45C5CF681AAD13226DFF19E22CE81265F308"  
00540 "D1293EDB098988B2640EBD8D9D66708E1E9880DC41A72FD3D64792B14AA18C13E371CA17FB"  
00541 "46B21DA9C959FFA2FD075852D42E5A78744792EC02F9A35F869912BBB44DCC648CE6075DBD"  
00542 "F457A9AA891488A5450FC719778739AF323E87F9633E621B404F70614F77C65697E71F2816"  
00543 "75C843C523740AB66756E4DA784F555B5B4D797A06A0AFD35A69DEEA6E948B9B3C2A62D7B1"  
00544 "D56832D9CCF56F2680A5A0A3037E4E0252413FB86520F2815C8824975634C0889A486963A2"  
00545 "C874638559E14F780A7F3E2318B88B2C2010174150AFD4BECE2C5FF2D37BC2FF791F4A3136"  
00546 "E19C373FCF03E471DFCCED19471182A93E91ED3EA68C402234AB2B00ECF62", // PRN 35  
00547 "87F0D84B811E4635AE8023B74306DE789ADDF1CABAF5F47885CF7F9A33F2C533093A339EB0"  
00548 "D5E05C0763A215459CFF0D31CA92573EF2074CE2B6A11FDD6BAA3C6FA100D23A9AA413BAAF"  
00549 "DA22F746CF74562F9DB0EF2F7CDF266142F177681CF2EF388E2E9AA012459ECCA332B779E4"  
00550 "8CA44E536082D59C3951ECA42F66B600D2621BF5F3584B59DC0DB76EC66ED7D00E943BFDE7"  
00551 "03E7D5050A8F263366948253B3576311CE88B463791DFA6F401ABDCFD7FA44B158AA97EF1C"  
00552 "FD7A8802662A33CED958052DEDB12A6FB353BF2224ECC232937DE3D39DC8F2650B18FC0B"  
00553 "C2CB8CC4561413CF3FB09A0906AC1EB77E50E8D260041E4B3FF4BF3A537FEB62E0FC503E8"  
00554 "E38F185AF795F67FCE17665AEE29BD0D5521024A70B61446CA4CD2B3DC274FBE72A9AE29EF"  
00555 "67A2217D6CC81BD8831B5160E4E81238B379B657FFB49023B040B6D504B287F684A0A0C073"  
00556 "04E6BFA12EBD0A7629BA32F3F2BC9A33DEE2EAC1A2C22462EA0D24162543945B78FB6E26B8"  
00557 "6E12621588735B32A4F9A50157F20BDB7A4B6D151B3F28B40A03CFEB3CCA635261ADA2295B"

00558 "74947F1B1D676FC014AB362C1F5AEBB439DB137D034D00591235B6F8D11C735A0C1964B29"  
00559 "D3002D5243374628FB488A04EF245E64F598EB6ACBDE8B87F2FC5D1ACD105460C26BC6E1C7"  
00560 "DCEC2E92E33F722E0A613A86356343EB111D0B166AFEC5C7A44B81A607A24", // PRN 36  
00561 "DDEC47E0FE3E2F4205206CD673EC66D7435E8BD4A523A8681ED77F51453B904E4468C70C22"  
00562 "24C5F1D01A1C5CA89BCD72673F82955FACFDDC4DA499FE8D18A2BCC5035B0C69D095AE0E"  
00563 "F1AFDC389B6253BD6FC83D3C37E809EE732A87D065207684332CBB9BB0519CFF51B553B368"  
00564 "9246EC1EA4E2E36773A5AC4A7BCD37381615F78A41E5B181AF5C502E22FE79436D6EEC07F7"  
00565 "FFAA739356CD9544C7DDBC42C56BB1C965441903962703986C93C6F8135A8EC42A89DCB46E"  
00566 "754EF5B5250E1171398FD31793594831B0775F2A39E63FAD99929A0F257ADF332C078B0B7E"  
00567 "209229ADA46D69512FF2DE930F962B8F81EEFCDD358FEF8E4DA0D49F39C43AE9D99D8052F"  
00568 "CD60305FF4D9FD4CAA3FA6BF258B867E3F266F1BCC77390D02132E370AC79B34FB37F12D1"  
00569 "02985950FA5FD53D3D4783DA5B284494EBFAB51DE6CE259E27712A0EB4D78EFD6573A03D62"  
00570 "9F29166B902795FC8DF59050C9FA48584F0EB69293C7FACEA8972098E1367D183C0B91C355"  
00571 "0852AEE6721A84341612E820CA4F53C71A51774134923FA00FAFD84811C07620917C1966F6"  
00572 "A26A032B125D76BFA149FD66EBD18F7536285CF94A8A750680D6BE0F4A4E7F2433018796A3"  
00573 "58CE465E7507F186752EA2FFB01941902129936F7B18321902780E2C6A5EA7D3607227D631"  
00574 "5C570815BD0E808EAE242B554824ABAF30614CC72B74E647D5B5A6306632E", // PRN 37  
00575 "DB29CD5101A3FB7A651BAB94E0CEF2BF737BBEBF0F755AA42F1C0B2E6A2E00422458CDE22"  
00576 "44576F139E811B2469E1B8F10FD95A988FCF872607A9B5CF81EB5875C427515D8D6D4FC58F"  
00577 "3C69A92AB4932ED3D1CE6191E648276AD746F12A0D7B1ACE96B80CD2C3B99BD1094ED29F70"  
00578 "20121076A66D5D750C39633B00F3DCB4793A27B4838C492EFCFD9FD49D955ED33814C954F5"  
00579 "8B8D29A81C3C62BB2E95ECB393F06EBAB479494628027CD0E59F1AA32F78C53142137965E"  
00580 "662DF4B3EEB0B92A6FEBDC91F8B31D41E2EF69ACDE1CFF2AB49E2A4B12C275A20439E30C69"  
00581 "DD14D4F661C81308F11EFA3D014009D80D794D9F2C8B51E2E6C8368DDB07D1790F56D7C8E"  
00582 "E495657BDC081A63B1353F4C4CA74CC0D02CFD7BE60E063A33D1A4D9050A9100F0E1817047"  
00583 "99357532DE152CA54FFC089E8DF80F29DFA14C19C9D6B7855482622504F95A8948A5DD640E"  
00584 "E88B87CA988248D148AE4F992B7D1FEB3BE07E70F2CC052D92FB26337528C41ABA642"  
00585 "C6B1C6F8883B21ACDC69196733FFB684451D188107C1DD219AB8B0AE5F49628A1CE32FDBA4"  
00586 "B8CF02F38513E37542D020F4BAFE03EAFD7DCCAAEE9E28258322E43DD47DA3AD625DC2795A"  
00587 "DBBBB9FE061917573936E31DE6355023F8D74500D2D032B7A53630F37FFA80567BA36F771C"  
00588 "4ACD71D76D5E28FF47E8281ABCD325E4C4620D0B73B2BE2787A7F6F485492", // PRN 38  
00589 "E8F24D8DFB54C5BF909CA2CC5496AEA76E583D2D865259C356E64E76793290BB00E7029A14"  
00590 "6E1E90DC0C45356E13EF59D60F20F080A0FEA743FC1C222AAEFA3E55876F2C9E6F7BD29C09"  
00591 "AE32E80B15DBA0E6B594E951EAA3BFD166DFCC17AFD2621EAE6539C74FD776BB998C24C30E"  
00592 "DB3A8B6814DE088E7E6B7CE9A64EEE9ED8C9D987A21A3BF35A9D59171DD456D8D0D7D1BEB3"  
00593 "F37D4A31812BE00077F0F0064DA56EF9C4D36D1D3CAB4D1D4C024665BBE1DC227DD29E1CF8"  
00594 "14EA65B64DE60ECA4AB9B19F937DF15914F3CA577EB3A6A0A2C8ADDA53DE536721FFFD519F"  
00595 "C5E7CD9A3BF4F030B1EB638D6A0F8C4E24085012D758414211585EF6E0DB8C670064885B67"  
00596 "256B8AD7E1D2E380C5E25BF58EB1731E935899C1FD2B20008CF87529AA3E714BFFA86DB8B66"  
00597 "AA94811A43ABD868A7711C4FFB339C01D72D4974ED53E7DDACCD36B5F459DDC05E9D475D3E"  
00598 "2AE383F6FFC92CA0D1791B4675DA843303C96A98ECA88B54735293C4E1906AF30221EE71AB"  
00599 "58F3E38271608DA6CA0A488C0850DD86F8B6588C91589632EB781637C14D9D24B57ABFC301"  
00600 "790B3C11C1B2938845F08F7280388E0B9D5A9682A6FD40374542634590F336E42D8FDD92F3"  
00601 "F6C82BE4D3B953CCDAC984F6C94D8F2FF70BDFBA63C922060C322FE34188FF70A37648C362"  
00602 "BB68B06D74A2012050FC007F276A54882A8A04DCE014655AE43E448639F56", // PRN 39  
00603 "FE35174BF6B586B7D39271D92AEODC2B64FE31AAE8D1135BD8FD308D7E281F3ECE84784423"  
00604 "A48C1362B5852719023F8861E861AD8B22219259E357EB9CD8050A66EF7CCEB53636B47D38"  
00605 "A2AF5506E1FD72D3E30A29EF897C54A271EE562D67B279806E8A5DCD78DE538D8121CAF4"  
00606 "C217F8A969AEA50D6FAC68066277242B1979F1A6B3051CE5B9949A11719F556EBEF844C808"  
00607 "E5C1AFCE5E312C53AA9DDBFAF7280A7FB8502D2C7D1B91614157AE0C6C962F868D05D04631"  
00608 "31DC841169946B732F8000E686467BD5D8086CC249693FF9794374266BD6A5C8AFEC65C66A"  
00609 "834012365D60179450C58FAFE724B8B9E99C33900A86649B04CD54351D6C4CC7068B28F070"  
00610 "417CC9C4430390493BEA50799FCBCD7A13BF75947C597B3D7AD486E3AF99CDD743B6230BC4"  
00611 "73DA356E05FDD17F7CB8D04A20E0FBA30946EDA96BED467A45EBA54578D9001637702F1"  
00612 "DBAFAAD16D2608C475B8DC7994DAB93FB34DF2237E4A13D0C04A6CADC42FA9463674AF8ABA9"  
00613 "7CF511A82E8E61F8330004E165D753323F4AFFE598E4D108DD8EA8EFA45693A2F9EC8335C7"  
00614 "56393585E052FE5D150A58F058FC8C3720F37112DCBC6324877E87541F06C968C46FF846C"  
00615 "B512EFE65CC401174BBD1C977694ED7BD1702EBD2D1EF23BB56D0D0C1EC2D90A27CC63F0A0"  
00616 "9D83F8B0A5ECC181D5D4265911BEE77EB1DCB4780B53CE74DC7A42252F3FE", // PRN 40  
00617 "8FC5496AC76BCE7284BB32C63FCCABED194E82FBBBCFBDPDC6AD09AF95DCE6A5D6E13CC06"  
00618 "B247E38253D359A6C0EEE00A660BC3BE6FB217B9B554C90186711B1E85117DF0CA17463E13"  
00619 "2333B8469ADF72C3BC6F218A96697E172CE2E6D6F4E70EEC2A0CB48F0A2E7991B1B90A8508"  
00620 "8CD2C59E3C9AE7FD939592C14AFE9A13829DD97E345D1AF92AB46BF196DF906BBAB16F5A5"  
00621 "8A4D99C7D586869C81E182352C210E3B1EE822F4A95DD0BB3B285632978B18FC1CC29BD58C"  
00622 "2152FD3BE73482DE1A6A79C392DDA1D20B6E99E25A666AB20CD371A85DD20BF0A76C5A810"  
00623 "41EC943A89A48A64C5207B3166441FFFBD4AA28A2A88173F280F3A2838A98667775F597D"  
00624 "0368DC0BC53C1AB8B7670EE23E0ECA08CB09FCA68FE2153F01FCB7061CA1B6AECD815D611"  
00625 "FF71868F50855054085AA1B7CA3309581532D658C7D2D6069C3B44E0E3CDE4522DD9EB40C"  
00626 "369762CDDAC6D9A6923FE0EA7DFA1DF73BFF8B60EFDDB8863538A38528803F3EC27E09C87A"  
00627 "2D0E160C2480F7D2003DA331FD5C7FE05B582DC9AFB114D2AD2F822922067F1FDB3261078F"  
00628 "33879C497035003171165FB139F79BC508AB9D3250D1B53508410A01C35B53179076D9F46C"  
00629 "5BE1A26DEAA2F9F71F442FD7D19D34979F6ADF96BECF1124551D4806E7136551291352748D"  
00630 "D2D58443978C3DEA5ED0544E6596769A760B476B9CE7BB09543991EADFAD8", // PRN 41  
00631 "BA63AF12FA5F7D03F714CA9AFBF19375D2A7E31EEF4A9E10C7C2F8650552A6CD22FDE0E012"  
00632 "D46CEBC773C87729C7E746FA4C0361099D4876701C0B2A16EA2B5A6B750DCCEB573DC711F9"  
00633 "5CEP068BDF516CA29FBE6387550F22502E53A772436F324569B25BBACCC781D30DE25B806D"  
00634 "369AD1EB1D1137EACD0BE765DF4D06E177D3E3F13E9A3165F269244FE8B130E9066B23474"  
00635 "418A5191ED481BC8974074336E71BEB02B1BC34CC620BDBC24CDA8BCEBE068416E5F5B5A26"  
00636 "3A51CB3F6589F77D4E607939F7A0FA637858B96189A014B5688A9DA32FE2ABFC31FB00401D"  
00637 "E475BA07265B3FD1890ED0FDA487D7C616E099B4EF4E47C9F9114E6BF8CC757C92C02CA654"  
00638 "6130ECF7CD44160F55A72831692A5E69146086C43A5F043BD1184965E2A6B154BD7B8BFB3B"  
00639 "4D28C081B0F349900492C703913885428A82A8D2EF1240414CC0B6D56D969BA1A121D9D615"  
00640 "84C6A880D533AB58B94C85F07732B5EBEE7FC87FF3AA279ED703858957A1C1501D71DC5420"  
00641 "C24BB2D570F7589F0AC5B6EAA87AF68442FBB38EF693D2200E73DC73F58FCBBE43FFCBF76"  
00642 "DB4241FC4A7B131F29C8BC0F77C95843D2FD2DE39E3D4D44237DB39994380C5F1A1A3AFD9D"  
00643 "7F6B736D585112920296B90CE31D58BCAAF8CCBCBF15BB36199B48F00F308F7E8264D039D6"  
00644 "DAA6E848CECDD1544323C9654232F45689076B7C4B1123AE0FF152124E930", // PRN 42

00645 "F2CE3D09E701F60716A413AF06E14E6FB7BBC29DCCD8273A083B00D429B57EBE9FD5270EC0"  
00646 "C299D243A9FF0FF73DE19977EC16C2772C6B7FF88F83481924C921F49EEEE41121DC79B7C23"  
00647 "255EFA50292B11C7D45D66BCFADC3C3A390E03E13CCDD51309BD4061782273CE5D0E1BD3E1"  
00648 "1271A3C52654D2B846B5600D3E68D2C234807BBC32F1A350839500DEB6D387AECBFF534460"  
00649 "6CDD04BE3614262F068FBD256B855EABD5A7C0AB5C28D5719C844F2A6FFC500628CDBCE869"  
00650 "D2F090E588B57B796029A84E2319FB6E59960553B62BD30DE99AA7ACF508168EED8D8B999"  
00651 "6041BDD78BAA2A1CAAD2EF366F4A3244DCAA3F06EC9A24572CC80C3CF44B557CDCF6279165"  
00652 "414066D6B700DAEA8D361034D9DE455A55E55CBE39898B65651FE709506D1A1FF67585D0D3"  
00653 "E1B2C5EB599704B3925CEFDA45C33D92C9ED9F0C45B7A80706E6629594F66A0F74A4A76749"  
00654 "3B73B23AF323D519B0D05EEC62C6123A0829CF612DEDFB7F275A55F1629DC64D2F77125A1B"  
00655 "DFD7B9C213C51DD9FFD83DEF42E87AD4E0F9E5ABBA6B120E131F9E5A3097F7EC766539C73"  
00656 "3522FEBF0F02604613CA1190024CC1C3297E7CDC3C514F8C3787943EE8CE457B516677437B2"  
00657 "212A19629A0A04B0DAB8258A17B28F52BCE915A680152225304301FD2BB1BA2A557B3E3CFE"  
00658 "6553CABB7F9A47FEC31DA590308D156C3537B97116224F3E21EA3841A505E", // PRN 43  
00659 "AB2F844287488AB6B0F47218F2A4C54E6BE79A80F1209CD747FD88A575EBB11F2F0756B2C2"  
00660 "63E753D9388876E159EF3BBF99448D0865173572F4B6A03FB72244334E4E861E0495AAEF15"  
00661 "276379F8DD88AF9313096805D5596446B48EF1F1BF2ABEDDABCB1FFC98ED408846D85732F8"  
00662 "07196C9CC9B283EA4E0D78681C73CB6AF89E5C361476F84E979814F30C0AB03E97156015A4"  
00663 "93E091EC5D854E8B08328581D80091564C25D6A714407D6B591F17D376953F18308B8BC12E"  
00664 "AB5B5C9FC11AFEF293109785CCE29C978955CAE2601A1DBAA274B72CC18CF27FE077A01232"  
00665 "58A1B87E581ABF7A22A335FA1A8D11C2719F07CCA23B1DC5E5B9AB198077177BAE0B5F35C146"  
00666 "DC7AE3849C7C47820126FC99B9766920516AD1D63E7EAA366005714D6C1695E731E88D43A3"  
00667 "8B615407A99B32A101C4D417D5E36B886C26EAE1CBA9FAB0C040468781897145489121EE47B"  
00668 "B2FBD6A064CB325AF5CECE5D5A4BD590C70C7A9B068D318247D8FE16A3609C94A8D431E987"  
00669 "2E26A3CDB19DD971971AE1FBEB2FB76F7A71AFC5815CF7871BB5EFE3BD0C49BAC441ACD9CE"  
00670 "1DA4AFBDB972659D13775AE77F843259EED57A62CEA93FC449EEEE1EFFD9AA2D36D8FDBC06A"  
00671 "95A6AC9CB06746BF1D813F2DB95D3456063D5B2339582438D5C68D3F869C4F639962D599AE"  
00672 "B26FE2D5148A51579EF385D31E744450E11307F0C803C70ECCA93F831EBFC", // PRN 44  
00673 "9B1066F98BA2299266D8C1351E6945D7EAA658BCEAA4D5AB353C2CE48D01E915E740DC90E9"  
00674 "CC487126AF9FA722A335FA1A8D11C2719F07CCA23B1DC5E5B9AB198077177BAE0B5F35C146"  
00675 "66FF32926AB0239F21811921C77FFB56F7B218A353F8EA67448F61ECA318196B1186514D27"  
00676 "416EE2F61764940A70300B69EBD02719191ACAD969B1075EDD09EAE26F883AC99DED1AE935"  
00677 "3C2F30CC656934C0175E824E76947F83D791CE7A2960FEF547CD3BA94CA064659D1F68C0DA"  
00678 "8C13AFACC07F3D5A653F259DD141E2ECE9B6D25F67CD38AF3F802CCE332049DCDAE1504061"  
00679 "C2437C7D3BF4E15D88185D4114E96BFF7EC673A7AF8AFD53C979C89FEFACDDF873686D892D"  
00680 "EF5FB67877734CD981C16684320AF392A464C7216FC5C8B22BC29C8430151FACC94AA1513A"  
00681 "08E1FF2F6A965F68E368245510B975595EFDAE80B7FAE432FCE9737962974905F367FF637"  
00682 "121237B2A404C42A5A9318DC5ED9CFBBC2EA9C17A6CA37BB98F8771CFA7EF58B1EEE40BA2"  
00683 "D6C2031EBDD4AF0590AAE8FD10B2BC02A254C97726521E4D4D9FFB74FA5D06F8AA0BD436"  
00684 "31FAAFFCBF01FCA87C9360C5E6A62A4B025B894E2C30F9003D29B642335DB0A3F2A3436A91"  
00685 "7B61776F96292A06A6B4DACE4F6F0EE0232E1C206427A0386075BF50D6074006D45E3A3ACB"  
00686 "085431032AC20D2690F0435D257D3E2C93FAE49D75FF0F320CB4D348C6288", // PRN 45  
00687 "8A7F8C99EA79E1B2118E3B203F72FD8921BAA5C336B006ED66AC7181575D66ADE21D00FC8C"  
00688 "7DCCA7DC9E430D1D086F2922487B5B025AD38750B4F60993368D829CC361B600ABA990E570"  
00689 "370AFFF9BF171610F2A0B0B93A0A3AC54244073A0E816DD691BF1B0041BDC165125A14C621"  
00690 "E01C8F069C3E05F48FB77E66A998C27A87FEAF07B5912B303A98AE5796F1B5D4E2EF52F38E"  
00691 "68F0EC5786C19ED93612D7152BCF0CA1A3044898F9347FCA8370CE6E4A0510750DF6A42C1C"  
00692 "35FD8B91BD246A60FB229CC35FC45525D12A092505D901A4F9E1E8D42D25C41B8FDBA13AF"  
00693 "ED8BC566D4F3BA13A779D7BAAC1E25B6710D7718B73A641F23E1D22CBDED3B7BC3AF7D92B"  
00694 "2DA1CF874D908CD8590C80F1D5055CB8228AD964DE099A4D037202C65D197A35D8A268D8A3"  
00695 "BFFFA839B93615DA295A09AB979AF925D895CB60B5DC5580055BC4347F0DDCB1090DAEF46C"  
00696 "8ADD1009A5126362B0B4F1FFDDAB4A00AD8290EB84F76AA345DD73FFF7733035026E3CC1D9"  
00697 "9428CA68ADEB8CBE98E4A630F99F4F33E8AB66895AE7435D2E84EAC95CF19E9B440373EC0B"  
00698 "4D4B2CF8672C7F1E1F8CE5F34F55B016F8B115FAD6CC7B53DA7555E67672FBBF6BDE324AA0"  
00699 "9FA18F13D9FF1041A12B04F30304774B07659F397554E66CA589D9D9F1F262CF659F718CA7"  
00700 "F443B8331BDEAC83980045562D909EA44E917FF5D47812A0390139B2A87D0", // PRN 46  
00701 "80B35D641CF6EEF705D1DAACE1EA5EE92057FD497B0D937C7CE9A546BB32DE580F8E434D6"  
00702 "44F191798A51875BFB9471AA5D03700CB0B7B2635A6A14750F03DA4FCCB1B363C25A529"  
00703 "4DD8E7943E4CDDA318C4B92B57B14EB0F8EB250686CE129BBB2B18EBE7FF53C9DA9C0B664C"  
00704 "527A3C69D905CC6FAAE8BBA7D83294C1656DA4B8308968EE49DDDC2D0C71A17B02053027D"  
00705 "7DB8F4E77E3AD1C80FFDBD37938876B671D80E99F5F1C7BAEF50B7E05CC0CAB8979A3A2A85"  
00706 "2A7018673292CDBCECAABC74B839FD3C084682357A5414E431C1F25E34850FBC779285854F"  
00707 "C6AFD2CFB7B7749E0DFFF5F93A8AC146C873B407F2CD6CCB461312AE35DEE8D6D51347B0824"  
00708 "156DD60762807A5C132C0667FBBCA7489058C47A156B2A50CA5C24B894C1EE7C44B871791"  
00709 "76905B7657A8E95AF7F2EA6C6D2A12384CAC9E6D14253DC5C31BB8FDD2462581C109D2DAF7"  
00710 "2238E4B6F3436DFD7DD5571548D2206BCB8B837D8CE8C9C3B3066E46A1655E3D84AF42DE10"  
00711 "51DDDEF438821E0F0C1EDBFE148356D707036B269C19C4CEFF4C4BB4048364E2A3886E3B842"  
00712 "EFA15F22C8F92D082EAF3FFAF9BF45247DB76C03E99F662884DC2A29EE488023BF0EBD465"  
00713 "39DAC307410846B25280349106CBAADF658A066C3664E35C4C696726140275AF7611D2369"  
00714 "ADB8258EBD2279DD24DBFE002377B9A3C1B120890A6FDF20597417F88F15C", // PRN 47  
00715 "ED6621A6BAA64578BD988D08FA8D3D2873C87C38AD7EB38C0922FF3F7E59F0A8DE0074469"  
00716 "8939D42D459E6105ADD3C77E9FC4911F572B3856C9AF6C15ACE704F026C6B2FC8BDCD9D840"  
00717 "22DC10C47E11E934BEB7E1DCE8B6A2BAECD384D810BC3AE587E7EDE57CDEA908DDDB0208856"  
00718 "24B0E042DE0ACCE1511CE38AFB6C9DD812FEC339B8137D88108F07035A3869CDCF8C9402CF"  
00719 "96B9E331297B644DB13DFA88F60605E067B9F35607D2D75573E0913F8080EF603AC4B7133D"  
00720 "836884F32F48534237CC559ACABA53A96885D297CBB572BAB1C3275F7CC7A3AAD10F29E727"  
00721 "B28BC29B038F4003F8C93FEDC88C63D72609A5330F36F4986CD9536812678A89305C6E58D"  
00722 "4C49C088B5E2732AF77A8D31D1B8574B9DF02CAD4930D7A7F76C067E7160ECD1A845FC6A3"  
00723 "B508708C279A2C94F2910F8CE34A63E5D554FB530522394BCA00508DDAE6039ED9F476097"  
00724 "81CC60F6E211BDF29E6F44ECC873DD24F1D37E49D7BB7D6279FA79BD08B5FA8F9364EF6D4C"  
00725 "8D129A5A97A8BBF17DBD5E64C4A31426881687BB79285B09290DE61AE40C295CB1BF3C8AF7"  
00726 "56E88068A771ACCE3C4B6E78C62AC1E8EAEBEED0AF3153983214D7459AA8E254633B5E25C0"  
00727 "ABF4647B906GAC50A62543710EE92B335A7162B540AC70F2B2F49FEA892BD72DBFC5F7A35C3"  
00728 "AE56636AF2887BE680FA63768C27040E888202F700007DE648482F5307C56", // PRN 48  
00729 "914BDB196CD56B3B7D7D3F1D7A5E4B0A1389578F11449DC2DF643E6E29F688227C3C07033"  
00730 "C2A3818342B229F63C229FAC11EE1AB6F0FCE8608E03B46DC983318DF15FD8DBF2970EB342"  
00731 "BE2E534BB0455BE58290A48FC60973553E94C4CB53566CE0250D9FCF055936523A8ABFC928"



```
00732 "7DB9DDEC54710859DF62829D2B6A100358EB64E6219451868D6BBC2AE4DCEA0C0E338B26B7"
00733 "48D4A1A34AC16233046CB7D346D0D79A3CCDD4CDBC435B9B3075AEBEDB4C0F18C5DC006F5C"
00734 "208D882308510C75E729D08C779CA99D5A685E78D5628094AD137BAA635B7FC0F492C48A9C"
00735 "DBE63209C8231455012EB3E830B5B2A79ACD8FEA8016243EBC85BF5D6F46A48FE013D2B3B7"
00736 "89BC5CF743200BCDE03995BB2B6A640CFB099788E380B4E01D75409A8D8B3887DF2B1CD3496"
00737 "0091653EEA6C52EDD745B9363BFF666891D9C8BF511C3C07D38F49DA2892DCCCE81E1722F"
00738 "6EACB3214E3335C93E6141AB94E5EC31BABF8108F6BEB3E60B1BFE37579B4D5DC8B77A347"
00739 "940CC1F6BFB5B46097B1EEEC4C354159BB3475E05FAB6BDE5672014D9489CB70DDF537F720"
00740 "9BB9EBF1FC6B8B94564AAD5ADD83CE6E51EFCF73DC6080D738C4FF1CBC87ED420A0B92FA"
00741 "459AD7BE58789F0A191D149F88173184A22874DF6D39DC1BCD4413648B178ECB03F8358547"
00742 "A68DE7B672BE9BA1FFC8BA392F8A58ED2806155C00F86B7669BEE4220D420", // PRN 49
00743 "97051FC67ACA30E8AEE73D3A8CF38BB13524D4E0EBD9BE68398C7C16227CABB1D0B0A0ABE7"
00744 "B6384ABA02905BA0C3C7363599D059C7B4C99DB165CD14FA12FA7912449CA7DD5E346D8010"
00745 "C85A757382270DAD15BA3CE36A76EF55F81A1E80BF366B37FE3A88EC722028C25E2346E240"
00746 "40450A99CD808F942568AA7133981D72E7F2928894670AD5399482DF1B90E7E64062F830B7"
00747 "36C79C30F36281495C76699CD48404673FA334F042F9E0E67DD7F3853BF71ABEAF6A9A5546"
00748 "855E840CE42B224D8F6490C6CE5FC02EBAF4FFC390107058F54CD635D4A7F2878099C1EF49"
00749 "5750E6921BB2F39AD808C4210F287319F811A254CEF8CF153FC50AB2F3D694A530949E5F7"
00750 "8D075DB96DDCF2BB90ED3DE09D9CA8E08662FD8982741DE1CE0A6B64C3D3D5004B5C04B2B0"
00751 "DFD976A20FACC94D1762D41EE03B40D2CF367612812EF4CC41D1BFE9CEB51AE3A22AF1BE7B"
00752 "85A0573048D0E73FA0FDAF1119EFD76F0A41BE63128B22D64A5553E9549D411483BBCA148"
00753 "3EF30CF6A6D317AD2C7973EFA6D4C1121F703D2F48FCD43177AD450D75D2A28D2C244AEA13"
00754 "F0E60AED8ACBAB444D400DF5E280DB799B2D9A984DF1E2567D39D1DE58EF78CA6B4D8BC17"
00755 "2B07DCB02D156CA96EEFAC69E556CFCE0AAB617C7FBB8C34871C1D35E74B7BD307D3F2E424"
00756 "C7A9AD676A1A69E0FE735EA50887A1DFAE6CA2FE4460FC7EF323ADE493020" // PRN 50
00757 };
00758
00759 constexpr size_t GALILEO_E1_C_PRIMARY_CODE_STR_LENGTH = 1023;
00760 constexpr char GALILEO_E1_C_PRIMARY_CODE[GALILEO_E1_C_NUMBER_OF_CODES][1024] = {
00761 "B39340CA1C18717D81EF4FAE4E95BF3504A7709089FB48560E9E3EF802180E85EB2194E05902"
00762 "C6C4C52021FEB7EC64FD416BCBEC8E39D64A4B5EE345291911AB8204A888C25B1CD3D9342A"
00763 "56C538636D3EAB957037D09E879AE5F3A39834FBB84A3D8D5090D7814246B62E9CA68533D2"
00764 "EC403B4FB9488467FF9758B0D15A8CEF89187A1D5897880040B6C3C5244E85A2AD14BCF2F5"
00765 "ABC44A7B1D4A87E8BDA05766218773ED4F70F8D1D07CBB1E8CA6298E64EE6DC5886D37495B"
00766 "A2EDB3E0B0B68AD9F300310B88898DDDEEDF484538C31A9BCAA76ECAD0C16607D32189058B0"
00767 "862EE9D70CEA9D304755CE8037BA4C46C2573181748A212E4B2BDD04F9BC240518273DC17C"
00768 "BAFF21A03E9120FA7DCA18D56DD1D9A7E510C90CF21910438F5531F2EFAFD185ECB6B911F9"
00769 "B7809D98D86F15516FFDDBE9BD1CF8662EB777C3F94EA3F962D7B79449FAAD39935429E92C"
00770 "AE5637E9BCF4E9AD413D27934952409AB53BE4055AFBC4330CD1E4B5509FE5F8EFC9ECBE"
00771 "9EF377DE7E37C479B9D3EE7745E4609B0A6D2C5D92EB3C9E2278C1F2221FF907596AAE509"
00772 "6ACF8990EBA907E43AD320F8019CB6355A2BA8670EE5A4F463E8E56F8F1D3E7F4922510FB6"
00773 "68E32C4CF23AD8496399638B095B47833E0CBB34977EB3E4242EAF870D86660D6A73F83E45"
00774 "D6E8A41EDCA3815079649544597C5C43B6C93FEBAD5700D22EDAF431FD340", // PRN 01
00775 "A64F94BB47BD4033C76D4924305907EC1F618B43C7535F3CFC093E5AF5DD5C4339F3BB6D8"
00776 "35B5C2C2053CD3D5693368D4E1A7CAC59425D1FD96809C67285CFD3FC05B01053CB0773221"
00777 "D7205778022F487BF99D1650566BE287FD7AE882AA8EF52E5D4E3C0C2F971C9FF70AA3786"
00778 "91EBD8ADE45CF213822D09FD05243F9726F6C69893845E57C37A7643E16B770E26F431FF69"
00779 "D437271905D70EB85D8D229D7D87662121F0BEEB1E895ED9589A9CF5833408A04197AC902"
00780 "5D8570AD9B75DB7E192EA0A089504996E9DC652975D83633619CFF80667D8B519536B34752"
00781 "48BA8213C8A4C66DE69B4B3774BF9142425C57F34A27B1E288119E3FFCC6AF6A21087F9394"
00782 "F09DDFBD42F32D059B8CD410A519BA640765D5CDE490E62F10E695FBFD33CBC9D2208A532"
00783 "C8EC25DA288CC1B6850AB43D9B5C00B6E74B7A148791AB07B328D347058C7E6233E18C5ED"
00784 "172C9F9E9ACF29D913E2A1614BFC0893D4967ED033B2B9AE6B51F908F1CED57C14FEEA85CD"
00785 "4D9711216BE779FA6721B7DCCA033C80127AC6E5FCF58EB4005EC24CB4886D787355362D5"
00786 "E7031B9B2AC2A86D730AD734181E723A811FF510A4DF868001973FE83288D78E6F9B9441DA"
00787 "F5BE2974A2848FD917C3BCD346A431922246EC852E4AAD467E60C15D61DD3BF4A207BB57DB"
00788 "45DCADFEF3210BE74B9DACC918A394469F2E2C95AD1E211947948F224F5E4", // PRN 02
00789 "FD1F6976002C39C87187C44E3D224ED4DF0B67750105944C651A5E57798F168A136AC0FB59"
00790 "79C4E847A82B0A2E6C45DB42EF2B930A80D3257BCC53EDA966F5DCD9AD47CFB226EED9B6"
00791 "2A8749F6404D087798A1005F4131171D3A47907A3CD602B83DABE094D2CB031867DF4595"
00792 "F3ED59FD8C4D76DEEE59E422CE5C7D0A5F720BE94FA24DF05F758348EADD5EFE9197C6B2"
00793 "292E2B14DB8C6DB24AA94C5FF0F5106D2B566058D32C58B63A150784F7B02478D9973DD4CF"
00794 "D2E84059AE0F471320754B7EE83F04A51C67EFC2EB1C301C0C58DBAEBE95474E3484A7650"
00795 "0103C14C40BB0B7D3A04D8BDABB605C1EF9FD4A6564934DEC50BD5878243AEE80F9796EED7"
00796 "0CE1B1E8B55725DF76472D12D4A7A487989F42E6705818B1F7E149E97153A7B05A82FA3FBE"
00797 "51763E61171A12931472E94CBA74CC09483DF93623FC60945070FDDF3A00B561650427E"
00798 "4BD64D675B1EB398B35EF057A66FD0B48EDBABBDCD573C2ABAE46F5CDD0CB1FCF177652582"
00799 "36F3DE40B5D0A3C5C978D81DEB07367AB20B2CAA9834B9576161C4F20FB9C184A01DC9021"
00800 "A4E92B71333354E05BBEA9015E5AC4C66312E8B79F0B92279AC7EF1936BCC30802B83DB3D1"
00801 "13BEF64452CAD7ACF6674FDA44023A661019841A101BE80FDA4E3210AE774E433A9ABD97F2"
00802 "755259AEC221F7A8C3B1A3D471F874D2EEC85B9B21BC0C2E2EC9016F847C6", // PRN 03
00803 "EE38BAF6F61704B01509B5210A0534E4702F93190C392E749869B5572BB7AC4D7120E2BECD"
00804 "6618CD376C4C1B4965F7D9D73400824E88A5C7B5B66BA88C3E0065F9628A9AC6B91A188219"
00805 "2FC553E3140349934D20698C9F291B5370948AF6CC90C837B9F3607F13CAFDA92CE172337"
00806 "6E6A5B813A56301B88A8799519CB7646F33F91C44CDBE7F768D7DD9B323A5002D2F784C410"
00807 "1AF90D6E4C5ADE7D085C79E827D43E10DF63AC70BCDF13DCE0471B487C5ECB752B9C3E20F7"
00808 "5DB243790C91355ADF7199081BFEA03D80E82445EC2831FB5014B85EFC2A52748A8ABFAC"
00809 "1BA3904E178DFBAB26C1750228C9A031104F58BB3B91905EDB9EADF7B0F6DF22ACEB0DE944"
00810 "E277809D77507D18AEDA1767697398421115D04AB2EBFC466E99F0AA540482A49C6AC8FF"
00811 "95E3F962734B03EF39873A93B70470B46FFDFDC15C89F8FE2F4637B59F9BF9C5752D9F8AE"
00812 "7EA75D1EAF1C22CA27E5D5C9499624105D61BE2A691F9194D277414532A5E6C63875F7F20D"
00813 "D13C6E773B0C3568392B14A5042843926472ABA343D2C427792199B543BE1D43A178FAA7EC"
00814 "F53B98AB7528DE1B8B88C52D973CA0427636505837F94284E8D6B4F496FC5A48B7958D46C"
00815 "1DA00651B8A7BC56EC859C071E4396A05F33588B087EFE9635E565E6B5A8A70DA70F50ECA"
00816 "D1A85E6E36FF07B4FB3B9119EDE0B611CFA91D9D4C58C1F4815B07B9EB1DE", // PRN 04
00817 "CD37D0FB0043D0344A939E93676B9DAF5F2D19A2615E3D97D624E62ACAC8098099FDB9A5A"
00818 "2F4B3ACF20F75B6807A5A3F157C20CF479158F4A10FB4972855F3AE2FDCBDEEC00A4D470AA"
```

00819 "DF5F5E571818AD6E872D897E2DDC402006965ADF16582B1E06B1861BF7D0C7E7BA491C79E8"  
00820 "6224AF6B246317F725FA74DD8376D63D7993FE2F2BBB2F1DA9238C6F3FFCAEC50FF61E645"  
00821 "FADBB6E03F883892C42CCCF904708B123C9271A670D4DCFC602951D12F5213937CA2C05AD"  
00822 "DE9EE3908E99AAE8DA31951C36D36D671CD7BF15DF60B707F00BF6EBBE5476926D015628A8"  
00823 "5758BFF35C4AC540F39E761B2ED3CA911E8680E28BC387058E0F69345CC6AB3AD160E9F2B"  
00824 "C4D6047A1934E15D3D7A242A296333C09296981BBF3B8577E4B8ED2A362486611F6638F89"  
00825 "55431195B60C5C089F9897DDF0D34A3DC627CE337AC8128C28B63A394908E4C083BCC4522D"  
00826 "B8CE5720C45EF76B2716225E53405FCAAAA72AC198226575D5225195F106C1249E4B87AC05"  
00827 "287A3ABE6C51A2A41E07F56ECD46E989A8568D35669B525A6FFCA90DC91D3013967F6A5F4"  
00828 "C022FFCC17751B68FB0D8F16FC9229851DFDCC060838F923BD44C1AD70A993E8EBAC1667DA"  
00829 "80F91B66F8F5B375D35275188E3C7702C2312CEAC5B20D67BB34400401BDF1DBFE79DFA0EB"  
00830 "73F173A04807215DA5CE8E1D28F2126424C3DB44ADCD7A961260FDBCAB31E", // PRN 05  
00831 "CAA02DD19DB9C721EB35AB7D64B8A387796427242698A47D832C3F1AD4DDA0B5926FFCE931"  
00832 "9EEEDA1565ECB0FA1EEDB424414120AAE8CFD0BE88D4D248899A0BCE31F9BEE7A4DC4DB3C3"  
00833 "B10444FAD6ADCC2E8F0EDF7B808536ACF5EB05AADAE92693EE02C9512B3EEF000844BA35E2"  
00834 "4620A2E8935354B8432C07C8FD615534BCFD0D8E3B572BF2CF06AD343997590FE8B244A32B"  
00835 "BE69125B5D7C5E513A493724EEA8DA6CB0FFFF3ACF1C5085A8120694CBC40FAE1A6326FD714"  
00836 "87CC3BE710A3415CDDFFA8C618B68EA93D330945586B080381F0076351B888087F56B969E"  
00837 "6D6A311AE03CC79FF6861E715C9DA9AE751F1220661581C75DCEC0515A1C9259B9CF8E944"  
00838 "CEC4B1754E5809E985D6F43FE45710893242ADE0D3B84F1E1942B7A95648611595FED13F54"  
00839 "6CA11D8E5A55A3C3C78C3793C6689E1B3AFB5F67526A480DF923A586A779F94A09CF96359"  
00840 "4FF4B0A387876EBB3E8FAB888C97F6773E7F0317B038E47DD7D109545BB07263B1AA84284B"  
00841 "86E47FFB9784A171D101E7B0A6D38BCAE7E63D827C999BF551728FFC642EE690B01D486CB6"  
00842 "EBEBEB9D5C888112589EA5CBC9BDF49E675965223416D6DA02D233BFD4614706BF13373973"  
00843 "207C849A0DE41EBA137FDF79A1EB25D74E30CF60B577C2787DF04740BA8CADE3F9DA55D3F0"  
00844 "084F02809E37543239E0A71E99751EEB21CB3B41488244193A4868CBA9276", // PRN 06  
00845 "FB227530F82BD527E648619E532D7646A5ABBD15DB91A6E7033DFECC65D095A3D83AB77ED"  
00846 "D2F3FEC52659CB3AD1BEB009D7A1C9BFB544291EC1C67B75DD6DAB06E70C32C714983139DE"  
00847 "4A41EE07B4F3C03BF566558484F19A3BB674B6795F0D8537BC31BC8D7A38B2FF1B2EC8B785"  
00848 "39B2A55D10E385DE484B05A411477681A3AE7527AC98BC2943AF1CF7F09ACF2DDE4530AE8B"  
00849 "BDE126FE916E833A1C0CAA2B2D2F5985AD47B2D0D1D3AFB6E50D4B3DA7DEEC4385E6CA8FE"  
00850 "22760F92807AC55556AAF7973E8016ADFD43A3919088B768351B1057498D2D668D7C1E8C63"  
00851 "438055DFD7D36C5E7DF02FCAFCBD9291A2149E7B429B3202D329E47CED51EA5771772E308C"  
00852 "5BEBAB7B934597540D83DBEC6C3BC61A96EA4CB2D7530D9D760AA9403338CD95B829F17547C"  
00853 "5A90D161F7B8CE0037EBF403C91C0D0C70C589BA87CAE8DF26CF14281E235A686CCD10E2D5"  
00854 "20A76265C4C2780EDFD0705E89EFE3C953FE760DE45A8CF1F2D3F36DE3164D5BC2CF322042"  
00855 "28ADD7C182EC55F1158AFA9358BE179C722ADAF1D0BF1306A0B56218857FC5C21001499F61"  
00856 "E273442281E585B3E6DCE148AA97B6622B23BDAECF983BF186F1B34962764758AC3C20C840"  
00857 "3606149A4C14567FFADC61CEB8DC2821B1CF048477E2433E9DC718DE618EDEF302CDCB5D"  
00858 "33750ED0B200CB28A113289DE143D1D5839D2AF8B0525E0027F34FF32106A", // PRN 07  
00859 "9E5DA18A19514CCC849E9697AE4BD1B317BB34927D0461A96A7AF4A5D6C13107FFB9DE38C5"  
00860 "E8CB7C5682827F57D94ED2E77D36F9F1CB05E4C2C62B1DE254C7B1CB236FC4ED70BF8DD1F4"  
00861 "3AC773C16A37392B895F8B157578C477C85E53FA7CA58BE70D9187AF5F7A18D5A1E5642335"  
00862 "E46C2F8F4691AE6E6A9692E21B9668E2C083D9F45C2DB3E991588BA87A0A23808732EE39E8"  
00863 "B3C876BE79227C782F07EE3FB3086AF913D71D71910A0F56D62B5DE5E224F7856A42A4A1B2"  
00864 "AFE380827B8E6B38F1CE486FD08A91B22BD91D09615F417E178C5593E41B0917E075133960"  
00865 "AD28B4DD4096D1E84BEF1363098DDE92C29CD508C40BA7E785F46C1E0DC72E729D394911DA"  
00866 "919EA6F94D14567FFADC61CEB8DC2821B1CF048477E2433E9DC718DE618EDEF302CDCB5D"  
00867 "E472656D6687DC41EA34C2BB4DF1CA08DCB933BE3EF4B419158BA0B68AE82A64ADD5855921"  
00868 "4FD88A4CB34D99F646310697DA982C2FD4EE069DC1CB102125C34A89AB20F17B6EF648A834"  
00869 "732730410FF6881C7919AE4E71CBAE5F8200E523934D84BFA897C44B89B9BC6BC0129F7F97"  
00870 "EE0EC049BA1AF67D00CD624A75FF5A30514399BE4801CED057B498B9DBBF0EB9944295D5B"  
00871 "6AE968C4B8BD2B9A9E17A3039C5FA35A0D30AA54CA426C58353943DDDD3FD185895C0DAEE"  
00872 "950455FC131F520B64AE118C7406D0A72BE6127C5307730AD441B6FC3D1E0", // PRN 08  
00873 "8589F8396F5B1C54CAF2B17D4C152CEF347E66EC7903C878F2823D4ADB9E7CCFAFE926B7"  
00874 "EEB4AE1BECA339A027CE8EF997957532FA871F356E032ECE0BCE3399F81179BF78C5C7D13"  
00875 "5018AB3C40C0BE58D3063DD7CDA4C1918A0187BACF830C8B6900D43B62E04DF6E831CFEFA1"  
00876 "3BD85E873A527F24327C95DB4BBD65C81A20F959F828F5DAE4DC13E5CAC7417EE089401FB"  
00877 "497ABE10144E28EA383E61D4A9B63B618AA7CEA4588B2911EC581F506062B05E7BEF723A5A"  
00878 "465C9FBE70E31375BDE3102845A79A206BF7D996F49A21752D534B73EE83B48C1A22585F"  
00879 "5103DD9B9B68830F61AAF26E5CA643EB62EAF58AFEE0D3494E4F7A4F642A3454F4F56A406A"  
00880 "264148FF5DAC9DF5F151C12E89ED9D4FDDC04EC5F0022DF8CBAF3CBC67CED2853FB4F8C589"  
00881 "4C96CD00550950E7AE2A26C80A72DF533270A0E23EDBAA4D0BE935D62CC885E1CCE653D66C"  
00882 "51E49C43952042E1B2D043BDA1CFFC1E98A3F806EB587A4EC9AE299BD838C68B9BBF7C420C"  
00883 "12B23AA2793CEA0248C932A91BCDD641DCB38F0BD2D7187D8986928DF4602B381BA13B263291"  
00884 "134628FC91C8DE92594B39650B877D9A91DAAA05295457DFB2C5D8207BBCDFE16AC5B9360"  
00885 "0E33BC970B38E18808B1A732889320352B524B109560136E605D32784CA01F8B11D077C81E"  
00886 "AD6B7A5741C82D76CEE764FD07E361D531B75106AF1572AD1375B2BBAB68", // PRN 09  
00887 "A3E17A4CAD2ABE76E32D18501899F8D60D293BB1AC3ADB64F81148AF56741790F87F8B7A2D"  
00888 "9A6E7645EA50B75514C394508884CBF9E320B24D41D8246EB3C163B9101240776C312DB633"  
00889 "33889E3C1218435850471C454486DF7FF4D2DC0AAA14980F394CC8EB7B828A60C53A2FEC33"  
00890 "15BEAEB30045B3E65006C6EBB23B47A8A069EAD45E32E771B9C467B4359EBB681AB48C891A"  
00891 "BB796544169178203BCC4BC6B4A278DCEFA5E9385C059346A23DCCA001FC9E47CFEED4BC"  
00892 "BDD947B12A3F7E5FF8B9372D9497EE1A508D8BD3392BF3CFAD58F0191B18F6A300FF9CB8D9"  
00893 "14FDF37B48BF24C2C5CA76ABDFCCF833D51D48FC90E06E7B972944BCBAD169232A8429B610"  
00894 "0BA562F7F3C55A625A1870A7C7B7BC9BD4C4783278CD95D07F89E8010E78876547F9AEC443"  
00895 "22B0029A922B292634ECCF2BB47BF87909C494049550F1E6D03BB5354DEA7E777F499D2D"  
00896 "6239BFA5C1CFA536F8CB16F4DB9EAD96F83A4AD34AE2C6893ECD6994C89E7F4FE426D95A18"  
00897 "F93B88CB357996B8E5A34C4353EDB1F28A8162FCBEF03704FCCCD80C32874F345D34E81EE"  
00898 "813DF5CC9B9C299362F8443AABE91BD0EAB9746E431804B6129DF32916303A570323FA121F"  
00899 "7AEB2829F2A50A82CACCF6D273FFBD7AC6FFC5807771D216F50742F7091946F91460115989"  
00900 "C87E8BBBC840284BCB95C102CAB53843D581FA9F160CECCE8944E5FC4BF4C", // PRN 10  
00901 "9D7B1CF0029261D65AE1F021DAFA81CF1673C9E0B47FF2C37D1B1AF46E7A91BC5E529C8F93"  
00902 "EE3BC74E92B2743AAB1EDE16A6523B5B8A591C617C1FD0150E63F3B7EF0494162437B0FD55"  
00903 "5A83A38DB519B3BB209EF7924D6BCDE5992BA6248690442E72CD5EB64B4C3D3F7DA339108A"  
00904 "18B61AD88E87B7C85A3A352D7B882FD638B2637A17A2D9CB0B7F41456DCFA66D62913F1"  
00905 "45600BAAEEE7EFA5071C3C9E6FDD0A6779A737071FA6965978CBC89776386B108DD7216FCE"

00906 "962FA87A26B29FE0E732309C0124B0C1E99E5642E5EAE670005B078C097D16C58B8923633C"  
00907 "18FD0B08FF8C4610B789387ACB5A2DD0B6AE7E0DF43A6A9E8C3B89C7E5D628D59759C58D07"  
00908 "E0687812AEDAEDBC63B4FEE8524D10E4B467696957E6791C1E94B13CADCC0ED60752C2DB1"  
00909 "B65E035EA72F89FC679138D3609FD2A30E4DD1A946418253C67AA69B07EBB95D4973F562CE"  
00910 "3773430007A6D877271D5F2B342CC5E76E115178F9C7B1600554F5C794961BAE81A5E9B621"  
00911 "BA17851008BED9B556E461A553FE9BE00A40891750E4EA4B475216283B530CB8D479DC70B0"  
00912 "26E07889229F6017552AB9E01EDE6703FD1E2D59AF0B71E0F1DC9A42ACC5823324BEFC52CA"  
00913 "D0CD25FE8B10C999152AA3676A30602D3506F78751477033DB7AB1A2EDC21A6FE51273B6B2"  
00914 "890088703CEFE74F9EA89881896E5BE124B1FC9430B92F0C0568F5A068A80", // PRN 11  
00915 "F23088E3EAA0A6BA04D0633AAFE85203E8B1829223FA6B730F6DEE6799B521F2E8323B8793"  
00916 "D0F7F2BB9305B3EF4F5B4F1CB822836E4D92C8E4928A851BCE688329DECA6F7285DCC85195"  
00917 "E5BDA3B503B8AE6F1CD7FBB158444E7DE8BF6A9A3CDA311787755A827BCAD3DA5621908EA"  
00918 "913C0316B9B52BFB07ADAEFF17D3766BB450DD71328A0353B09DC24DE93CF83A2E5F98BA9"  
00919 "D612187B601157D6B140E675228B58C9398618C3BF0D11A226E489366102B9C35A916653F0"  
00920 "DB36711ACBA5F32B327F5789F3EF48A338E467F4BC2C6A1308597171903D2AA299CE78523"  
00921 "C2ABE4B15AA4FC48954187E0097583EB099419047244B4931326E5923B6313DE08423DB008"  
00922 "66374ABBF5C31A00542CB97CDFB8F71046AA2A6DBFD7E1A71C068ED70E8D7C3268EA30EEF"  
00923 "2262BD799186C59FF471F73A4E85F4FA015E164F9C15FE0AA5F4772BF2D62B26D3EAA25CE8"  
00924 "3EAECE5B3577CA83A68168FB64C40A7A155905CBA6E64159E55EBC928D125E55165C639F54"  
00925 "5B0071EE3CF1A3F58B4994BB4BF50C2B24F2E06E4ADC90BC1C0954A257D88444347AAECF13"  
00926 "6C152426033463DCDF984BB673666E38F1A45150B1B7D1C31DE06EB9C2F4097E9D9B4D21EBC"  
00927 "9F3A918000DE2449DCB3F5FDDC3C773A645DF560F7E013E847E2356D33EFF1E215782638F5"  
00928 "8034B09F4739F98915BF0B1DC124681492F58021670D03CBF5E8F962351E", // PRN 12  
00929 "EB07F9EDF03596ADC2A3B7EB6DB1CFC911E9A4C42336A57309F7B6C3389282E557D94BCC71"  
00930 "827D7C5737B1C530D2A087E3F507242F3DA5BD1BBCA4DF8B78BEEC1DBF7EBB2EA1CF1DFA79"  
00931 "E60785BAFA23658490C9A64AC61C45779DFAFC6C55CB5C9FE457BF47E45A3FEF092E178ED4"  
00932 "4302A88A115F8BFB6A49CAA9EE7EF5BC95CFAB26503A9D26033370A4EF3CB8A5D094C63305"  
00933 "7C2E2D0ACB0EAA34B0B56FE5A0747130B1E75AA923F6F94C0D024A7FCD22E7A4ED8B201966"  
00934 "C417AE864420767AB3223BFF56C64D4F8F557DD950F7C50D9A39AB2C742CE686C8F92B3571"  
00935 "1904C600A9D463DCDF984BB673666E38F1A45150B1B7D1C31DE06EB9C2F4097E9D9B4D21EBC"  
00936 "A48A93A2F44E2275ED7E59F3991EFB0CBF2E26AC1F8D9A41AAE4563179254BA37028867E68"  
00937 "C8179454B8B71FAB49DBD1F889104CFB64C8121151364BDB64BAF854B0DA22B8620BD7EE3D"  
00938 "4302A88A115F8BFB6A49CAA9EE7EF5BC95CFAB26503A9D26033370A4EF3CB8A5D094C63305"  
00939 "A833387B4F8371C6FE1987514BB458C571E6CB5DF5FC900631652D3FA4444F8F1F03122043"  
00940 "40FDB2092F709FDC51D2680753131ABC33712B4F1067EA1CC87C40B281E69209EDEC42C22A"  
00941 "88950EC1CE8130DA9291897BF2D8D1D106911743E7A9DA36220FA9A0A2A34EB0B28543217"  
00942 "839374EBE79F40B3B612236C902E4CD05CE2E1C07F3DA10E2AEE8E387494E", // PRN 13  
00943 "E9D537A821DEDE526B441BA4252785779B54DE76F82747F8607B8952DF990F268C039CC792"  
00944 "883B1C7C297D81C6C0CF17DA8BA2C71110B16741728725839D33B5942BC0A5614A3650675"  
00945 "FDA5D70F291544249A42819D6EDE324C64596F93E84CC9B2C9DA3717AA6DFFCD03B75AC965"  
00946 "43020A9F2024620353E1364E4320FD44933799FFF083E73F5D20B83BF77EC2247964ECE442"  
00947 "C3213DE99026F8FBAF0E96302EC60067EA38C5CA0CD989475205FA38869E349FC7F79EB81F8"  
00948 "457CA3D1A875ABD166C96EBAF1F39C88815E2258EA1A14943298DA39EB9B738AAA4E0035F9"  
00949 "567A09D572785594496316D56EB3D39E1F3F243D4F16111E194FC537A635FAEB2FB4401CA"  
00950 "A9EB0091CF3CB28B366CB5446A6D3B10AB86B4B1A0714D107FCCBB50EAE520D56A1161E03"  
00951 "849192F5096346FBE5150B6D04025A564A43A3D22BD4B7E10DD4061CE20FA2ECDD36F66BAA"  
00952 "D7EA96CDBAA0F063B814707718F47278F8570F77F3B15799D0E354CCA50DAA38C31C746B17"  
00953 "482297D9C089DF379454FCCCB8730D89B1462AD95426370AC37DE50B775B952663B97AFBC4"  
00954 "03F6F729B9CC1D21DD89EE78AF09DF8558F7E68B3711A7D9075DD4754174802F52CB9683F"  
00955 "FE746471C7E543FF388D024327D1866CC5CA6775C58A14D70A3ECCD3EFAB52F9AE6CCE1467"  
00956 "66A8419FB546E39BE604F43B15AB88C72741F8C7D0A7FE2F462D360676D6E", // PRN 14  
00957 "D79D916241BBE52B61BE8210A02543F75A47032E9C0CC128524A675E94D8F79A69B6842B0C"  
00958 "5CFF53C1AC98D2085299BDBAE6A7A41C724CA36B6275A80D377DC3A6EB4C8D0B6B88241334A"  
00959 "95300B53FF46163D2889D7C85F1D1397924F126DA76085BEF131A65C7DDF60DDF4086BD3"  
00960 "3B44D25025D689FF41E0C256EA12F4353D9E722EE37907AA8BED0A5A606333A031AC6B9A16"  
00961 "614250916759B72FE6C1828BC6C1966C9EB5D51413A77F41F808BCA2534AC49DB1D32D3787"  
00962 "8D5F5C0BEFC099C56CAF50D8B92E7CE616AA026EA1D81DC7ABC17C4705F9B57A0F99FA749"  
00963 "F30F93DFA982A083EAE6582C8461A11ABA74B11663ED7D66EB4F8DE14F090EB1CA6D8D81CB"  
00964 "6B063A391FD354DCEAF7DB71C277D0E92B4B463873DCBEFFB698BDCA17F80845EFD5F0FF15"  
00965 "0ADD9377972E1A4279B54BDD47BC9D403D9FA6101604B79AC377780A54614997140829423"  
00966 "13CF74AD1147CD10571A31D82871B6B3A055D50C6CDA4BDDF3871F41EFDABEE8ABB995344D"  
00967 "B6366E35C6E06907AD7FC76632F99124A58A32C86360FD6DDBF50324D86694518AC44F1FA"  
00968 "19662C0E0FC0860811B5B976A9A96C2A1449E53A7E4A07923E9F85794F228E441D92903922E"  
00969 "5783F2FA21C677251B6B8DB02AC2E242C0C8652E0C17C9E3858E52DE78DC712B2DD5D2AF9A"  
00970 "42DB2ED2EBB3FB6E0FFF13DB9A1E02C8F84FCE37F7C4D2DDC09F2A2813E8C2", // PRN 15  
00971 "F8E2ADACED88277D482951555C657B3E3C5DB79E5A43500F7A2C8B30C854DBE611FAC1087FA"  
00972 "03D439AC4635D39211E234B82A91248DEE5D4FE67A02D5AE25C67E64C4843E419EBB3C4D8"  
00973 "1FB606B9CA0836F8207CD19D106C0E287EFD8F8DB5C1A3A22886C2765FED26B5189153657B"  
00974 "7C47D5590F11C6340067B800669B05A0849BCD2005DFEE6DF95833C9E94328D72F931D69CF"  
00975 "BB2BDA81AC83DD660B3B17D2BA4023491DED324FC4F22510ECA4A5194B1245F4F3FE334DA9"  
00976 "C1E6BF83A3FB30897BE54C688D2A7C5845F425866F25DD0A9852BA6DAAF8437DD80BCC72B3"  
00977 "E258A906DE079A2D33EC5C5F6927503BA13158305DFFD3F86345524394151AA557D6242060"  
00978 "F276BB6BD25586F632942ACF5E0883CD3F8393688F360323A000B82BD89414E9C807994B02"  
00979 "34D730BC6D7CDOA2BF75D9F51078E83EE98D4ACF20EFF86EE9C38B8D52455D8A694B689F"  
00980 "0D9A632E7A6AC6675E190A12ADD716D2C6322657B878FA97267C1BA4631584356768EBBD1F"  
00981 "13FD2F37EBDCD1DF96FB943942E8A5188666235B455BE2F770C9759A8F070971CBA4978974"  
00982 "4FD2F64DC6D6E003B3F9BEC7617C7EEDF6BACA94D374400499CA6813C90A03DFE2C537261"  
00983 "DA93A1C0F6D8BA93D1EB5FB17255DF28B78737582FD675D056A4C474A71CA8EF0D77BAEBE5"  
00984 "637711EA3FF2B014700448C3D74E3DF264D773360F45CCC3342987169C9A", // PRN 16  
00985 "94741D7F05B0CA50908E6BC14801A28E353551F01769451B1482FAD0043D5C7233124D9AC"  
00986 "3344F0FA2E28FD00E86B38F5E0452F46CA111E92D01B37E966455DF1374883DB8B055C4DF2"  
00987 "5B421B2820F86D0825C096018D2949B4BFCEB7BB2C8A5BFA2C79E27F11A7F9B43A50AF928"  
00988 "D81FA95CEC86A11422B99786072311025672AB04B2593CSAF50100B71D052AE268FBA992B"  
00989 "F7868E58EFCDD07A24D2111774A36115C1C527B5192EA955722EAE849EF83817FE8595C96EA"  
00990 "2D76FEFC6747D89F65A262D94B3F5E89A5DE8B1A7333EFCDFDED17FE1CCADEBAOD1E7B73E6"  
00991 "7491B413A862E3A308D5C211787E6ED8683C6E1DDEB8EE2D281166C03E7A72D7D78B8878"  
00992 "D07D2216C21B855CDA76B7B75DD1B2CB876E59F91F040D42B97050043499DCFFC65AF803E"

00993 "2F7455C9669DD9896FE1F62227936DF905835A644D31130A39479DE75B4DC4361E41202D51"  
00994 "D50E0E4B4B218AF7F5CAF264DCD060C296E777DF1EED6AE8147E9B6CA73184C345FBDD89DE"  
00995 "A4999C42AB4681D9EA3B86DD75031A33DCDC807F8FB14EE0CE61B16068AF01CCE7378C9D96"  
00996 "5943476AD21A469D8B0CAE15BA8FE04971FE1EC61D3AAD3386DF71B33FD0B4F324F3DA518F"  
00997 "0CC0353182B3D76CF4EF5AB150FB9E74C28234CB3D907AC81CB6D3B99D510B481E1F0423D16"  
00998 "F49B7F5517ABBBEEC07F46AECEBA5F15D91AEB0FE91490E91F739D465225C", // PRN 17  
00999 "839A01464B7346A43D1EA24EB363EAAA590F4BD0E4492FEC4E3D4DB5883E4873BBA17595F"  
01000 "F48134893F16F5C4A43659C4648A268C3303B2DC345E8C98FBBA6D06946F997074AE15680"  
01001 "EC9423D6464585D98804B3541662E183F6540503BEC204749D58E3DB9ECF11C80CD3A38F8D"  
01002 "66FFE6CC8A003BDD35F547E5039DE9A21F70A8A07B2DD89B68E43B42C2E021A11909817C54"  
01003 "3F839E6862268E38DCE712B4D49C39A5035F3D6BA19AE028AE70CCF557720794FEF6442999"  
01004 "E740CD6AFE6235F165515FDC24AB6F578DB2549C8065E008577FCF8B8DD8A3BA679BABBC9A"  
01005 "747A4E2DABD91501424E4191097E689A741EB6644A771CABDBFE6B74ED3ED171DF8DE641C1"  
01006 "D42213B9D0F8CAD1E11FF63670F5587F1FB7FF92276AB48F317151E7A591AF4F0966F390988"  
01007 "3EE6015639671BDC3D1378750F66F5DD165912CFF1A54ED463905404EB7D3412EE2B0F0D9E"  
01008 "6B99EC81678ABCD1789BD8F1D72D3DF8754A16DC2106B83B325807E27BCBD22A25DAC32F27"  
01009 "EACAB6A4CB6CB4ACC90D5302BE5E9827B7AB48BB696B2902975C48B3A4BA4630B14E0FDBA0"  
01010 "50B718C1923DC6E987296064148A9BA2D938116266C597D9E1F11A46BE0EF526225BE70F0"  
01011 "4253B9731A017BE2CA1D43024F75E26BBE989C4D514D01538956FE4B90BE17B3407B55BD08"  
01012 "BA50FA807D0E448B7CAC65EB3FF856772A933F0C5F3E6F41E051015C6F9B8", // PRN 18  
01013 "BDA2B72F0BB0265269F198207FB061DA29DE43E30847E7C062A581A7EB53491EA51B51EDD3"  
01014 "F6991D15AF89AB53198537988350FD5FDF8E003019BE115840B9BA55C238C3CBC72C0E24E2"  
01015 "5090A3D6A59BEA9FED0FAC9EAD40451A95649638FE0BB0F8FFE61AF5B9A8AB84BE84C65EA1"  
01016 "E12E9F6650ADB59A824E608E80D1FC3AC19F418169B3879CC946165511D5AA280AE644AF36"  
01017 "0C42F7A3EEDF27E368E46480E3353E67F536E02B33505341BAF3941069567B723D7C125C8F"  
01018 "066F9A2655436AAFCDAAC554FADF0A9AAD91F1263DC62EF91A748FFB29F57E325D65A38E"  
01019 "CB4F2851923DC6E987296064148A9BA2D938116266C597D9E1F11A46BE0EF526225BE70F0"  
01020 "F3E5B0AEB7DC2140FA3A48B7238D0F5A872000782CB6F7751443EC6A1B7FA1ED02B9ABCD1C"  
01021 "1DE4FC85E9B405C7851913C60F85582B1529276AD475AE52BD8115B6E73A53506E7A0244E1"  
01022 "C29BCECF4F20CDF883392B3990BE2A11B3213B68EC4A166C77D724CFAEBDC34C45ED0984"  
01023 "A8994BCE1FF6A9BB80C7F5CA8FD44D3FDF8DEC8BA6552C234EF8DC52382D52D2B01BB23404"  
01024 "FC453725C7C9269A785FE09C712D4ADE7072B66295CA0C6405D9859E134FBBD3737F2956DD"  
01025 "1D718A9F8242CE95BDB1E49F265EBF19976BC46E29F7DE0EE5C89A43AF2E107588A46E1B67"  
01026 "62E6F8E48B8FC4F4FF93EC60938B8E5C3719022C750C4309FC62ADA4E9028", // PRN 19  
01027 "D240216C5C4A70742CAA03AE910E8859C92E5A90A352CB8B45847BAC7793E1F75720D44919"  
01028 "E896AD4581E11F8386FF235C9834BEECAA1556794BE49033E79D4CCDB4DC67C5200E8B6A3"  
01029 "EE891E700B348CBF092E4D3FA5E648B620E34E491D7B628A1FE7E2C45586B6577E50788687"  
01030 "F0858C10F78F371B25C712ED2760C3D605D4ED4F052E8B66FC308D3ADD4A9B86F00CE4257E"  
01031 "ED085EAE95FBB1E113FCB42CE12BB6076178A20903C55DA570EF8A25BA7AC8B7E134B8D4E3"  
01032 "5AB172CA33CC9729A45E7E579B9361B92B49B63BB1982740015DFEC16882989C917F50D5FD"  
01033 "D9166FE1001F3282D3C54A28AC7FD773CCC0634AF7CDF225F94107C169D2F2BB757EEB5593"  
01034 "3CC0E0FF116D7FFBA992F9A075A2439CCB369D5B5DE460CADCF9C81D98E71651AEBFC2A918"  
01035 "C551082D85F75675CDC8CCA1D3E486CFB3B025D27C8D67C451FDFCF59C3BFA163EB791152"  
01036 "390E9488C604B9B8116C329453A98F7A104527BC677411034CC49686108E569B7595E1DDC8"  
01037 "5918D90B8CB301C855860D6E4718C0679DAB6982D23FCB6648E8561F44BCBF9B052D8B583845"  
01038 "23BC592C9B7F824B96AD1A39AEBD2232D6D34DC171E8FBF933900960F207B55597759D23E1"  
01039 "E794507586114228A2FC100CC200D2B862DF3F26E6D1C9370373FE165C326D8C29DF2F0B30"  
01040 "71AFD5215781BFB589F605263FF065B7A5CA3F6AA9DE3FD8BF5589BDE3526", // PRN 20  
01041 "8E7752C52805DD0A723D61F0BBE0122DF576A42B5AFDF9F196A766C9B3BFE296DC16A892FA"  
01042 "ECEEDD8256D2B1AE6BFE5437D4A2691803043B59862B30D68E4FF94A0700D735CFE9672997"  
01043 "24DA9D680200C898ED1C785E7B8CEB14F1DCDC73FC625F9678BA07603587220C2FDFE0A47"  
01044 "E82ADF36C26F942797D608BA6B38A3AD1A967315E1F2D665B27D51E350F075531A179DB2EE"  
01045 "D55547EA61761CD2B3962FCB347279117D1C7A7574B9F9FE0991AF572A2B0C962A8A79800C"  
01046 "FD524AAF9E6401C44569600F41F04422DB891D25B9F714713086BBFD0FB268E66A4FB10C0A"  
01047 "BEEB31D0FBFBFA20B0E4FFF404051596FC6F6C8093AD01807FA52041CD33007B205D15D47AF"  
01048 "733966411A36F4C7B846D0BE049ADC21B89EA4CE0FBA414C005E66F36FACF3C43B474D47DA"  
01049 "D78AC114D0171C031DFBE4A15FE1A22603CD79B6BB448B67A4DEDC97262F7B869C54F385F3"  
01050 "682C744ED5AD6C0B6E16793920E6B45A024010896D5FECFA111CC9F0C34E728B32F2C4D45B"  
01051 "8AA69B621AB9AC3D9D79B38BF205E8D0D19FAC44A76B9F5644526E06858F76B3EE2D74AEB1"  
01052 "971D6B6888377339AC32203164564B102B26C370A9FEC673C285AE0D1D3DF239D48B649"  
01053 "2B89846EBED4618AEC940DC62AF4C3FF0D56FC9FBE23EE3B0A4890BA2665A88E9F40C4B6A7"  
01054 "70F9630234ED10A3A7FF3C5BCCBA836F3EDC8B821AB18D4B1D51D9962C328", // PRN 21  
01055 "E682E9D8E92A7837823C9B7714D267F9CE290E9FA6CC0A8432D3F7507DAF6CF681246A4C2"  
01056 "323C6B53BCC6E53B31F49742EE5F4E6F79DC36727E98B06D0300ED21FOCF5F2B51D8304A51"  
01057 "D0B498F4BFA3C90049B8117DAD334D4B2E3767EC42D0EED63B3726872CCF9A10223A8A4"  
01058 "563BE8AC266E967004921DCCEEA5DD80C62567FDEBF2AFD0F030192831A6FD871F63D5DADA"  
01059 "4B270AA9EC0ACE47E75BD19018CB09B548D4F2C24831C384DD2B807852F596BD4FE32ACB3"  
01060 "A16899D0B100E9F6D06AACB8DA8D51DB0B0F600F3B614461F5238188B5DA68EA753B6ACC"  
01061 "58569E841BAF92CEE04E6E2626B1FBD01B9B67D1311B1C3D67427298E2D193F0647EA17D16"  
01062 "FD7FD6A40A1BDBB320A1F5FC64B97759AF4EA92AAEB759B5DD30A726E9B8EAF372FBD83CB"  
01063 "FF0000CA75F219A95D6A3CDE38B8DFA9281609A20EE39B73FEBDF6A155359476D073E7153B"  
01064 "C918C1191C9BAAF0E0F161384DAD8AFC31A3FC1E9EAF4A95E22D18C05194EB85298AB0F042"  
01065 "E447DD627904B73E6E505712DF010531C88E695F6510C78B443C731D7FDCD62EB7C4015AB5"  
01066 "D530BD09CE5229FA4DC5642AF176C39D60FE070DF635CC5435136C7BB9C4DC83B0D382B9BB"  
01067 "636A6C2B3838542904D53B862585FE6EC8960A9A77783D17B2D90506F5D60998602AE5430E"  
01068 "86025C8864883CECD7CE51B49CC2953A2A41D7EF8027F1A83815BBEF6F6B2", // PRN 22  
01069 "B6BD4204243CBA14DA15A256FBCD138B5D875E28BCC0BA36855E648434CD04F49935C3D07"  
01070 "4DD5BA2EB82AB14E82C30991A1159E990D1D36DAF794853A23C499AB6B3DC02A89F0143103"  
01071 "72813643F786BF19D3FA8C463EE50D9FA87107E91C461AD2E5DF2FC99630D2005894CB7698"  
01072 "123111FAFC0C5BC9D1E8E84FCCA5179A6C9AFE3369222D66854F90D2668A57FDEE00C300A"  
01073 "EA4E88F03F05C4D7695B206DE9F7E1D429E5E6B65DFE05D4C861F4E7844DDB9062C0B6DB46"  
01074 "B27AD0368992F54A44829DD11A05AB97BA8AD854E428B87F20C4E5E4BB1FF3803809A81F2E"  
01075 "4C109572006729A5E490E0AA40BA55F4391C9FB758EFA79B97E6D413BCB02D33A00DA6705B"  
01076 "FBADED66CFC21291C494B7C3293810012ECC61415E609DD97AAFFDEB795DE36026B4602DD5"  
01077 "46A1AD937F1A6DEACD3393F5530C48A7974E2882CB327AE600C05A535BDE5D15AC52485958"  
01078 "2EEED262194B73E01643359E7B2625F3EB9FE7137514ED549A3196FCBC8072B4F6C18CC67"  
01079 "AAFA0ED6029A805EF0987E2F27A3260F849C68F3EF91DAA9E579AA16FDA698CC18AE8706E2"



01080 "8C6D84CB3F593273D763C2969933D8EFA564E8C06C427809E6A5A6F76DE7C8B07FF4EDDF6C"  
01081 "F2B7595066DFB15F5C6F3839DEE642FC86BC1F3AED7ED2E65B665198AA034817DBBBE0FE30"  
01082 "66DE2B161276CDBD969FDA05AFD6D6A570C1E3CF7E32463441983F257E2BA", // PRN 23  
01083 "A9366308475F2D8D0C2D451C4A65A01EE58A0AF19B791D97382EC59A52616C7480B86EB1D0"  
01084 "A83E93224B0DF73DE1D7EE6D51088F3B20B7937E6C0144E0DACA6324F0C8E5F9D93A8CBA10"  
01085 "45E5B509D70F6C0B7D582D69008D9D3ED6C9C1367D9DB7C04621D7CDD8A5A25"  
01086 "99EE45B87A82F8CE8D60293E7A71D11700CA9AF117D630C5D8B876A9DCE519BD653114448C"  
01087 "68B265813C608435B96CD642A420A15FBAB467692931BCA74F1F9D23F5BFDDC5B8651139B5"  
01088 "A73F04FEF3DA64B7BD56E49235069EE5E8A136B921051F1D1C7D5993E6EEEEA2D58583152A"  
01089 "DCD87AA89CF5962BC8341EF99CEB3682A2D0686602CE140ABC2FDF79A778A9D75AFFDBBA00"  
01090 "C0BD6A8A8AFF9B5D1F30C8373572C81BD9594890102F46B5A393ED126C36AEF6A66E231A24"  
01091 "6FDEFCBD3DED198ABC54CF357ABC67AC83680C048932D7C902AB7DB16952B3C95DF4E845B46"  
01092 "A362FFE1A27CD1388483FFA41AA563933371C0180848F9E3C03AFC1F00D6ABA29A953327A4"  
01093 "E3D9FAD4616C8546C9AF89FB4D08D4256923B736A8F68FEA5A097E0640C1E0F7F942E6A6F"  
01094 "5CBA76BB00D81C606C7FED908789A63F01F9B5FC7B7BE434E85A0A44B2070BE71AB2BA0132"  
01095 "D9D7B32E2D2FE229619F85643E75B4141D355386D1A09F45738455BC21607086C7BBCD4B73"  
01096 "F87DD83E905BC8F8C5BF1824E904C4F5C26518B2FEBF8EB06B22437270C", // PRN 24  
01097 "92D7BF3F540445C0E508E80F9CB0C502F0897D717CA232004362F394A023BFBFE3322C1"  
01098 "D331AFC6454FC756FB48768693FD5C46DDB40DCBF14C726C24ED67D8F3EB613BA80B0E39CF"  
01099 "0747DF62D258613640D881E085C377DE1C3D149C8359407C2C6ABC0D2718A2D42439A8E7B3"  
01100 "8CD7DCED72414CDB08CBDFE1DD356E167A9D7B20BF7441562EE816435A78BAE7E5A5EB4FA6"  
01101 "E5F6368B620C1750278DF393F7C5035D47897FC05FBC419A61330135F24365F13D653D77CA"  
01102 "2930DBB05A3815FE83F75BB1BD8B2DE12A2FAADC1ED62329C55B87FB32CC8F3B42D888981"  
01103 "B4192480D1F57CEB0C55897BDA6B9C0ACE1E7E4595E30C7368306243208444FCF4574C47B0"  
01104 "7725B25EC2E28F4C50B744B3860B361DDDD22D949AA94EBA4F97606FCAD91394B6FC0E634B"  
01105 "D15E099E697403B2AE84CDF5DBDF36D91FB82C0BC12B984FEE83CA9E97C194CADF8382CECA"  
01106 "AF49EB3BD446F660F94C188C074CC312E186BEE0F6585535B050C226659A94B4C4974DA32C"  
01107 "DFF30DBEB4DEA588C6F490F7432DA5FA2408BBC931EAF60EADD7B891A61C157147B8DDE7A4"  
01108 "5F909BD20D5B12009783DE410940245FE4E91ACCF72942E486AE773CD665912173EA29875A"  
01109 "1722DF85D8414CDB08CBDFE1DD356E167A9D7B20BF7441562EE816435A78BAE7E5A5EB4FA6"  
01110 "AAAC36F594C93E2851D76B6A18B0B03B30CD38B97E38109C494C557643D58", // PRN 25  
01111 "BAA2716F115D72D2037841EF9138D19833C7C5FF40F058A960826E690315577710EFE64BB3"  
01112 "7691564B3B0B6C577DA603CC3ACDFE1785541AAD23904758A5A13BDB018E7169D479A1FAA0"  
01113 "31CA72FA6D6AE9613D6B2F82AB07500B49DF535F86A76350C140F9CD25295D6BC2F38C5D13"  
01114 "C99540E2363862F06DDCC486D884999BCB840BCCAF2AB884F5906B9AA0F77D6432F65315583"  
01115 "92641C52FEAF9B8EB86BF0158134129F34ECD0768BC02ED442254515A74999C6B8052A1FC7"  
01116 "97F5720738C69DD9B3FFABDDC8515CD279B246EA7C67754920038C5A4C8D301119CEB95FAB"  
01117 "2765DE39DDA84180CEBAAFBF4976118A8373FF6BBFC7FEB3CFEAB1DA69DD3DB9E428C5949"  
01118 "50FD51F4D98A393BAB96001461F2765834ED70C60BC56406FCFCB3E784C59B91C19783E67CE"  
01119 "6C86713C43CDDA9512B2E7173AFC2EF9A172C9CFDD3000D7A981440AD994C39DAE6FC0B64"  
01120 "5BA0FD49ECA19E572ED0FAC748EC837A7D6F28A8D004402F71CA209BB9403B21E29836C5F"  
01121 "E897268DE0736E985F9631DFDD1AC59D5411E684BE082F41108E33D2B92B2D45ED70FA52EA"  
01122 "2D6DE121EB9F9C886DA479464A9DFD9970A406491E334372D7B78936095A7459BFAFF0E909"  
01123 "0C2C6B6D62624A79334F879A5C92C685B50F75F04BA664EC95893FF40D62EEB24DCDD28872"  
01124 "9D0C297DF5AB83C77FC11D0EA3EF18E3BC7C2C065CAC51390C610B591D24", // PRN 26  
01125 "98CDFCBAD056240E180F347C00912F2D9ABEBCF5464D410BE6A50404B830F744D78F7D9718"  
01126 "0404FB3BCCC2288B7991810B2562C4D509200CE1F9C4DF6DCA4C600D9ED49C9C1456141C7B"  
01127 "71513E74D98A1970ACDB6C15B4A4E327B9A87ADA73D1D46EB0A21F2F5481C3B42931C5B780"  
01128 "FA526C29B98E6B9C714B20049F7A05252CBB84B8E36026DB2379C9632A0843436ECB72D15E"  
01129 "A2950ACDE18DBDC6DFB01BF08F7E191EC885F11D1D8B7BC96E9836B395108F6854545082A6"  
01130 "94D5974CC36C8A658349186C1BA892DAA85D3F156BFBE94C73BCD815E7652C38E178AAF020"  
01131 "14F0E6F23A4E7EF689EBF3ABDCCDD40E2DECED316F07E20716927C8F7B203D51D957EE6EAB"  
01132 "062B99ACAD0D28E0AB50B516CD92CBB9BA90333E73D58DE0B4B633D81EC93D15EBCCC813EE6"  
01133 "3D63BD18517F46E85C37469574B8122FB91388123E1D5E805166FB7157494F8559F90A4FA3"  
01134 "DE9E71DA6FA7CCC6086E638BDD4FD3E4487506ACCF84F1E1678D714B86FAAD57A6B76E085C"  
01135 "FAC30DE4693E32E2D03C63B43F073DD24F4A1E039B941E7A97F8BB28B516217455268B6EF"  
01136 "BB0E1745C2D36D12A8CD13E5D242F562F56FE92496342000A731BF3DB0A7D3110705DFD0D8"  
01137 "DEFF885665B77347CEFC8629F3757304F6129DA9845F6509FE3D32DE9FA86EA4FA9BF86FF7C"  
01138 "C8E726C0A9F93F889C467642C5E944501BEF8ED59793AF8804A9951B4B88", // PRN 27  
01139 "906F6C5A1D3BD03A03802EEF5937E214E87B5E2F0182BA2C258F44B516EC66EACB705E06EA"  
01140 "GDFDB5660088463A421DB03A51460091D7FE889E6DAE32EC19190E7211F08D37846CEE7364"  
01141 "B6ECC07C1740CE990141C4DC4CB0AC9F25CAFCA6BC9111102EABA250ADFD505201FFF638B3"  
01142 "1A77CCF7A1ECB273F9C8ED84EC2F403C1191596A53EAD823421EC47DC5E78F3BD1339532C9"  
01143 "7E4EAA024CCC906EBFB870C1467C3D845A178EB07C11BE8D57E4EDEA7ADEF162923E952145"  
01144 "1B871DF6E7DCEEA7F62022106F647DD8A23074AC10AA632C56DC32B34A4A184FACC64E5D"  
01145 "1E8FD69269660543EEA2FD584117A3EBCF6268352F0212ABCE7CD28A93C9AF76722FB5A71F"  
01146 "F9E5AC4579A2BA32B91818CDBC62C77A6A8EB1F4C34132EB463812B329B6B22108AC36E71F"  
01147 "3838AE3A52C632796E45189632B73FDC0BD37A457204757261B7FCF01E06BC767A57A5FA7"  
01148 "CFE43794F65398A94B4EF09D6DC2A8691BD0CB018BBE7B66E0C37BAA4723247AF3424BDE22"  
01149 "614A9A581A7982E8C2323178BD2D46E6912A2FB2D2531819A180689D7F2C9B5C5AFC2DCFC1C"  
01150 "7FAEB1927EB79A72EB1203BB0FF17DAAF27D66022195890BBDDA786CF1C36ABFD96BC36FA1"  
01151 "D2A5A0CC3D7EEE1A1050CA840209903CB9FF429C7EE9DF9CBC2BAB84CF28FAEF5BB45AE958"  
01152 "8970A2B86D9ADF8DF134C1FAB0DE274B5C7452C4836A573A26A0B4C14B74", // PRN 28  
01153 "C6D5046A5000ECDB54C872F2DC494F2DEB88430007C9BE8EC39FFB148F00F7861D8277589A"  
01154 "C839AAD30AF7D7A2E0F9EE8217A39C521311E9BD59A71BC6663A7738669D6D3BB28124A80A"  
01155 "BDF905DFE2C9539CCF0C8FA39EF84E9633D63BE0C32F3B2AA9FCD18AC38C3C00924E9D949"  
01156 "77BDAE61410F997038BE066DA6C945D8258B7DD133EECB836A7A6A2907C431C522619D466"  
01157 "430E6ACF15030F7FBA4F3D6BB545CAD85678E81898D2DE358CFF3951C8184066B18930DDA8"  
01158 "67890871AF6F4133B492FC894DBE4AA5F1E44BD361C4560ABBDC3101B4AA4E065FD6030879"  
01159 "5DDAEBADBB704A3D58776006CDF17389AF49A0EF09586410015C7DE4FEFE8E8A6262B36FA1"  
01160 "B93BEE15CDA2BBA60B6CC72A7DC9C80C81C9A25FE3D149C7A8BB2F704BE11177F92E2CEF0B"  
01161 "BD12C0766D691CCF093D456AFAA411A8FE5F1C1A44F31017760F0D0CC3B271FB15F56D9F51"  
01162 "A594C34FDDF8FADCC91584ED8D7E1B6DAB27B2BE1BDDC4486FB1C822F23704BB2EF4B521E0"  
01163 "2E42FDCABF69588B0B9D92AAA73116D26E8E9E48DE94F6267414AC845467597B4C1F2A9A8E"  
01164 "1E82C0A1C05955022CDF87386098EAF5BF1A040716A89BE53A36B143376927028A561BBC0"  
01165 "7AFAF42494DF5BC0D95170D853DCCBC22FD36B7947712EB369077D02BF85B0A4F57757ED8"  
01166 "0B247E521AC640D1B1CE30F93DEBBE2389D364A8B7971A51AFA4F557A8E12", // PRN 29

01167 "FDC36E6C842D2ABCC9E9D8783D0D7A7EB74992EACEEF6C618AC7DED4E457B1A708BE2C82B2"  
01168 "8A9563F4A088FF7D8146B16B47A900DF49A4F3FA8EDAFA09F408B025D04EB673E1050F05"  
01169 "5959B7951CF0E999CDF68EA9B3233DBFE0516D272111CBB9933CA8AD8AA6025E5F9A062"  
01170 "8305344CAFC3CA391BD8DEBDC58F7FDBC041B349900E397609C71E4EA3A9D8407C63E8A6BB"  
01171 "EEAEFC92E9C939147920E48E35DAC6D123DA46E4F0838FD732E43FE4EEF6BD68D5AF0C9BA3"  
01172 "A0CB2823743B291D4E10054F6E5DC10A847E661F39C4C133289B07ACA8B544EE3E2EC8C10"  
01173 "B18C40CD9A8E48A93378FD50E077EFBC21996424B539A397B3D2A6C7DE58112CF55E82E8FF"  
01174 "10F75571A15DC248E6B77CBB91D8BF2D53E5C4E9A85C7EB8FB690F74BE029CE1B569EFACFC"  
01175 "16872C5008820FC6A7D12AB43E08B4AFF57DE6B43B613DF8480ACA556E29D792C6C81CB1CB"  
01176 "54A67245C571A04965267BA0F9CD3FA0950B9A5B393B4A230A41E455267CD396F42285F0E4"  
01177 "9C5AFA0B53EC7B60C1C317EDA3FAE4B1713A80D4EBAD32FC685C13649C4806D6FD887A24A4"  
01178 "F7AE801405EF28F058B37112A680F9E9AD0456314E9F490393CA25775797E4CCF9184FF0C6"  
01179 "A237AFFA8DE1B84C420A6183B1D49D6F2AC1E673E7FDE161A8159DCEB00D85F032EE76E393"  
01180 "1C459CE935DFA6C6110591EE58496B82A16630E82320B9510880BE4E72", // PRN 30  
01181 "94964FC9F66389FE3880283C4250E6E19F195DFEBD2104FC0959E084308BC9CFC6E5ED1C4"  
01182 "B48B4ECAEB4FDE5F215FBED85A6CD4D1C1466E68A4CF21AEF29F77933549A3A6FF7ACD8AB6"  
01183 "E6C689F1E8DF0AD8AB289D5C3023DF90B21A26320CE8C1CEB2C099FC1DB5873766585DDC"  
01184 "20D587E176483323EF14C80AA4760F751EE5B28460811E5110FEC3D689AE2A6E91D0A31E2"  
01185 "2623E88571F4DAC895AA428D42634EC142E56D0D57CE68D7949BE13AF234229E546E9D66D5"  
01186 "C58E510BF3EAC7B73309BE16DCE6E2280AA80247D9EDED20E06295C9876B412B786CF7E5F"  
01187 "1073792158131BFA002FE7750A17015A9C2580646A9A0D2A3F0243AF1AB4FEFEB3D02850455"  
01188 "3AF9C5C34D1A4A2FE3B8DD8BF8CEADA82AE63C19BD7981D97155AA2F105D724A8C09310D5"  
01189 "C316877062152419A006ABF56AADED74DF0DF325D666C31DF51F194CFEB331E7DAF0041037"  
01190 "2999D2D05B023B2C3067E6CE4A472FED3B8BE1C15C24DFBF4956A5B670FFCF128E5A230397"  
01191 "64BE39CBE55636B83674060B3CCF5EF9A7B7EAB0813ADEE82E271C422FB78A9820007753B1"  
01192 "E62BF4CCCC074F7796D5B2008FE6542DC0C77ECA3810120ABE9F90BE5934E8EAE365D02B3D2"  
01193 "DF4E4A482716483326B113EE5823DD3912FB31E3C4B46B274D7115F34CDA793DB6AD2CD8B"  
01194 "AF4B13B832AB605BE42B2877EE2E66B411668EA29A7DBA5BD969B9F152638", // PRN 31  
01195 "9B8071D96E7D361B2462CA93748DE4D31746972DAE582AD4F70A188CB40C2E6E418288B6A7"  
01196 "13ED46A7013B3EC31C9EA6217DE55D016A1977A0B285224129CD5C9A9E54F3E5094258F11"  
01197 "C0C995F60785614E560764312CD86C6969B3274236EE602EFAE392C015E4C3972D6FA2A47A"  
01198 "B48D5C5F6836AFA54F28CCC03BB4DAA0A1DC0DCA3FD3F2B15FB2ADD907D3BF7719D1D9A828"  
01199 "4A47C30F32712A8CD40148B8DFDB851FFD25ECA2864150B832F8B5DC3A7C701371785A662"  
01200 "85601E96D285FF88947804AA4D88665B3E15760CDE327FBD21393042BAF62FDCE6EC41955E"  
01201 "877ECAC331D594ED40547AFED34D410714CE57FCB4F01C8826519ACB85F447306C86BD1BA8"  
01202 "189E0621DD09451E8F341AE47E7FCF1FD2DE2AF78E0AFA27A4B6DD51A0710FC1FC4A59823"  
01203 "4EDA1A4DCF0786B779F637EE1A72058774C1B4BF5E125DEBB4230645ECF87E3C6FDC91E1D1"  
01204 "43977A72686784815D9654839AE8FA43864709EE0F4A336E3C399CA20B2E652E2AB17719F9"  
01205 "253F77E7B7A9E883BFED4EBBCD0F8CD977583BDCBEBBD925676F56AAB0C36F3DD915F6691A3"  
01206 "0D60D523216FB233CBBEDE7FDFBA827450E595AB51237F9E77058E40F862D3A5A96E4AA3DA"  
01207 "74503812EDEFA501E526DB6B4C642222D7F33B06D9CD0023471DF5730CF8E2BEE834D108A2"  
01208 "5729C1C1484C207CEB0E4598965EAB5216D4E7C30577A89FB8BEC0B118F4", // PRN 32  
01209 "E43093167F77AA20EF9F4AB7D715CD0E86542A0523BDF0256BCCFB1FA987A6565F808C03C"  
01210 "71E6343EF41D2ED4C4FDBBB1AB1D617BBBDB776F07B000922FCBB3718F5FF326A7BBF4BDF1"  
01211 "48206411E86506FE5CEB8EC0F7AAD1DAE3BB46482F5E85D3B46DDF93F4BA19DF9DFD700C35"  
01212 "DC038CF4E499A058995AF06BEE8535631966751A85571271E6877BE6216AFF16AA2B9046F1"  
01213 "AB9126E1DCADBF40C4B390D23379D52C80C10CA05079D1F05CA294EAF7E2C66B884DDBB52D"  
01214 "C85A45FFAB82BA6EB95869689486DDDE372DCDDFCB408496E647F33575BB99FF1699219D00"  
01215 "D2608AEA1A47CF52DD21A3CEB3F043AAEB40EB27F042D276227C09CC8201304C55D254AC22"  
01216 "8FF309FE8DB74C85430FCF9441EF1B51A1192703B140CE3703E5848477FB0BBC2E82DC3870"  
01217 "2D4C6A78A41729B7EC86F9890EAD369DE4F25EAD61DB0B5B0ED764EC84FDAB904FAD9BDEDE"  
01218 "BBE89F338DF52F7A6FAC8EA1D76001846802B328E8D1F110219BD6BB51214814026BFD6E32"  
01219 "627B696C7990F53591CC811A9A09A1154A50585B882DE0B404D7B5846DBDC8C66E4AA0C7B9"  
01220 "0B08CC96A5CF2352A6592BCA536359C863D639CDA60FD23326619A9014FD878EBFF3E73E44"  
01221 "24B045CBD506ED744D3243725EF9491D6165D332FB12985022A71C7DD4E3F942AF5CDF2396"  
01222 "F6C346088AE89535E1B21DA8F7B2E1E12DC8A37738310248808B9973041E", // PRN 33  
01223 "F5B9B9241E8466AF61609CC856B6B2B33DF526EF1E4BE4DCAA1E1055F033072B3801B642"  
01224 "700249C024414B3217D160513E5A804B3AE8425852CF5FB9F51D2601F9BFF9F5E4512A45ED"  
01225 "8F615A6671315590DEC31720387533A4E090646F17244EB571DBD6E6A10890FF19809D935A"  
01226 "5E8822BD424FF2E54B1A37576C059A756290BAAEA5E36AC977526339CDFE915A763B2CD065"  
01227 "7F3A3D385A5D92466A44980AD8A1B4E1593DC177E2447ADB53B1EAE15A91BD6703E46F01AE"  
01228 "91D6C316C1DB983688576B3B2BAB280FF196D5BF545755097249C517E1271202E3856475A3"  
01229 "C95324F827961DD7A827709DA2924B21949B6BF032675D80FD120180F6850BFC385E72399F"  
01230 "840B80E5525EE70B89E1C524D580390D487D0CD31AF3CA5691067BDBE8E2E91E0099F59700"  
01231 "CDFCFDF8B3ACBECB61DB23A28E80C3E463D9E53328C37EF8FF0D4FCBD287932C8197DF57E"  
01232 "EBC9DD829E2046C7FC14970F9F37CED37D8BB0958155192C8595E2A9369944DAF58FCC1D9E"  
01233 "1987DC4FA77ACB256A65DA78B733178FED14DA3E3E41DB808849132EB05E48BCA03ED499A3"  
01234 "14C875FF7C771FC8BD2478635396BD7F252ED984C10E2E23F55450F57ADC7EAC50A57B8BF4"  
01235 "F988565A172E878638ADB1D992703D0BB742605EB76E3C85319F567D71B9F2E40251EEB936"  
01236 "B323DE98496229B1F8A996AE4F6C0C2FBB3B2244F3BC57EB3FD067F535102", // PRN 34  
01237 "A7F21341EF8C75A1E9723B1FBB602EB4B92BE5FC8DE70183659A45A31EDDC46643ED1287A6"  
01238 "9DC876634231D3AF95F2D02FC9C0C7BBBB12319CBDBFDF6CBF94E47A05204AB14A490C6FBA"  
01239 "BCF329B5B0EFD8C2FA28845764510624398BFDED1B250003FBA61A67D2127550FF07A6FAE7"  
01240 "CB58FEB23D289FA235BB4959FE8845B2540A57A30E7D05D7BDEB721BFD4A998DAA6F1A547"  
01241 "12901551E30D618DC80FADF7CB7863EADB7EEE708F1050AB0510A956C3C7706756B6CEAF61"  
01242 "1F02656C3ACBFD9E4507AC2B4B78D0EB0BF757496C84C7260CA972F0F45A393A89AC2BD70"  
01243 "F15E629473D61FFA29BA18A468D8D4B1F72DDFF99608AC9E6A6D6A9231F24388991E80FBE6"  
01244 "A9269AD6D31182E79235A0E33CFEE9E3526FC068A9253789407AF8CAECE14BC337FCE02333"  
01245 "B63224A6E1913367CB36641A74E2DB2892C4E939A50127DBEFED7456F7E61454FA501A6AB7"  
01246 "5E6133B0417C2860B2F640ED8EE62D5100EA172BBF8FB0E2E2022D29DF32C25654C96F8FB23"  
01247 "05B72A92105E027444EE0F27769946A67E58D9A26AD01483FCA4BCDA4C00A8119E355CE733"  
01248 "474349DC8311AE9853A064931FD68D598B02E0724A13FF63D88416FD5D03A962B5D8F79586"  
01249 "E696C484D7287773851831EBF33B0158F51C89C493C116157D20DE66F3990061135F5CBC4"  
01250 "6862BCFDA5EB0322793BD6AE188698A2E1DEDD7F1702B6CB01EC2BDF85680", // PRN 35  
01251 "FB2C40A2713D1E1581890C14672C1F232E8A1332655108F24F78E5363BEFCA3DAD18D19FB"  
01252 "B1E91C6004761017B37FA76F70BBBC4C8E015384DC9E9EF14C6AE60DE55974D448DB8C36"  
01253 "B030DD55DF5EC6F6C56DA8CB9BAAACF6784628B669913A7B291632730C11710D7E181401FB"

01254 "535FE8A9A0B81578CE54123A6C3897496375CD7C60BB4E05AD16BDD4CB8F652448E3020EDF"  
01255 "4D51FF5EBB9FA93A740AE779419002994348358CD7827A63BABCDC7C4501CF8ACD77A220005"  
01256 "9DFF5B180AFFFB3CE52133C7451CD108AF9CC8FA32428AE9ABC55D89E1931E6607160BF960"  
01257 "15901DC097B56299F9F358DBBE921FBD7D0DB43C9C5BF97468813C0AC632D35C858D8384C3"  
01258 "2EBDEE933C9037EC36D2322F54FB0D3CF305801DE42B4EBCB5F83F3D3FCC1AF56910B7A547"  
01259 "F50F851079B6DD0262676145C27DB362AF383037FFAD5E3B26BD2095143B361F8242573484"  
01260 "2E2153C5F428D89DBA45EF59BD9DF1FE1BA941F172A685F08C18B76B6A8FDC86EED2867A57"  
01261 "AF48E00467993864EF70D80CE5550C5EDF085E94435ECA89F6513D23A56235739534C777C1"  
01262 "7BC6DC7BF6E04BDAA9B0C94C4D7AF1368C6B4B9D3C97D25A26B05703EBC4D1F8869C5DDA35"  
01263 "9D189F4A3E548D53731FB429F3536880CAD9FE13F70DB0A080B949C3636611CBFF0F644ABA"  
01264 "B56D2C6E988B3392A7452A8C8E52F2288560331A78F4CF021D2A41B2ECB6", // PRN 36  
01265 "CB7A566070A06CE759017D62CF78A429913370E6498BCDC038C1D3079A0A3E6FD4FDC851DC"  
01266 "BEA9EB3A7D266479F75C40452A282CB406E9A2581372BE2FFBAF6E7F3359EAA594631CEF87"  
01267 "6FD62DEFBB3B816EC98A1F72B55F6023B572A51F320CD7ABF489AA905CE2274729C0EE0F67"  
01268 "7A04D7449A489BD02704CA65BCDB8753B689CB90C87C73FB4A45219DCD8179AB584EFF049"  
01269 "EA711CD3E24CE41423FEE6F258A65CA8EC3A00D45371EC5B846CF1549CB5CA4FF14B696140"  
01270 "D3483DDC7801BE6CFEFB9189B5C7AC3DB9C0709F630D80F361442BBD22636B00530EEF21D"  
01271 "96D4A12CCF85E0124939B025CF35A0A5F1B4C2FA5B5DFCB3772CBFEC1F0C1C824B5CC22BA8"  
01272 "FB2DC92164FCBC6033EBF920A14599F1E8EFD4E8CB0257134947331D820595BF5630EDF00"  
01273 "814C32994039B47E8784C29D0D1C5B5D219A9F55FC14CC18DACBE6A79698A8073276A424BA"  
01274 "9A6606DFF369CF85E0124939B025CF35A0A5F1B4C2FA5B5DFCB3772CBFEC1F0C1C824B5CC22BA8"  
01275 "E12EC2BC1EBBCBADBDC631CE7D61BB9DDF143C3EEA4E36EAAEAF001849812A7CDE3EBC194"  
01276 "A51EA2A9A2AA15A0A4F29C2FFD8B2CB4ADEE99B59553BB2398273C04329A5C884989D417C1"  
01277 "31D68E9B505594C9B9CF1C13AABF05093A9E4224D71B54C6F61F1255E25FF7527E47768F049"  
01278 "51523F42915133E02CD7E404147E7AAE95BE651BFBB65E29130D0CA31C2A", // PRN 37  
01279 "D90A7D104DEF06F51CA5FE107C115C130B8823C423423608DEE00E0C2E6C202B60C5D20BCC"  
01280 "9CD018767D89E5E7857BABC9531C21B08436AC14C5B34EB986E5521369BC3D9B1FCD07C992"  
01281 "40FD0DA306C69303B310B8F78D2BF97D6E9E88843234D73214431F6BAEFE5109AFDA22C4C3"  
01282 "48A469370BBF980A81CF27ED919A0A48DA3E9061113C56659F7561175A608B7FB37A76EB9E"  
01283 "SABA5F4A6B485EA5D097482A28F4EF7DE785D31DDEBFE41DFF7C51CA2B64A0D2A7A70269FD"  
01284 "E399FF7628202289EA97141F995FF51DB5AF7C9C5CA7F60AFB4981D021FEA9EE6A49ECCDC1"  
01285 "8AF290B6C231265247DECE33B3E165FDE4FCF82EA84A4A8AE3FA90972D96DB68550AEF21FC"  
01286 "E82E0FC75FCDE33C2B6F659A873AF09067D94B068DCEEDCA4D392BE5077587785CB6A1F194"  
01287 "7841F483BE5080E85B49F1073D4778A39239266DB2A24B57B5F51B1F021D05C634D66DAD08"  
01288 "1D1668EF0520424581E5E0B62ED085D005F169950F971E1493A49A60A595342AACBA236EB1"  
01289 "6D5EF94E749889E50656D98F94EEF205445AC027387AE1363696185E24749EB67518B89057"  
01290 "899548BABE29F3487D273C54EE2BE6254C7F3ED3D3B22FA3FC94791A7BC7C499A391179368"  
01291 "D9B1F0DDC325F55022F5AB7B91142E2880BAC761D154C11F2C58F4A42E8B3DBCDFD44777F08"  
01292 "A74676DFB7039F24C9A4C4620726FE12849C9D3C15293A3DF7F4870DCCF56", // PRN 38  
01293 "9D238D97CE04FFEC0D292181D145EA0EFA727DBB7BADDCC60357762D262DD7D7A3BB0189E05"  
01294 "BE490018ABBB61B154F94A4FA81985E774019CCCE4B8ECE9858C0C2BDFFD7923B3354A2144"  
01295 "ACCC5C9DA3A5E0299EBF68E9BA0A269760386E5531F8964F8024ACC99E7AC3F40BFA9D94EC"  
01296 "A1B14A1B173A06867100C57883FE7333C079C5871A8A7603395A87387EFB92A77C5812EF54"  
01297 "8CBC3B9178E47E45DFA8796BAFECB5D81687EDC0AD5F804FF36A9CDDC58A5F991A2A0BA9"  
01298 "C7CEE893E1670AC2CE4D01217474E0226FFF30CB78597726C74B95732E44D03A626F1D1DBA77"  
01299 "708E4ED6D3C665FC788A2C1D286750F4B6D768CBF1B6EDE8524C2A7457A8240A491DB5EC86"  
01300 "8E6B95B0217A0C4E785290156943FA091F28234C4CAAB586F8095895FD21582E0356BC15E"  
01301 "C7CEE893E1670AC2CE4D01217474E0226FFF30CB78597726C74B95732E44D03A626F1D1DBA77"  
01302 "85F55859B71C2DAFCF71D99EB99FFEF4877518C579D2623FF6EB326B8D6C7674826984CCAF"  
01303 "98CE41724183952EAD69F05F44750F3B857EA01475555FD25A828BE1DBEF5E8B1C1237779E"  
01304 "E69D35E7BA9ADA124CBE710490280B64B2DC5EA2C41D6644012A18B369E3CA2FD93A254C4"  
01305 "93506C54FA6CB4DFE24E0B5035230F2BDE6FF21101164F322B4B066ECBCA3BB4F3C476B41B"  
01306 "D02D8948E555AF74EA13228F07A1BEDBF95FA7E16F2BDC5D178E74FE5A1A0", // PRN 39  
01307 "8DD8032B8D741905EDAE449212146878D8FA15AB037E41C36D21B0064D65BEBB74A18F437"  
01308 "C080DB44EE93C2AB06DCC6C6FE7DEA83F270C9FB52CE00BDBF546348F0E6285A35A32A4751"  
01309 "2823DD6FE27B7949071D4B58B46241544AB97B0F6023C70AFCFF0C7541B4B9E1FC6B3C3C55"  
01310 "500251665D964871B9D1247B3A9728AE02FF47368053B1DE928B6A8D01ED3BCB18A3180EFC"  
01311 "9700871F480959137CA1480C3640547497ED7CDB4912ADE1385F9F35B6D27BE64E362BD71E"  
01312 "0981BE952B00AA20DC57465DA89B87BBE07F27454FB24B3B86766F95A5A5B4EAECA2620C3B"  
01313 "F87C0654E6561398E27F96178EB1062E42F6E6AE0A669FC6C7170F217C850E822B4E70C903"  
01314 "3A375BBCF6D62D11B3DAF7DA976D4A9D955B924E7F0A19CE77A53EDD424ADFDEE8558F0688"  
01315 "7E82A936193AD5E508938DD3FBFE02DFEBD98C2272DEAD8AD3B0FC86C3637A6CE694DD95E"  
01316 "4FD55F433E151D35DD4C9301772CA66322001110B9BD0B89A096C8C3A3431C100E39E3D2E65"  
01317 "504A770F4EB2F19E9D8FCCAF15B8E58DFB52A0B88406A48A036193F3EA9F8ED2322FC69CEA"  
01318 "EB9E2DD6AC627DCF4CB109EA05AB5DC5A5F92902F3E7BD457C240C958FCA7B17F2EDA12489"  
01319 "61C9C82E799BE0A3D60B3E27E42B1FF696DD58E48C7B832D8FE9529B9C8F075E67294C1B1"  
01320 "060EC6736DB73FEF7407B320D850B049D80F9F4D536361038C4BAEAD92F24", // PRN 40  
01321 "C1005108B11B578F56E6C24FAB973C5D1B4A2F937E4A402FFCC45267F4A3F4CF4EED3DFF5"  
01322 "30838A570000F7E523D1B97FF798A3C6E12F9704126A9B6FDF606031E4D8E1AE39A470FE92"  
01323 "9124FC95961B85A5A6E10A79CBA60F54F37D4CCB13FA3B0A3A1729184C59036428532E3DD"  
01324 "B3F7B8D373DC08842EBC2ABBFDEB5B73B2F9B7E83C81D5D0EB3037A32AC0E5B6DCC7CD84B"  
01325 "9289BE16106D1D019D47209AFD8DA78EFA0F9108377C187B170CD53F366B17899EB41903FA"  
01326 "1627AA245C07EEA5D6DA78039B3C4D170DD28870A40EE3A048D209C4088A0847D6859513"  
01327 "E01E0F4172C38D64F0B307D721C01495070336A753A75E1CA433FA973BB131E558FACC298"  
01328 "C0A6E934E9F717E19EBCEA11EDF3736D4EAF593A8E5AC8EDFFE4D2B17B613E024B41AAD19B"  
01329 "COA27A1E9BE9D25344D4350596B9B0A48AA014191511B26E61A9753D8A38C4E3BECAC93E67"  
01330 "CAB5696888C06B9099E9AE856486436FDD1AF1C193057AD1DFBBB14AF772F7D9B00F37A6DC"  
01331 "9E805A8DFD9D53A161885E5C2C2A6DED0C54D00A8E0704877DA2F1EB6F6D8EB93B5604A4E2"  
01332 "545D454C3D4F0D606AC192E170D44E85E2ADC91E54F56906909024DED6F9050BF3D3928D9"  
01333 "AE23D9C42576AE4DAD1336F47519EDE436C3B1829CD2C5A0AF63476C7928B32F5E51826D84"  
01334 "4A1DF765C5578F9C2E5F79024BFDA96FB3199A3C7B47D3755FAF376761B4E", // PRN 41  
01335 "9C6674915677A62A7AFD4FCB27B45F8E6DDDD08980B7DBE1131E6E2425C5ADBEE8474E93A8"  
01336 "A2DA4E95B195A6B83F2CF6472A36E90928362B9A1FA994A9FEB7BCBA7FDCB771F59FB43AE"  
01337 "D289CCEABDAFAE29113389EB98D6F17E5508D5976E11BC8A1E93AF9F7B1C81686265930B4D"  
01338 "334568E3F29E1C2F58A62572A610016C1C1C9E1D0EB3FEB2B3A210C59EB3980C44BC656F"  
01339 "A7C5E05A4472D4255B40B8A1604FE39DB85026A976310648D5C84CEBC87A8BF6545DC843A3"  
01340 "A0B64DC4CCAF2D2203122DDD75BA42E096844899A35A899DFDF72C26E3392EA03351DC78BB"

01341 "9F62F51D913F8008DB00969C64003773FB2014FAF97E794A45792495BD52D7BE7ACA47FF2B"  
01342 "F570CF88303277092B5B6BFF3B01D38A53E8A68B0D81FC2D1D375EB27C7AEEDD70679E8DDB"  
01343 "A6DE656442ED951478FB96A979B4A9091F344EF39AF23DAA886CFBAA8611C61686332C630"  
01344 "690109E2869D18EE7A2C21B22921B9E3DE40BF063E370FF64E7AFE160B7EBFC4AF6AEDA043"  
01345 "042552F5F36C2CABD339FE1442242EAD931D1B83968D1A31A7E32A0838401DDBB9C1034D56A"  
01346 "DCAF5942462EF63440FD70F91520137A50372D0D125A6285F7D715FD9225D03A109E1FC5EB"  
01347 "547303CD7708F88FEED2814607171930436B249924714E8D8E024C24B3C0C9E40127DECE1A"  
01348 "D966C3F9DF01793864615F291B7373D27B624ACBEAD3D371B8D4FDA823C0", // PRN 42  
01349 "A1B9E8AB6C4A43610768CC17E3B899CEDDB44B746319EC50BA7E006EAOFC30C9E9D67EF8A20"  
01350 "154B2F93EC265B800503BC72831800CE0C9FB6CCADC8327A840F69DDF29BE212BEFE4AB181"  
01351 "0026FFF786D9D3A88B29745EB61131FB47B385F83CA211B4A2BC0449F7ECE43B2DDAC94C109"  
01352 "55FF2E078AF573FFA7349907571AF501FA6A2FCF24E4B5676F8213475351C0398748FBD60A"  
01353 "5EE8A1128C0998E57D076A201453EA7C70B1F171E7D084F444311C9829CAD03E0C2330D7E8"  
01354 "D39DB6B7901C30787FF123CFBAD4E974A5F6412FB1BC927FD65CAD1F3AA4ED5E2D284AF625"  
01355 "7EF311222467DE179E89438524116D8E0A4A94C19A4D6D0AEOFC535D69C4953C2E43136AC"  
01356 "07D6A5774D59DEF1471B4097146C2124DF83D36678FBD7BEA6E258D826645DD88AC9A7EA5E"  
01357 "05FB49F2CB29EA9081323FFB2A00F3915D1F36675BA1C9AAA3B166D9FB2529150D3573E502"  
01358 "668E33DFA5E95AA6C2F1F106D69218839BA590406B4FE4A03C4B16E29E6BC31335234BCA5"  
01359 "5D34955880B702F08203198EB7BAE381B231FAFE513F24C8BA9E58798F35465C2679DD5BA8"  
01360 "B16EEDAFB04E5A807B9BE6EE4B3AD77609D2B9E0985BB23397BBDE08F4D0B21494A3129857"  
01361 "E37D13862C1FF77A57AC663C206A92F8358E6C05FDE1A9D68131BC04B76F6865E8DF5C48F9"  
01362 "424CD9AB6FEFB3B49BA8D32C923CDB8602C5367AA9ECAE48AEE6A276CF25C", // PRN 43  
01363 "ED1196222AA470CB4FC8A0B7D2C785EB45D232E06453B77DB76F60E5F7BAA0FAE80467EC8C"  
01364 "656FC27FFDB8BFAADD2368B41AFF69460495923942ACF85C09EF01BCED0076960E5CDBD36"  
01365 "061933C1B6037548D27932CDF969FD5A910D5564F3A6BAF896A2A69F40EDA76E813AA26153"  
01366 "0F686042235A39A76081F58D952662894E6E648BA72098D840235EC4A2B963F8E74B52F2D0"  
01367 "088DAC67EDF48115618DBBABC698C04AAAC0632475BEE65BB26A4EAA08F0209B75EB253B17"  
01368 "6724F580209F8B991808411F823819CC5F42A9FC1F0CEDBDC54C046EB1D1658FFF7252908B"  
01369 "09F8C1D82E5F6D605F0B17B25506FAF91D4B3D4EBCB7A705E5BCB2A6FB409ABE4B1B0B4763"  
01370 "0F1AB0840435F161878573F82A7C139F072659D80671961D614D9F7DA72914BD5D6F417E"  
01371 "C152595972PDFAE876176923CF1F2F745A6FC27E134CEBC2D9AD5CC733F619C1B296026B7C"  
01372 "4717D2086D91765D83AB76605050B810C542819BBB7CD87EA2305AB4A567B2C500D9913B7C"  
01373 "6046F748DECEFC33AF829EF97E03799EA50F7C9EB38990F0D466498C091C3ABAA18453B17"  
01374 "3F0AA3A486DFAFEFEE68E7998EA30E5C1349F2A54AE753963CEF58332A114F964B83B4E446A"  
01375 "44147B11FFC60DA25B5FA705F34725B1388B17E99018236BD646DC714F8B7CD3BF2648063B"  
01376 "2D3FB713AB94D82ED81D571D5BB4D3E5F2F067E2F2C90E8B6857E6A651B8E", // PRN 44  
01377 "EBE597534C0A6FEE9CE6AD10A80D2DFD160D7DE7A20154F62AFD7C3187D51D09DA39A95843"  
01378 "2CDCCB8D3FCE19131E65A30DE99DB796D93D0944223D960BA233798323D368DDBD407D51B8"  
01379 "0350976951AE050CF409279301939624C741D20D5CDBF4BC3ABC8723E267BE75707024F63"  
01380 "98CC6D32A7BCD0989555E0EE49C1A2354D1E961CA9910A5169ADFA71D56745181E0CF13469"  
01381 "D8F3C6020637B811ED8421A680FD4D515DF6B31C39ACFA36B601027AFBC861493D34ACEAC"  
01382 "FB501DF9A311B7EAB9D38D7107E6B5ECBE74FD35CC8BC8319DF9F1948653C94C868DC24AB8D"  
01383 "7E650DF870A3D1D87855AC9BFA50D425604037F3DA323192654F1296C8320AF934AF60DAC"  
01384 "AD7F9630EA4ED0A635678213F6D09D73A2544C779596990DA42079C23581E22F541E6A3B6D"  
01385 "3FA383597A8BDD7CA42930816D74FE8B12F92CBB9303CA5FA12BF5FE954FA7B53E0D88B79B"  
01386 "E263B4B55B42A7672F5C4F7E6E2DB4C02AFD6034B7371E01215CED3F73AB8E3419D7447781"  
01387 "AE5DEB4F4547C5549E9B0AD59940C162790650344E5ACAB33A187A58E29C9032963F088C04"  
01388 "4ABE73A855F1FC93AB8DC258E41977311B7997EA95D32CF1F0140CAB9E6707E73E4FE575"  
01389 "E0752816122C39B8D12F34ED6771EAE192D830797D8E97A0BFD00E39CA382D44F080BB4F8"  
01390 "FF6FAAFF2A82744CE14D3036E00EC2ED5C9D184ECCA300030D7BD77B27DA", // PRN 45  
01391 "A56990B45E9AA5F9523D563D35E854A47629A4183E85CA1995BB374EE5601BDBCE5C740799"  
01392 "9D28FF0F1B3ECTA7463D3C941C57B7EB71E6262DA3A7136627814EA6D28C8D16E91680B0D96"  
01393 "1AFAACE2CA4099F986CD45A2AB387E39497B13678447480514B4008F0FA3B95AD7D2B4751C"  
01394 "D680FC7D3651F9621E1355CEC76F10468ED33DE4C7D7D68B786B7F0006C604BCF893739FFE"  
01395 "6AE56AD5AAD07D5BE76DE1CD581B10EDFEC9DF65AF925C4C1289837BE1854A62FCCECCCE4E"  
01396 "2AB47B48664C467F5E4C276047501EABE444E051DEA933863E0FCB2356CC0C3817BA59B2E0A"  
01397 "D78BBFF4C3E4C35343331CD81C7F4C428B0418DFA4D6FF655BC5AF49C9A5E8A67BC265A3"  
01398 "F30800B93E42C0E66F117E6C59C85BA5E273234EC7212C7B7AC3D87FC0E0EC5FBAAE6654461"  
01399 "F24BBDFF9063FD0E3F98741C5622ACC8BC008EB0E3F3D57680F12E52FB94CD957EFDD49BF5"  
01400 "12D9131757161A73DD9650E3561F831509A908D3EABDBD1C3AA95AC32E6960E0D17A218A25"  
01401 "784332A88E85303DBEB117BEFEDF46882995AEBD704AEAE032668AEC6B8E5BD28D3F110C9F"  
01402 "8C1DCA8FACFDC1188A073037796555DC4FAA26ACB9F51B545D32C224424BAC3DAB20D3C08D"  
01403 "7848E9FEDB47547C25671C269A62163742B55AB4CF9406D2C5047F92C76356F0B90221E019B"  
01404 "31890C0B8FCFD04E6B5D06114B967D1358797238D5B14D824C8F0BD7B8BD0", // PRN 46  
01405 "D85AAA53F2DE2946EBC0A09DEBB6CB61D91D27C8907B90C89E20C01F681D33BF0DC70B6C79"  
01406 "F19E4DB64793505A0E055C33D1E07B7F5AE09FEFF0C8CEBE80C84373804A48A945CA406D3"  
01407 "A1B17CA787DA265C1D8FC735CB098C1AB37A0452F1A287B8E19E3A57E59EF8F2E6CCCE2F0AA"  
01408 "7772955977982B1F6880AE8717520753636591BA36E351310E98ABB77E26E2CA5BF267935F"  
01409 "C38DE173CA20C3A964B56924A5E82A9E36D005EDE6680D543C021C7A9F53DB69BFA3201880"  
01410 "B7797E90D31011AFA17BAE836505F5B337C23A07D6AF167258C314C3001291DD9FC565F6B7"  
01411 "FB74D99C1B42B496422B3D3E1A564769C2306E40264641EC14724301795B4D54F5FF8D6C4A"  
01412 "B3645B56C1D0FFE9B977FC5017E2A53392E7458BAA308C343AAFF5B46B808BDA1FE5C28429"  
01413 "9A9EEFEC755132F9D3B475E2FCD6BA84917C601B5E8340B5A68148A33563E40AC5A1E49BD8"  
01414 "D5FA77BEB689E6F4F68A10ECA112475531E7C1942AB9504B8F76EA7BCCAE4A19EF859FD6B"  
01415 "6753F07BDD9A19421A2A0B8F27643B2311FAD7978F4366C41BD9F609486FC6EC7CE833A00D"  
01416 "7A48AD816412BD201F40C93D136ABB8E329DDB816D1F59A1F3124D6314C64C9B640E4D5BB2"  
01417 "01CDB4490A65C73228E7805AD6632D81709AA443EBF29A4607A504DD5C8C0D271FE595677"  
01418 "B8E6A5E29E23D171D50B878D0D3F9416657E3997D077677B87399A6D1936E", // PRN 47  
01419 "A0E629454DE594F7693888C9ADB907F83EC6818D6FC0019B70077BF1B0E3C2CB16A263BE14"  
01420 "5EA8129FED6BD98F3D9120BAD28379EC8E36DEC69C59B37BD6C2B65D172C70C69C23E29A61"  
01421 "43121AEB9E6A9AB46C3DBC3979BC876F4357F6BAA5A0CB33C3AFF1654FE547D5D760E8CE09"  
01422 "5A9B32A1E5CEC1596BB7E0D56433D8908EA69925A5CCDCF01702811AD1103F237D36C5F12A"  
01423 "3CD938A61EF76A3387340499EADEF1855F892F57AC39AB419F1A397CD43FE7F4C5E609ADF3A"  
01424 "2A8E401FFAC870985826CA0DE211996A3549A203B798A5162E98FFC8B5CC2CFA68D618AE80"  
01425 "D399D8DFE0FE4AA81ED9B58EA89F646D5C0AA8B107506C7F86879419D0F717E8F70C5A9DC"  
01426 "2C3F122CB53B691087EF5971C27086512F1890494ABA38B161996F98846B8876B95C5614"  
01427 "2F5B171502FA44ACD0299E82FC94EBDC4CDDBE1F5DF978871128F79C1758BB512D5C3BDE59"



```

01428     "A89786CF4E4716F0515F45A1CA31B06B23E4883F99EE0488848EB0A60FA14AABBD41EC5421"
01429     "D0C162590FEFFEBB4DD0300221471DE8DC0C87DB3F1D54DB11744F8A7115DCA0595F6CD99B"
01430     "74A6FB10C0D14E430F8B40E3E7105C89B2C8F12B37A8879AD9BEAD49AFA1BD56569713CCDA"
01431     "3FB634C3011D51A8DCF84437251A7B19EB61FCE5DF6E2278784722BABE384BA5424646FA23"
01432     "F6D10CE23DC5E0D8D512E7C5C094B1627D6285B57EDF38A7DB1BD57B72044", // PRN 48
01433     "CD7AAC98501F29507EA4E0183E8A40D2E5117E47BB5D18D01A3732DE4C821DFE86521CBEA7"
01434     "DB29BE1148BD544ECC681689BCD1B41EAF755310B7659342F8EE11CB41550CC30E566E1927"
01435     "96B66C1A83C0B28BACCFAC6C393043A0A2CB89712BC1CCB174DE58E66896AF39C1CEED1E05B"
01436     "0435F8CF6FD920D100F51584FE24879987399481DBF27DDB6286B6353919E552E669290CE0"
01437     "2AB4CD5113D7F484229F379C7332767EC69E4336439B05DE1C1E3563DD303A4F580BFF20A4"
01438     "E49CB8822F715ED0221EBCDB5DBAD751124B1715E82F37488265135B6C8BBCF4F801ECC4D"
01439     "3525FFF189493AD4EFF0C042B070C4CA8FB1FDF43D79F06A6E4E3D35D7B07D4B728D5DC54EE"
01440     "DACBBBA1EDDCDC07ADF7DFCFEF835E44DF1FF66DAF2A7BAEBE218AC3B15E183044D6A8A89B"
01441     "3C101B40BED97ED5DF93BBC1B84931D56B8C822A6D058AC74CFA4C85D8B456698E82D5B757"
01442     "4C17B041E5F4BEED09F75012355CBC322B822C63F10C18A8F279E9A0E18E1FEF183D23E138"
01443     "94E31F6D046956FE8A647558228F6D4D6910151EC03937876B6ED7A078D33DAEB3F2239353"
01444     "BB8181E62B286BBC41588DE10F478A5CE5B508F205A41820356767B0A0ED4B8DB9EFE34836"
01445     "2E9A90D6C30218B295B338B51C09239D02FC8A1E7DAAAB60AC37F5E67CFC88EEF69567B5C8"
01446     "1A03B449F4ED38B9D295A36AA3503173F6F6F66D93CE72D753076040FACDE", // PRN 49
01447     "ADDCEDB50E907D20E826E6E8A0D30C20C74B2DF204EA784BAE9F618CAE33A3C937729DF9CB"
01448     "10BA2A4C33E0182A37200C0CC509729D828B8A2A20F283AC4F9306596684EA3FB5492A4C9F"
01449     "2DB459E7531C9F9C0950E7D9E93B3EE5912AE7E39AC8F4EC14B18F24E325003F477E347C5A"
01450     "C1B67CDB11AF3BBBBCD0AC3703024B5767AA67A208254F798684BFD1D3EACD757EEC772549"
01451     "50A146620400DB95E694574F739A991EBA771EBBDF1056BB39A77DBE0636A032E17141332"
01452     "F951C57C6C90F348F165E3ABDD60D429D5D6BEC7E3E3463806F819EB2D212B3528A5EDE51F"
01453     "235AD100A35E890955F8A1DC51FDCB53EABCA2540997DD054C1F5B29462995B876B44D0859"
01454     "04E55E1B838BEF600A992EB49CE078DF75AF3D0F137685AC0D07F0BE1EB87B63A41E74DDE8"
01455     "69C8A683BDE60AF5D77FF18F7137495BCEFD0ED28F62F9C3E25D332B5F861D999FCDC0B485"
01456     "1A984A4DBB53401FD40351ADA4335C702BCC8D900C737507B990BDDBE91D201E3A0946DC96"
01457     "8D43FD10D04B0B76667FF5B4291C2124B0124C6B710A6D1BCFAEB016B9DEEB0F7A4FE044CA"
01458     "4EA0CCD84B7682617C3A545071EC295B0663B3F577D562DE1D9DD80DE6A1EFD6D5991EB524"
01459     "6F1597B86D0E9A90FC6DB0EB2B8E7BAE9431E567F01AA98502C773742246467ABF911A91A5"
01460     "1F6C1B9E0C3233DC1A37D17DB91A5F0F661B0EB5886964456C7818601BD0C" // PRN 50
01461 };
01462
01463
01464 /** \} */
01465 /** \} */
01466 #endif // GNSS_SDR_GALILEO_E1_H

```

## 11.701 Galileo\_E5a.h File Reference

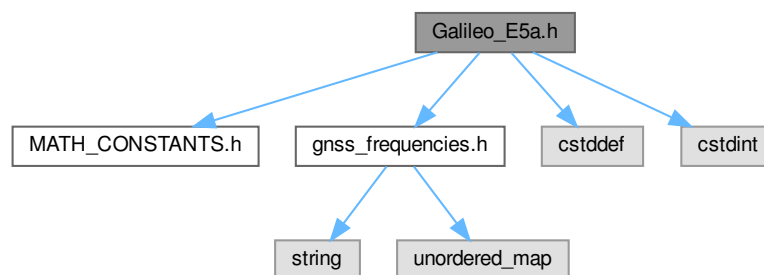
Defines system parameters for Galileo E5a signal and NAV data.

```

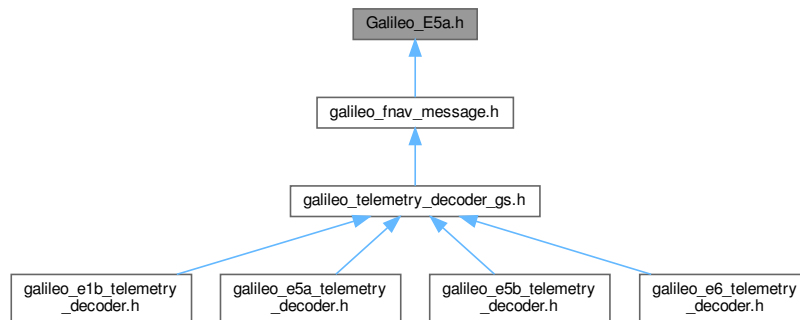
#include "MATH_CONSTANTS.h"
#include "gnss_frequencies.h"
#include <stdint.h>
#include <stdint.h>

```

Include dependency graph for Galileo\_E5a.h:



This graph shows which files directly or indirectly include this file:



## Variables

- constexpr double `GALILEO_E5A_FREQ_HZ` = `FREQ5`  
*Galileo E5a carrier frequency [Hz].*
- constexpr double `GALILEO_E5A_CODE_CHIP_RATE_CPS` = `1.023e7`  
*Galileo E5a code rate [chips/s].*
- constexpr double `GALILEO_E5A_I_TIERED_CODE_PERIOD_S` = `0.020`  
*Galileo E5a-I tiered code period [s].*
- constexpr double `GALILEO_E5A_Q_TIERED_CODE_PERIOD_S` = `0.100`  
*Galileo E5a-Q tiered code period [s].*
- constexpr double `GALILEO_E5A_CODE_PERIOD_S` = `0.001`  
*Galileo E5a primary code period [s].*
- constexpr int32\_t `GALILEO_E5A_CODE_LENGTH_CHIPS` = `10230`  
*Galileo E5a primary code length [chips].*
- constexpr int32\_t `GALILEO_E5A_I_SECONDARY_CODE_LENGTH` = `20`  
*Galileo E5a-I secondary code length [chips].*
- constexpr int32\_t `GALILEO_E5A_Q_SECONDARY_CODE_LENGTH` = `100`  
*Galileo E5a-Q secondary code length [chips].*
- constexpr int32\_t `GALILEO_E5A_CODE_PERIOD_MS` = `1`  
*Galileo E5a primary code period [ms].*
- constexpr int32\_t `GALILEO_E5A_SYMBOL_RATE_BPS` = `50`  
*Galileo E5a symbol rate [bits/second].*
- constexpr int32\_t `GALILEO_E5A_NUMBER_OF_CODES` = `50`
- constexpr int32\_t `GALILEO_E5A_HISTORY_DEEP` = `20`
- constexpr int32\_t `GALILEO_E5A_CRC_ERROR_LIMIT` = `6`
- constexpr uint32\_t `GALILEO_E5A_OPT_ACQ_FS_SPS` = `10000000`  
*Sampling frequency that maximizes the acquisition SNR while using a non-multiple of chip rate.*
- constexpr int32\_t `GALILEO_FNAV_PREAMBLE_LENGTH_BITS` = `12`
- constexpr int32\_t `GALILEO_FNAV_CODES_PER_SYMBOL` = `20`
- constexpr int32\_t `GALILEO_FNAV_CODES_PER_PREAMBLE` = `240`
- constexpr int32\_t `GALILEO_FNAV_SYMBOLS_PER_PAGE` = `500`
- constexpr int32\_t `GALILEO_FNAV_SECONDS_PER_PAGE` = `10`
- constexpr int32\_t `GALILEO_FNAV_CODES_PER_PAGE` = `10000`
- constexpr int32\_t `GALILEO_FNAV_INTERLEAVER_ROWS` = `8`
- constexpr int32\_t `GALILEO_FNAV_INTERLEAVER_COLS` = `61`
- constexpr int32\_t `GALILEO_FNAV_PAGE_TYPE_BITS` = `6`

- `constexpr int32_t GALILEO_FNAV_DATA_FRAME_BITS = 214`
- `constexpr int32_t GALILEO_FNAV_DATA_FRAME_BYTES = 27`
- `constexpr char GALILEO_FNAV_PREAMBLE [13] = "101101110000"`
- `constexpr size_t GALILEO_E5A_I_PRIMARY_CODE_STR_LENGTH = 2558`
- `constexpr char GALILEO_E5A_I_PRIMARY_CODE [GALILEO_E5A_NUMBER_OF_CODES][2559]`
- `constexpr size_t GALILEO_E5A_Q_PRIMARY_CODE_STR_LENGTH = 2558`
- `constexpr char GALILEO_E5A_Q_PRIMARY_CODE [GALILEO_E5A_NUMBER_OF_CODES][2559]`
- `constexpr char GALILEO_E5A_I_SECONDARY_CODE [] = "10000100001011101001"`
- `constexpr size_t GALILEO_E5A_Q_SECONDARY_CODE_STR_LENGTH = 100`
- `constexpr char GALILEO_E5A_Q_SECONDARY_CODE [GALILEO_E5A_NUMBER_OF_CODES][101]`

### 11.701.1 Detailed Description

Defines system parameters for Galileo E5a signal and NAV data.

Author

Marc Sales, 2014. [marcsales92@gmail.com](mailto:marcsales92@gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [Galileo\\_E5a.h](#).

---

## 11.702 Galileo\_E5a.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file Galileo_E5a.h
00003  * \brief Defines system parameters for Galileo E5a signal and NAV data
00004  * \author Marc Sales, 2014. marcsales92@gmail.com
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_GALILEO_E5A_H
00018 #define GNSS_SDR_GALILEO_E5A_H
00019
00020 #include "MATH_CONSTANTS.h"
00021 #include "gnss_frequencies.h"
00022 #include <stdint> // for size_t
00023 #include <cstdint>
00024
00025 /** \addtogroup Core
00026  * \{ */
00027 /** \addtogroup System_Parameters
00028  * \{ */
00029
00030
00031 // Carrier and code frequencies
00032 constexpr double GALILEO_E5A_FREQ_HZ = FREQ5;          //!< Galileo E5a carrier frequency [Hz]
00033 constexpr double GALILEO_E5A_CODE_CHIP_RATE_CPS = 1.023e7;  //!< Galileo E5a code rate [chips/s]
00034 constexpr double GALILEO_E5A_I_TIERED_CODE_PERIOD_S = 0.020;  //!< Galileo E5a-I tiered code period
00035                                     [s]
00036 constexpr double GALILEO_E5A_Q_TIERED_CODE_PERIOD_S = 0.100;  //!< Galileo E5a-Q tiered code period
00037                                     [s]
00038 constexpr double GALILEO_E5A_CODE_PERIOD_S = 0.001;          //!< Galileo E5a primary code period [s]
00039 constexpr int32_t GALILEO_E5A_CODE_LENGTH_CHIPS = 10230;      //!< Galileo E5a primary code length
00040                                     [chips]
00041 constexpr int32_t GALILEO_E5A_I_SECONDARY_CODE_LENGTH = 20;    //!< Galileo E5a-I secondary code length
00042                                     [chips]
00043 constexpr int32_t GALILEO_E5A_Q_SECONDARY_CODE_LENGTH = 100;   //!< Galileo E5a-Q secondary code length
00044                                     [chips]
00045 constexpr int32_t GALILEO_E5A_CODE_PERIOD_MS = 1;             //!< Galileo E5a primary code period
00046                                     [ms]
00047 constexpr int32_t GALILEO_E5A_SYMBOL_RATE_BPS = 50;           //!< Galileo E5a symbol rate
00048                                     [bits/second]
00049 constexpr int32_t GALILEO_E5A_NUMBER_OF_CODES = 50;

```

```
00043
00044
00045 // OBSERVABLE HISTORY DEEP FOR INTERPOLATION AND CRC ERROR LIMIT
00046 constexpr int32_t GALILEO_E5A_HISTORY_DEEP = 20;
00047 constexpr int32_t GALILEO_E5A_CRC_ERROR_LIMIT = 6;
00048
00049 // optimum parameters
00050 constexpr uint32_t GALILEO_E5A_OPT_ACQ_FS_SPS = 10000000; ///< Sampling frequency that maximizes the
    acquisition SNR while using a non-multiple of chip rate
00051
00052 // F/NAV message structure
00053
00054 constexpr int32_t GALILEO_FNAV_PREAMBLE_LENGTH_BITS = 12;
00055 constexpr int32_t GALILEO_FNAV_CODES_PER_SYMBOL = 20; // (chip rate/ code length)/telemetry bps
00056 constexpr int32_t GALILEO_FNAV_CODES_PER_PREAMBLE = 240; // bits preamble * codes/symbol
00057 constexpr int32_t GALILEO_FNAV_SYMBOLS_PER_PAGE = 500; // Total symbols per page including
    preamble. See Galileo ICD 4.2.2
00058 constexpr int32_t GALILEO_FNAV_SECONDS_PER_PAGE = 10;
00059 constexpr int32_t GALILEO_FNAV_CODES_PER_PAGE = 10000; // symbols * codes/symbol, where code stands
    for primary code
00060
00061 constexpr int32_t GALILEO_FNAV_INTERLEAVER_ROWS = 8;
00062 constexpr int32_t GALILEO_FNAV_INTERLEAVER_COLS = 61;
00063 constexpr int32_t GALILEO_FNAV_PAGE_TYPE_BITS = 6;
00064
00065 constexpr int32_t GALILEO_FNAV_DATA_FRAME_BITS = 214;
00066 constexpr int32_t GALILEO_FNAV_DATA_FRAME_BYTES = 27;
00067
00068 constexpr char GALILEO_FNAV_PREAMBLE[13] = "101101110000";
00069
00070 // Galileo E5a-I primary codes
00071 constexpr size_t GALILEO_E5A_I_PRIMARY_CODE_STR_LENGTH = 2558;
00072 constexpr char GALILEO_E5A_I_PRIMARY_CODE[GALILEO_E5A_NUMBER_OF_CODES][2559] = {
00073     "3CEA9DA7B07B13A6CC0AE53DAD1EE2A0FCC70009338C08AC0EE457F76A1690815C3C940AB7"
00074     "22487CC8F3D1F4C428828E7FD2A12130E42A3BBDFF1E792165F644D0E0335F95EBDC93D6005"
00075     "CC0C680DB7B0E1B8C4946B7974319F9816141DB9E01011E4F20DA8F1B8E15A6F618CF599C3"
00076     "F5C1A1B276D51318ED4119BCE0ACD0332F3DD8F88EC5215AB311C51FF4987DA93B09A43BA8"
00077     "4CF08032F6CB28F43043C54586811D870AD6FA27AA63785345C8BCDD3DA26A0134738BC7E0"
00078     "8461D5409FF0B791D8574CE797FC5EF7821055028CB4AF92AE1088F8806CD55F0E5FDFCD8D"
00079     "74ED801B2B44AD5D79D1924D41DDC6AB2070B5360CB64CCF487FE517420348CC39BF50BDF7"
00080     "8BE7DA91542FEAB689457B3EE69E43C75FADC303F31032FD96B7DC70A88C3B7BAC7322B285"
00081     "D9CFB3A93AC8B890165F23848FAD8477DBDD3D0AA4CB3CD73A48000B6D134DA2DA70B56E59"
00082     "0A101AEE78864DA0C64A7BCC6B37CD6F31E9AFF10CA4D47630752D253946432DF6EC60AEC"
00083     "CD223F29399CDA3B74D1DFA5471277EE6C814464A8C55D3C0B83B36B6AC9FA90CE876ACDF6"
00084     "5E3EA3DF61D309EB71ED29A3D510B2F4C0B6D6C5B57EC9060CFBE48389DCB17CBB2284E7F5"
00085     "78565B91503B06F49CF3E8534870AEB6AD9707265A9A1E6E2E5E6DF6DAA367239A96FE5B02"
00086     "C19A4543D537EB4D9D73966C09E9B5B24706F57B3E0987885EB84DEA26F7823D895F620151"
00087     "88ED38C04CC6714F797FDB0BC713E3D0208462F9A68E3872A167BF1BF9791AEE8BB73CF527"
00088     "C50975B55C4E52F2E95B677F833ECC878D1764839608CC1108A75EE9E58FFCFE4CB52884E"
00089     "7AF15EE0632E0729DA1CF5B7A227028CFE1E08F8B881E1A743D52DD27BED33DE0EE75DC031"
00090     "B4864CF192DFAEF64F726D73321363A233F81C57232432D2B0A5A4C44F4320847A9C143F37"
00091     "BF204185D2B571482FE45D6BCA152E6EA7223BFC6DCE06CEF90CE9114623EAB9B1EC789B20"
00092     "51B4AB711DABF5B16FCD970F437B8860313B4F1F14D384EE3976B7E55D2FDCB7E1BD9BE18B"
00093     "722E37C853ADC7E1CC2870A02881F95B78487780E1D1C296415109CF07AB63D0782A9F451C"
00094     "BEB3E8B919917AEDBCA8A8E563AD3784639793E0F25CC9CC62240FA04B2F141E71BF5C84EA"
00095     "C5643115956B8BCE077A51469A87737D3D6F06D97DD479FCC35129F4499C19EF98BDCEA9D"
00096     "4941B3756CDE1997C3AFCAE62B6D9E23341E11CD05A7FFF52F5814011A84D737E126410900"
00097     "B6EF5F1E3C6A9C7521B44741A8282755A8F0DC2FA0E1F6CA4FB34D8CD5FAA27E188088687"
00098     "25B9634376137C1BBC46934F83958112D03082DD6148F353BD1DD24B9F8FD7AD89C40DA0A"
00099     "92ABDBE360838CD56FFC4ACA35241D76FAC4CAE1211AAD9D73D51C81C59BCE05F71C34573"
00100     "0D3A2C670F8F533A950EF24B00EFE6A3F1354694ABCC6FD9EC4E74DDE1F287AD4F847A297E"
00101     "CCCC39AF029EFCDDDB19932D906B9CEDFCBE0D422CEE305DD05E407340F28EEEA866664D60"
00102     "AF293A45D5D65C0000B05F79463DB513ED488DE7BD4EC9EACFEF973B23CE4E9539EFCB797"
00103     "456CF5FD1EC54FDCEE80B39063C48B91A5C2D2BEEC81B9B46D0AD6503BE5AACED2BA5EBE81"
00104     "F630B4E07510356E8229F7FC5EA532B8729CDB819E066A15379AC6942CD4BC5E97C6791E09"
00105     "8105C323A3A3DA3880D5EE5562ABBA2BDC9906F4486B51ACF8AA4405E9D7A63DB9E3058782"
00106     "DD9AF3995FFB3D34AEAF98234A0B3DC62C339325B60706C068F0198BD8FA658396D06931B06"
00107     "9155217690CF788FD230CDB38E3E48530BD47722FC", // PRN 01
00108     "9D8CF144C4B667345D44F765622A956CAC4E097AB1CAB05CFBCC6BB68C709503AD9DB09C09"
00109     "C983D46A0A405B6F7EB26DB4D46F868C10E112828B1AEDB3C0074BE0DE3C9B7821BABB4F8B"
00110     "8E24F69869CCD981B09A783BF6A95F39ECFAF25DED6B16F89EA09D3A8413CCEBB545651B36"
00111     "3DD385D12BB72420440C40E804FA27DE029A1E08629BAAB598C035DC58FDD309844F3BEBDE"
00112     "40FCC231F38605DEDD06572ADD85DC51D3D8B89B4480143D0B75283522354330E5CCF4DE1A6"
00113     "E68047D5B8D45B835A891F2D40C9DB8A76CEB1D18FE2BC38D080A8D97064CC87D692DF2118"
00114     "4ABFDDA7642D0BD6F3209D06B4AE7600F7DDDB71DA751120599117ECCE645FD109CCA2EC7D"
00115     "B98F4177F14DB854FEB314B5D7CDC3385AD203464EADAEAFF4AD08DFEF3D21240BFB8EFCAC"
00116     "1356C72A0F5C61BE03CD2A21A7D756FA9003D562FC4A49A6BE788EC8D80054ACA881DFF72"
00117     "C2966EECD09F185EDD11218C6696DB14E05FFF3644D11E508F4F1E9C5AB3074FB1C3FB2109"
00118     "2A1C8D5AE05688FAA9226C3C3D0BC3981933DC8648240F8CB67085F53AC5295428D8447"
00119     "A1E5A46C2BA86796982C4C6CC647FD8079BC4024BB69E2B226E6F3D0F8A90B4D36DA2ADE4C"
00120     "6BB60D318AA7479FDC2031143C67CB4381C27072E12935001524C7BECEDAA9954BCC2AA218"
00121     "E9EC2C95498FD8DF655C015896D9ED42CE7F91CBBA2CC4A7920038EBB5F5CE638F969F8B17"
00122     "9E72AE25BE7E826E5CB53C2E85AAF1E1F1AD8D534F78A681928818AC3154651FFC583DEB0"
00123     "A6A1F40B987711ACC528AAF80D210ADAF83597869968D499ADE9A19BAF341E8CBA20F0E1473"
00124     "BDD898C224C7A5466F9924EC7EE992A2086AF295BEE1F6D0F8843D91180BF2C981C11FD978B"
00125     "23B6BAF7786BD526B458B76A87C31D7C52FDA43F3D362C8EEFFB3FE5FB3F6E5F34B1FEC7E"
00126     "F1031146F3F609B32677F148F7DEBCF3526BB45582436A3092408193D6312626E46ECFA96F"
```



00127 "EAD12A234CACE10FAF9DE75EE2D238088146328E10E9ECDBB0B018ECDF2725415CF5A06AAB"  
00128 "857403BBF6CBFC350903A982864827988BC805A3484A31FECF7A40D4FE251BC7E487613B9D"  
00129 "3A48D3C7DAEFD49C4B7E625F868DB53A798515A61050978552699EF2A5BF2F13BDD444EAD"  
00130 "C9B60B479FDD4633EB4C1062AA78BEF06692DED203819D3160310FD7F2343732156A9CBCB0"  
00131 "B50BA9A8F93E339B702670E54BFA6DB2E2E773202C690FB71EB03671AB0B1B02B2F189BD99"  
00132 "061ADD23F75F4914067AE638C9A29DD3661C28AE272CE692CBDE6AE880FBFCF272E54834237"  
00133 "2CBAF6370C7E3AE9648341CE7310BE1C534B5702B0611AF65868F840B6B7613FDAEA21DEFB"  
00134 "4F2024487023B02B8B58C9E9F27AA787EE775249EFC40913CBBD69C38538F239B203815F00"  
00135 "F7B9CB30DC79E6A0C3E069D109E4A1BAEEE36D354C3D0121F1342F1F4AC504A68D69DEC158"  
00136 "D54B04BE8164B48F31BC0827A0379C5237070B6F963741AD9ED4F3865698FB8233D7F49ED4"  
00137 "E0EEF3AD927CBFA4FAE183252BC56AE4CDE3E329B1D9C87C6C11429B15B8EE589213CFAC20"  
00138 "8A12AA01B4F1F7CC35CD0AEAE217471B3DAC1C279F353DC61994FC45FEDBBE0005D8EC729D"  
00139 "85645864EF98A3A417E62F1EACA7E60D4E773BB2E4024D62830F103A7988733DD7BBFCF3AB0"  
00140 "CD0049006FE2F7EB3821724BEC37EAE44681A9699A025D212724CD98CA3415FE2BD09FADC0"  
00141 "2F1501FA38A6083427B662DDCBD0460E12A09072698EC8966C47B8A640AC79C1B7722E78A6"  
00142 "C28680F4BB77BBA477BE0A6FAB959B9753217C5708", // PRN 02  
00143 "45D1C8FF162EE106CC87C3EBF6A837930F8CC797EC7A446E8A213ABD239582350636B19B5B"  
00144 "E428BA9C13F980B7AF5CD7F32630AFE8693CDF0EC0BC2C84F2472F5B86576E8C43136C14717"  
00145 "A24705953D392BAC96C1055B782C7941D82FEA357E5FDEFF772FB9F3DF248455CADEAC4CBA"  
00146 "2EBA9C91184006D1680E00D59E4BC8FBEC22F7CC2E78BFA5B60EB292F244E6CF497D5A287"  
00147 "432F2520B31F9D9087A5F5CD7F32630AFE8693CDF0EC0BC2C84F2472F5B86576E8C43136C14717"  
00148 "00D81463C388CD05C677BFD4953D2627105B0A776960FEE916C75D53981D30DC689581B7E8"  
00149 "E0723D65949662ECFAA6FCC9F0CE8892E367721718F906207663F9AD450AE98D75DF004080"  
00150 "FC15DC2CD7A1DCE013A0E547ADDC29A397ECB9E7FA02035327AC40240E2091098708D42456"  
00151 "3AB7C5867F3F2D78EE3EF5B658FDDBD49435060CA2EA3D559CDE957B7E48B98DB41CF875F7"  
00152 "B3D9EBDF6547B4EDD98DF4B747B0793152FA8CC07C6D9EE5A2002464566D86466C2EDE54A2"  
00153 "BF4BBE8230F480E87AF5CD7F32630AFE8693CDF0EC0BC2C84F2472F5B86576E8C43136C14717"  
00154 "E91D489CCF24811C0890EF905E9E2ACD399E13DC8133A54BDD295B872EB74E412E2FB654A"  
00155 "9874854FBC3A68C73434C5FC5CED27534B2B13C316205FF4E432FAFC13A7B5B7A7FFA9FEED"  
00156 "B5AE68036F8F2955DA124CE5856E8C53F24E609F7D3386DC5212B2E78B5AA23B59D45FE98A"  
00157 "A08E9CFAA9D52ED260A36AF07522C047ED43808A39D7019E444EDF84D885A9AC84092A0F6B"  
00158 "FED562F3E0D79BF5CF62F98E67EF219FA3F5AEB7D4E344642D3D4B1A7EEA18464F6CE8D4CB"  
00159 "3181D9EBF64122751B54D0D7F3FC470A91B547148AAB1CA0DF59872120190640555A7561B"  
00160 "0F2C11280768F74B1A56674FD5480B0F510491431810D99CECB6D88588BACBE2B020FB8B"  
00161 "3D78039773229714156494EAEC3A2D0A59E718F72205747D69C05DDF1C678E2E154A1F84EF"  
00162 "0CA2E24DC4A6A996F0850A396D2432596AE84AEC0935B8C25D5C65B5A23272F01D281C4F"  
00163 "753EE03EB10020E9FA02462CA303DA3956066963753D381EB78AE5EC0F6DBF6273EC97944"  
00164 "2E6243F65FC51F26C6C9554C6C0E3EFF33BC4EAB6A27CAB9383BE7DDDE421804998033B479"  
00165 "19503E1C9A789711EBEAA6C0298B3DC563F54D28675260F6D896F1B8D4FD0001C42921039"  
00166 "E89544B3DAA12C31F7EE82EF4D2234E26F873610B76756DDACD24B6132BCFFE735FE75513E"  
00167 "D527DD04D7DC6D24059F85706679DCD1474A9DB9571426BE17E6DEBA58B33B708567697F47"  
00168 "1CAB878BF647380E18CB6F88BF9E4F442F0FC21FAB89305484828F18B65F9D373A6A2B38D0"  
00169 "73F5924F80DA234C1DD87416D025E4E663C96F287B0C83DC92C2164D81830781B715209FD1"  
00170 "1A65E64962D805389BAAA91DFBB990D3511E506A8EC101131C5B7284252F861D047DB2C202"  
00171 "7DBAAD487ABFE429CA21CBEA7671350618E441F4D62F2D579CAE297023A8873869B55329"  
00172 "3D9F54DA929E252AF132325A6E3BCBF7B36D0DAFA1E56A39A5D801FDD0D5A41111017BF62A"  
00173 "AF8346C7D424FE007C32B437ADE60AAA9540AA5078FE6C3C3CCEA53EE863086646C976FE6C"  
00174 "79434A0AA4F53B2E9A2C3B4CF9C9C4015391E27CDF5C1FCC00BBF5B99715A1265F91E2"  
00175 "94D530DB14DFD485AD34BBCEA32E5B5D0EED15F88BF5D96D058E6D70BB1A232597E35A625E"  
00176 "5E8C2EF5E7031A71F70309019A0591BA0A50E87C839498255A3602C0FAE53166BE5E49E29D"  
00177 "24AEC47002B698F80FC49E718B66A8959259ACC540", // PRN 03  
00178 "7A01335CC3754D6B259A2CC4EC029811D098CDFB40549E5C40B36A2846CB4B256672BA18"  
00179 "9CD3A05293BB36B167508A7BECA3110BFF339BA06340585DE8EDA03AD244A77F54B7931610"  
00180 "B6F9C5C54D688A0526A9B52605BD7D7BB01A63F3D1565CF78ED904BBE4AFA4A290EADF9DC"  
00181 "156B59BAC162A818B6CDF38D2BC715144DA4A1578BFC727423777784D15ACFE80FACAB61F9"  
00182 "E58B5D3FFDDB065A00C5D49DF0237EA6C488D7758F1A689DC59DBFF78261016A7C723FE52F"  
00183 "FE571F58E0ED5F00A90BB82B27BCAF5A67374284844E06BB1B2D84B1D20228F5C2208"  
00184 "CA7E8EECE2E95027B09372A309223A15C132543FF3A89B7AFBA56AA7A8DBE70E0805D3A5419"  
00185 "1CC6884D75ED0FC00C06D9D488B0F4816E12D6C2A4324EF742AC8FF885E42100849DA0E53B"  
00186 "7C451D43ADBDDFD13076CD8D22BDA101F665B5878E321A009B970D1F4C48503CE3536554"  
00187 "3B3F36786802E5C5FDD756C595784E4F130904044660784ECD9C9161477F5BCEFC9898754"  
00188 "0AD1E86CABD3EC7823D83877605FFC79820F1381DA29282C3C5B3443B6A67973F0622EA5DA"  
00189 "A14F239542EB140082F242958B39014486E5D5632C62C3EC8BE0E09E038C0ABD52B1322E0"  
00190 "F7407FA53AE8D761858136CB371AED5E6FC9D32CADCF870EE833B7120BC0278D9A05554D9"  
00191 "0DBEB24ED6F5A8861698D48B4BF7AFE7763C1401572643E246DC65853996B0480D38DB7302"  
00192 "364409357137DC0B416561B9511BA43CA34341FEF7954C28B2D9FF96EA110E0818309C32A"  
00193 "D306DA077CC911299FAD6396C872F3F6AF7871395D7E67879FEFED929E4C05AB4C09E8BB396"  
00194 "048150A4161D7944CBD99C94DD16CD8E0D8BB73768B17EC02C0D4206AF623037D6F4257DA"  
00195 "4C07B4A6C0B4D2E0C9923FBADE3DFF7FDAB45F4E6BC5A895FAE4F5B99EA247F2D4446E260F"  
00196 "7988C452203EEA1DFA64DEC2DCC090BB3ABE13F6A8718F8DA2BE551407B59B8EF1806A6552"  
00197 "6B6B872CB8922BB929F09341554A71E69B41B60987FE3A5E7E3424D947455083A827FFE27F"  
00198 "B5BC5365C80998DE01CCDB66213575FB61B3E6F877D0E2E4EFDE4467D9F07B6A28148FE2FD"  
00199 "6EDC9202F55FC855D0DF8C49E244C0CC3D95FC06C6778D397461BD157F4A0FFD915799820"  
00200 "D5F52C96AEC0CC5A3E7A2151A845EECEE78B82ED9A217E326CF6C49F7D31C4D8ACAFD827E"  
00201 "6ABB760150203448C000819E7E0B6E424C3A5500164CF128D686B4810D9838480604A8917"  
00202 "92987FDC549D87F95BCA120AB84FCCD8C9F93F988C87E79599F3C1952BC0F7773BDDFC50B1"  
00203 "9BDF8E38F52D887E45B643297650044E80124BEA0ED60FEC4449BB3BBE394CDF7CC7AA39B"  
00204 "C1A5023044F6A843186C01EE1BF5834EB5401AF7905FC04447AE00DAC50B051B432F831FE5"  
00205 "AAA7506160CFD7D4639C489ECA447F4F993AF0503CE5EF68A837FCF85B85993ECC55A9A367"  
00206 "3F8F2C5CB8D3DD4C60E8421E3417EA958EF87E0764B061A39C32ABD5E0E3A712B54B0A3E2D"  
00207 "351A0E00FA4E901521C63C1F4ED829F6E259A1F720FDE96EA9CBB8F7BD7485531A81A49CDEF"  
00208 "CE725493A04B5EAA7F411DEACDE5A95AB6C2AA7BE3269F6AE4D166D8A5FD5264B135FF8361"  
00209 "FF75B2FF22A61905A349C6B2C1DB2BD7B385B8FBDCCB2768C7926F138D5F810711563CE527"  
00210 "322AC42E6BD42485668106B8CE91F157E0C94448869F7AABA255821DF981CD5298D40378FB"  
00211 "0E33A3BF8A037BC21F0AE268E69F7CF61E7E117BB463EDE3C7D2EF95987C66AC3E5C7D79C4"  
00212 "4A7590BFCF998683701DAD7B98731DBCBA455E61428", // PRN 04  
00213 "64D4236F326627BC08E9B2B96C1A9E5BA2631DCCA3F7A5B63736E4EA8074056DCD6BF0E5D6"

```
00214 "DB8845D9271C0D706F972AF22E652E3B2A7CC482B125EF8BE005F25D5109F6F9DD84DF966E"
00215 "2E088950FE1C01E2DF15205EC48BECEFB32511DCC39678F9DB08ED0EFF64C7B5DAED1DFD20"
00216 "2F63B6EDBCFB7E4FBB431718BBFA2B65594D78D3983B0457DDDB350AF2C1B9743AB9EFC260"
00217 "A78C3144622C50D528B7D47DDE46FB9DA33D1E7DE6D5829258C2F02D54085AD0121A1BC833"
00218 "9A28476F6161CB6EEFC5E3FFB12E6C77E5FE10E0D4B02E58142DBE5BF900B7B64EA79D520A"
00219 "35506AF4987E67BC880D00FF467A0889A14B85B4745888B09DF7B4D3316D9B4A9B17C4CEDF"
00220 "CB0A55E2AAC35B38705673B2A2EE4ABA914E20ECAED730E3D5E272A65A74C366177A71109"
00221 "5624A679C9300FEC18E20E65DA3645F23AABEF7EB7C7FF49B289FD2F44661F6F5F689B6E06"
00222 "EFCC99F8A5EAF558E10F2236E002F7D97C316698D6833792C456F812A339166E9FB3B3F617"
00223 "34B98AD2DC9A484A14A6025114D072B68C47DDE97322E213F5D6A93762A45C4C73C7832BB3"
00224 "1209C4B1F0094BD24BA23CA9378FB893AB078FA3BC4763704EB6A0C8EC965690CA5B858DAF"
00225 "E623AB88ED39154F46B9030462DEC9070DA9DC34063C1CAB3FB84D60337FD16D95D1A17365"
00226 "0C96128D3C036D0B72B4D2A298186CCA8E1F386E1F70C0716F0BE370A6B325CDFDEAD3CC67"
00227 "E021E9D4C839230708DCCB062096F32C4DBE3C4876A7D1A26072673A6CABDD8D65A5E91CC5"
00228 "E973F00FA67619F74930F9D40B767E434D0955D47FC17BB37E4B7CCE63B1D666AF5D67FE"
00229 "5FE5469D3DF6B6855F9C308DDFDF6733FA7B8511B25193C27925F3619F5F836014A64A2C8B"
00230 "783A50C6B3001001D621CEC582FA4E52C2B916A418F49ECFDB5ABCA40E1A8D38FE400FC2D0"
00231 "C18500FE85546A92E829DC1AD2A7AC3DD23EB77F3D80015BDA4135B1194B10ECFC87C1102B"
00232 "3D366C5460D20D83E778142F1A4EDFDEC3D0BC448CE24A49C609B035A99B64FABF071C58D"
00233 "F592BBC625359BC23A565F4BB077DC879E14ACE87F709D276982A3D10B23A9538A6A3DEEB"
00234 "BF0B712DB4C1680FF508F3E4AF4AA759738302D24C6D13037DC81B0394FE785FD14A322BC0"
00235 "0ED95E0E9D527FE5316748DD893DB03D5C4149D471DE98C386A2E9DCFF05181E837A1814D5"
00236 "4C9CA708CB5E7A10B0FF4F540338BE403E4B9713015A99260795DF5832CCE8D5F763C68BDA"
00237 "51133AF1D012C28388F4B1DDBB4EE8C351C4AF86FB436571F53B2E80AC04BE26EA61FEA9C"
00238 "5578FBBD81DD1809EC5BCA5DF4A85B24970EBD88D9856B2136616F85A338771BEE80F20F6E"
00239 "5652FAD00F49F3C349061DE3052841992EAA04CE15DB6C4888BD3CC5EF4D6CEBF00431CCB"
00240 "7C5289979E67FC69C2DA88AB60CB65D0F687455A535C33547553966D197BAD9778542F6CC"
00241 "182289AED366ED1CDB0C5EC1E9C0F0E21D4D2B8D01EF4BA394E0C6B05B8EF42D81F773D3D2"
00242 "51D6264CA32DA5CA2D4884544FDB81A243974FECEB416D0748449BA0CDC69E56D666FCD24D"
00243 "9BD7C322CBC038CEC658B5253227D87C527473F34BBA3E2FC18784F864233FF8528A874A85"
00244 "03840AAD03B63F79FEAA6C07E4467C4A04366ACD35E32A142DD76A23417EDBB8004944A1E5"
00245 "1880CAA97937493EA5BD5A41298863A71D33156CDB942C166748DF66B80ECAD27675A673C8"
00246 "69280584E97F1B2DCC9BCB58FDF8BE55E2034E73E44A225DDCA5B46BEBA51AE98A2799"
00247 "0F622B32B2C337046E459B173D94BBE5FDAE31FFD4", // PRN 05
00248 "23300D5D80ECA6A471DE5CBA1D29B7010240D95FE341A62FF8175ECA9566011AEDE6EDA0E"
00249 "2CFEFA7BFF9F49AB2C97F6554BB182B23F2772090FE7C4ACB8BE7427E7A8535DA3A670341"
00250 "D88CFE694D1CF40156E9B0557EA317A0B3E21F3A629D1CA971482B1A54697BBC2020BBACBB"
00251 "B9D1E67DC33F52C4B446A36D4331FB02F19BF1A647D05714450E591C65853176A04DD561E0"
00252 "933CF32D42E4707C44FE29C5D7706723FF53ADDDC67A67C23769CA32876F0CF31D233D352"
00253 "FF6A7277E5A3A7578A6F2A76456C0AA876680CDF2702B114E02D2D9F59077B9DB2ABFED67"
00254 "3158EFAE4C23A75FC8BE701D973169AA7015297BDCAD4870D4F152DC556A06BBBBAEDFFCE4"
00255 "0E6BA4F0FEB154F32D8E1492F74EE7085937600EE176ABFD6B8638E983EAD26C63805B9874"
00256 "5BA290813CB65CCE33DF6B98240E571DEC4BFB2430C4B8FEC23C1BD5C3E87E0E746BA8A772"
00257 "2A6A660C5095FEA8E1C4978B966F487376DF42C9668E5D3102F123FDC7B8D80E8BC84AF0D"
00258 "91E9355FF6F7482BA74E0BBA7DACE85BC053F31074151566334B4DDA37C6C51BB947811F28"
00259 "4B671FE53FC464CC3DDDE59A3D9CAD26822183E96D2F4BE60905B94BB8B167B734A0F0B26A"
00260 "D0E5083A34E68A620151966D073293E5D430D0726C0FE35BA3F1F44D851D374ABD80E826DD"
00261 "84A665165B3B82D540CDED9EF60CE48E87802F16F5DD89163AEC8E9C523E9F99AACCC6F001"
00262 "26C4C8663A7D64D919EC41ACA337379496D0CC876199A86404CE5844F8E2CACFB00A985B92"
00263 "B7A393CE464DB19E0B5FD65BA92D8EC1164CB1ECD994F82396F6A52BE2B66C8780A5AEFC02"
00264 "97E437F7FBF9BF8880B61C8313C3865561A4287BFCABABB90B3A11F6AA57C2B2A6047F131"
00265 "6AF61D5FE540C4F1D5270F3EE3E023E202AA530211DC51F8C2E87636C14EC8300271A6454A9"
00266 "24A8093716756F79780D94DC6863C24C80A405D6F52DEE81810EB850C1F60732AB24209773"
00267 "F66B2D2AEDDAD6FFF60902910D4858B0F706414F5779590F2DA7C249E5DC1484EC40EDBB0"
00268 "1920A5175CB9D74EFF957A61FE3E08DD7C5DED5B299C03299F25E39B161E1DC1C586E39D0B"
00269 "AF38C09C2EC0B22589AC489C3199EA4E66611C45A68A7254D292C78C3978C381F297D7DCD8"
00270 "C0F7646EBCDF155B67F10C3F915F9B3AD96C21379993D8C5D6957847EA81B3BB4BB0863"
00271 "F0CED12F5CC48A4325CDA65268110C1C156F845951AF3C3C90280F8883CE0236FF02DB0CB0"
00272 "7721261432C7E0D479F859D8EB7C433F67721B06E001498656578F0E3CEF2A6B941519885B"
00273 "BBB03733DEEA0780226AEB473CFEA6EB894F45D1BB937ADF5180F5FE22857CE0EB75D251"
00274 "B02D89E5502560A0B6B012C191DE9D62FD28CF503375F2A1FA9EFE0E42DA81A6EB3A6DBA29"
00275 "9726EAC6F3BE91ED51A25EAD5F3E7067720F7D4BB72F8BE2DF978C46E1DFB4B0EA17BFCE"
00276 "EBBF40C66ED1F288BE08D6CDD0B097C7ED1205E43F8FFB7086120FD153C47272188799D0F4"
00277 "554E9A4131C6B1460077A99E8198B3717AFE5E7C95D3F49B3779EFF9E935FA63A6F881F039"
00278 "436EBDC2AEBE6F0E0109B658BA555BAAA4A01D1FFCBCE0369798B3BF8B54FCEAA5FE25F"
00279 "31AA02208B0F072070F9E043BAEEDAD4432B1C2DAC9F7B4CDCB52965EC43C2E99764AD2613"
00280 "BCBE468C9477E64B8BDCB64CFCE01C06EB66A15FB034D1AAA507AFB6842AF66AC8C18807E9"
00281 "8884C6A780805720718A3A4D1A7B094846C55B0808736199CF4EC3F66B713259CC715B22B9"
00282 "2AFB1599B7AD539B188E99B39F9A92987D0ED15C94", // PRN 06
00283 "91CEF241899B4DD47BF31F3BBEF20F5BF13D9A98D3F133AE61F3E4A87A299A6B115B96DA6B"
00284 "2811414D20A449EDA0E1A763E1EAFB78CE05C181DD0947CD50276A10D62543A0ECBA573AB5"
00285 "DC794AD7006A520B419533CE8519F4EE4194B58B2E36F9E9705266B6E304D1DBC6B7349104"
00286 "548637449E5C657B263CBD9577AD8DB7D5A5BB5DF43A869FF91BB8706D4E81A4F8243C214"
00287 "D988410447E5F2BCB426CD785A28F48563E1ECC974325487DC40E7304D3D7CADA5DCCA6E"
00288 "60F4457DA181A39801D35F20E1DFA05BC09E7699FA289A87EAB311BA8700AEC0F90950E5FF"
00289 "740DAE7EAE7FCAE8B0D9FB82ABD25BF46FAA635F70C22A76C52F9733D5BE64BA7AEB5D82"
00290 "628BD03D8A23C5BD3FD6DBA68082B7815044E24859865ED557C7DE8F66823CD4716E573BF"
00291 "F30067F0931D0AFED0146D55B0B2A3D6C57481D94F61279DC42D26DCA1CEE27DF24426E55A"
00292 "B4FFB87068AC8AA6F286151356CEB29D993B4FB26192319C373A1F0F58FF5949783E71671F"
00293 "B58A683DC1CFE166FE1AEFEC8A41C5FC06A31FAEFFFA3EB7C388E1BF99667A5D2697F71324"
00294 "09641AA812EE84EA8735BB46E5069A7BF1B2618B56ACB97C368AA7BDE1BE0F0BF286B6D08B"
00295 "42702F1EDEF408BC529B4158824F673B30E3D3247FA59DCB83D004459CC5EF87BD6C7D188F"
00296 "D91A2A7FEDB00824F67E818ED46A73323E9C7F9D5A83298F8C81702FB4523AFE5A89D4CC269"
00297 "BD226D04689F132F766770FF23C832EB5695713AB99B74035540031AD1D9D9047D72B1910B"
00298 "C320A65A9763A40C2FAFE217BA0C51FFE24E96DACE57C1318351038FBB81E76EA34853484"
00299 "E7ECEB672A4180A3D24B6C692747E9D306DD0AF275EC0581A30AC2E80EC73B04B1EF83160F"
00300 "D88DC43406CD715EAB1677F1422FCAB85D1D1EAC64F698B9ED69FC9F2B5519AF9153D5AB72"
```

00301 "2339434C780886B81DC2878B27177B4616450BC1DE99C49C965DC9257BC78ACA052E998493"  
00302 "A0CBF7B6AC86EDBC2627E43E7521A9DE7824DFA7DE19AEEA5892685D50EB10128B3D14F3C6"  
00303 "70F639AF6F2B26168A1E551A65781A418ED697E6AB3CF91EA2470995A401B08380CA58B94"  
00304 "39DDA7BDB7A92E91016567D57D57D5D0DC4D8E3C1490CA1BEC03FDE065C7B7F34DBC40BF70"  
00305 "4149A0573C55A0EE4C415533829EEE71BD8C392FD622572DE009044E937AEEEDF672886CF4A"  
00306 "36ECEF7B0F8BA9CD9EBDEAEC40BD74AD34CCD2C7DD07F7AA64EBF4DA9C8810C05F55619B80"  
00307 "8127CCA00D171BFEA3E1EC0895A22218DF1A0514FE5D16DC416D9FFEEA8EA3BC9FC650F678"  
00308 "EB731E70A6110E138E059371D5A10BEF6420A0C8BD45F83B67D310E732BDD0FE57D93A6934"  
00309 "317FD8E2449BA0F0C6E1B36821B62D1F80B9CB6A5B407B6755977C06BB2C3CEED0B14D4D73"  
00310 "A3554FAA54EF2C350C388B230B30148605035E4F481A769564E2814A160A67E6F0DFFA7814"  
00311 "D0C12772C0496737C52016A93C6753C82FA896BDF3BCC72105D6EA5B52B3810004DDEDC26"  
00312 "F61C90ACAF49BA7DB5016A4E485F06355441CA204F7D589F2A3159541F991682E6AEAC0C73"  
00313 "59B3282A04B2DC69B0AD6CF49C48343C8A769D3EFBE8D0814E72934F7CDE698061EC68FAA"  
00314 "B39016B9BDC5363D2B53363A548E6476647C10F55611469828564C189430D351CA01D2F10"  
00315 "29CA3CE1D28538E595D46C9C3EE9F7D3EB9C72A25F7B2138063A0F9186FF41CA9C579DA61"  
00316 "772FD60408668A877CCF6B65FC2B88935822A6DDABB97D19F7455A9725A8DF2B8E6B28D83A"  
00317 "CDACC66F02370575831490B8D8838D4E5EAF36ADF8", // PRN 07  
00318 "AA82DC5072AD5A21B7880DEA3E2691FCC22EFF3AC815A2576A7F480ADA6F8426EADB4A96A1"  
00319 "7EB949BC049A646D46926F0D69E0B3E1911D2CE652FF4D9CBFCFDC30065FF4F779DF896D38"  
00320 "587B297BE8E224EFB1DFD04BC2D2832B2955A250B7D0448AD9C0A76DE7C33A4E2B5DA1B3A"  
00321 "868852BDF04D848EA6A9495BE8D2501BB47F24E8DC4B254DC56BB5A4760DF62BDB229F3DF0E0"  
00322 "C06274A63A28D11F928AE2DEA72A9974E3C55CE261943F78B771CAE7B5B98642A01DBC0611"  
00323 "1B137774898F0F113A4DB23BC240BFF147568493FCFC0C3D767D3548A1C0EF7AA4B6B2DF566"  
00324 "C3E94BF9A183264F63E4FBA58B517FC46C4F0017F8D763A54D044C644D6326677A425F0F5A"  
00325 "2A7C8796B9C9F8CD0FC49A1F035BB6662C0CCC0E9508AF8D2B834B929D80096F9BFC922BBF"  
00326 "1BF6C6F101317EC08ECC1DB52404B30B153F43C266D3E64FC496CFBBD350668B2AB3B596B5E"  
00327 "0F0A00757ED04771F5A386AE86F5105C1718F7E93CF00F588C722509A03597B95512CFC70"  
00328 "5F08D3B4D5A3E23C56EB461A719A955AFD3EDC8329BC97E3EC6BAB24D97E16A724ED2D4C02"  
00329 "B2371FA3ACAF49175A441C3FDE53C2DDA440C18BD8B86F27805F944EAA7991D404CA74544E"  
00330 "2F3A669DCC4052B6770AAD9EE29A6EF7791C6DF93054D07A2B3CB2327B0EBE7D9836246399"  
00331 "61AC4C3CEC2A84DBB3A10A830DCAABA9600DED87FD5CBAF4D5F1DB357ED42374175257474B"  
00332 "AC200AC1B6A5B87D2FB2092458F51185C380B8BFF682C5BA1D2CBF8BC02E5D485FD811797A"  
00333 "ED167DE6ACA668927363D12EA405D75A2E9182B1FD30ABA700C5683611A24EC5DC453F523E"  
00334 "AEA44CE062621A98346328ED9E86CA8F7EA79EEC551F36836272127A45984EE165BEF8AED8"  
00335 "0D26DC45E34E1794F56C1265DD92B4078FD2AF2F13981C08FE27C55C3C9238BBDE193A956E"  
00336 "3F834445D949CAD84D3FF0FCD511C6C598735D5B307BDD7D437ADED3ABD6EC3171F373588"  
00337 "3C9F511A06F4C6C7D0B137DD0D57869B8FB1375FBFCF9C3D08CDBC12B8D01614AA3C965550"  
00338 "E5862AF49B04410D25BA4BE86FA6B0B9D9142461AD3CB4BD6D902EFA49632A78F1463619E1"  
00339 "309CC89452C2B453BF9F08A714D67ED90972C62C0468CE17006F9B60138D28DA6362670BA3"  
00340 "048CD8D099AF193619AA8384A1758FDF4A04CE56ABE464A66E913DB7BBBFF7EB2ADB47E63"  
00341 "47D7054CCA13000DE13F550C6263AA4135A2E16F1A2A58BEF962113C209CF58CE514FB51EC"  
00342 "162D05A4DF832697E544037CA18E62A267D81D539F879F50C654E74BA21B47FFA5C704ADF2"  
00343 "147EB3DD8617DE98B3ED4859B605310A777F3DD161F4038486F0872AF55FA3610EE0D68C1D"  
00344 "51E0F91EF7D9AF4ED01BFF53FDA16BA3197A518BAC0F82F8895A2BDD9FF4C3035379E870C4"  
00345 "9DA1AF18B3668792B6640E687BB71F13DF1650CAE1A2CC487C247D1D1356EE16DB8F973634"  
00346 "65614E9E63F36C853FB63C12963A8A5D98B52BE8DB31F0954B35C3C749A62C2B34A690803F"  
00347 "A66087015506225181C5D6FC101D1494C3A7961ACFB4D9B905323DAADABA1DCD2DDC1F9CBF"  
00348 "ED7D726E6022578CFDC1519925F8DBF9AC5FB4E4CA723BB264B5D106B6206574C46C1A49309"  
00349 "C22E1935904942C36148F764B59F10A5CBA0C9397F5798E200A606EFC75DEEA1FFB10A8539"  
00350 "8E5BFBBC6AA3619A0F611E591245D03B6ADC0C50440A1B1C236ADBA933B2BF84D2C60C807E"  
00351 "9F524359046CB2B813FFF8B9C824819C760D0BD636B10572C781B112CC0C603A260AB87986"  
00352 "B0280FB913A9E0BCDEE347F0C744BF8299A9099B8", // PRN 08  
00353 "F2A17D19EAC96F67ADACC0D83FDCF235F07832E8E78A44A53F9C4CD09C3F88DB2267019631"  
00354 "47EE0E1ADF0549E11BBA70053A105BD62C40DA810BA37BA72D999CC4FF9BA4D01FCFC706E0"  
00355 "D7819871D06C4935C10B85594CB7202F037B25F85141A580AE1CB3E0CA91AD73C6947D2492"  
00356 "B8FD2F889B1421E04FF0FB7866218A491D9EF6A35DCACC12CE098EC575C697F5F8D1BF9178"  
00357 "B3B36C999EADFFB5CB3A41DB672165CF73A7874FF68B3B901C4E9B8112EF4FF9B9308742ED"  
00358 "678DF30E2F7C40BF922099C1A95BE0C231E9F8332EBEEFD99C6DDEF25CA5EBB2985A968991"  
00359 "2616815AEA022BFBB87353B8B799D64280A6922CB09044ECAFA51037AC2FCB11516E9E286F"  
00360 "64B4ACE3E30A5519F24AB53F2E10E5A8920FFBAED77FD4C476E9AEC751763D68409ED9BF7F"  
00361 "435E53D401CAD787A50033BB8547C910D90DE100FDF6B904A1F529830E3A51919299FD4476"  
00362 "F02C3600AD5D4E3E154AEDCD1C99C1C5B31FE179860EC1EAC69EA0381CD4DCDE8716F94D"  
00363 "6510F10FD1915864BA968440CCA6D0D5AD8983C421079B033B56FC34481B9F72188F829F91"  
00364 "DD73F27BE8E0456B0F75DBEF40A3C67F274F6A50CDBFFF9798F8DBF1A6CE7158B577550E40"  
00365 "639D38861A3CC4E060C9E5BC1F0D3760FA9A89C2CF0A23505F4A642F5EB0A055B96843198C"  
00366 "E20133FDE022AFEE043191007276427DC82E877BDE27E20D65DFF8232E9E4DD786C23D4E8B"  
00367 "5E84637A22AE5562651BA45369947C159B641710C98494ACAD48B4AD6AD9828897437AFACE"  
00368 "E442DF2330669D5F2C6A9893E08507ADB104C62F6B4D568C5381B28D5162EF0FBECECF396E7"  
00369 "C622B54A7864F9B31CCB396A0DF82AB86D950B4657237FB76912286E6B783ECE3F798AF68E3"  
00370 "54C521C77735EDB97D580CC80877ED702231CE2F8B73262AE39EF94E84736949292E065515"  
00371 "D40A16BA113EB9437D30AEC44AD8C67E3AFCE9AB377753BDC481E8E79EB10F89A7C4F7AA24"  
00372 "E6FA48A36DD1CFC02CF3FF6DFD0CC1C4030C312532A70B60F678FF2B3586D77DDE74167CFF"  
00373 "A62C12F7D0086A6FA59B36CA025DEA1EA0BFA9861BEFEBEC6B2601DF225E29F36AC97EBDD"  
00374 "B21D7E8206E1EC42F3B9CB314E7AE646D54ABD53D090DE83B466FAF5384E0FAAA6E67F2FB"  
00375 "D6A72E00D9274C9B5C768DB4AA4281F25EB2DC6BBE0EA85CE1CD31B32BD1971D6AA20B6C68"  
00376 "D66426F09C0C1128A679AF06CAD490DCE3A2DB50181278BC40DF9E094D1D09B281CD1EF1C3"  
00377 "69E5927407E04918BD21315E12583D7D845933AEF186DAF8B6609C3CA9452857DD2112E862"  
00378 "27422491DC7821D7E41DF759BABE6ADD11A4C771CF16469F86262D169F221D97DAC9686B6D"  
00379 "A295688A42D2579C76CBF948C321F5158EB3FF5F6D4D6D68FA255C81F62A8DD605507D0C6C"  
00380 "9D28E6AD258B32DE4AB6FF8FCC4A8C237E270B9673C208E56AA6F4796867762399FC8549DB"  
00381 "D232B2CD0F9957A8A24FCEC6E8B3DCCBC67085C542A134F3E80A3CCE8D3DBEF0528432F2BA"  
00382 "B2F5DC89D6015BE7F24B2B1F378EE7522EF125A5645AE37AFE6ABDF84CCC3026507BAD9CD"  
00383 "C82BEAD02E8D60E0F85164734C8E51532FFE705C111CCB7D738BA8C0D29C5A0A50250D0BE53"  
00384 "353A6C407D4F7EDD62F271F099528CCB520E5B6B5695D3FBE628FF65D67CC0D1B766149B9C"  
00385 "89D3C770AA133FAE2BF3C9846B30C46584E9F93A8216627CFAD84C37AAD7986B8D69F83392"  
00386 "FF8F98C828F8B136633A09718D375327B98A0CF8B7736D229499A52D91D19CBB896A5BF413"  
00387 "1691682741797A25FFFE6AC60E30C79385DC778C90", // PRN 09

00388 "3D84AE0F7756BCB2518300DB110A07EC46869AC35F1B02650A36C3CA0350356E2808B84A0D"  
00389 "9A8A648F920B7D48AFBBF6CA9A37F673DE2B6FB15B189FBBF2444BDE8808118595AA995AE0"  
00390 "DF53F5146B011A0AF5B7FCB4B3D3EC074F2DF15CC9708EC50EBEEAE06F7A0235A50E2DF7F"  
00391 "6C023B53819AEA87318C4E67C4742FDEACF57DC539FBD5BA01F64CB45322212C707923303D"  
00392 "96E58EB95F7392292885391283DD9477218697698AD6D83AA96B06D68F58FC2B8FD9D61E75"  
00393 "337C6815BBFA8B76833FE1856B714ACCB6FB1B5E957DD385658AC760AC5811E3BC079DDF2C"  
00394 "E30BDC21AB8D2C7B56D2F6B384FECDE4EB0206A6C00282950C1B72B3E464570364A919812E"  
00395 "86BBFD0E1CB2DC98D6FF42617355F2131F72A11F50618CF21A4C4961083FC6BFAB63F54467"  
00396 "7428BB9F10FE341B947C41AAC1B367E338CC2AA08D6CAD1067F02654794571FA4AA11E211F"  
00397 "5E75D194197BE4AF6543009C40EA86F74AEADAAB134CA9E35D79F7CD062B53BE2AD86FA7C2"  
00398 "E30D9222A49A25044153420AEB6C8FFC4E0D4C514B8D9F5DD922D8EA387003409539D0B40"  
00399 "210B05C9209B008D3E47E8F84C376F46001FFE463F21A92E34D3345751E8E94205EA06FAC1"  
00400 "6292B8C6D4FA028867AD4C4E8DE0D51BB7AED59547707CFBC2CF3A9235C29EE3A3873B96FB"  
00401 "75DD50E758CB2C7196BDC09336A97756DAB994C46D5BBB87F9D80008D09C603A68022BADB1"  
00402 "D2F6B98B6B76CB3C0EDA07A23065B93DAD67BC59C9BDEF7BDCDEF39D9AC04C9CE8F6DB4A"  
00403 "925F77BBF8150AD478E3C6D7D32512ED75BFBCB82C66D505CD8BC5A5679014B355BDC66F36"  
00404 "5BE9AA1F5B76C0CFC0B51E53F86BB4763F0302B448E84B9792F34442743E2D979C5133514D"  
00405 "4BB1DE5E8AF938C4CF158336D04B6723F9422161805D5EB059E55C86D0C461399073A705FD"  
00406 "6E919D1760B2D686D65665B486DC0AAD8F7B3ADA9BD681D7FAE4185350BAA543E9F8671F1"  
00407 "B4BAEB6D18674516033B2E2E4E427521C290A6E44E1EAF9974BBB360FEA3535CF46D68193C"  
00408 "5E208AE56B2685CEDF0FA44FD67580BFD23BF77C61B95B533107A25684FDBAA4C1285AB568C"  
00409 "4694ADE0E31A4FD70AE5702C0A7710678763B035C5C6B4B9214D507313F5ED85C444B98278"  
00410 "B40446C861057AB98B97638DE4739088FC28A577CBD52A3910885276A2DC8DA99EC74D30DF"  
00411 "707BFC60663C869DA4C47BBF4D6A4789425C5C894EC2B512C3296D0070DE4942113FFC0B13"  
00412 "D03515F2B4EC07A0E3BB7AAF08DE24779F9EEBE26CE7C316EAF47396E538672BB646E1B201"  
00413 "6356A407B8828DC25D8F8A8296B4F855290131734C94CB92CBA5689574CE95B60988C42558"  
00414 "DAD978736D67BDDAA92FEF21282683EBF6D59763384781EB4D3431DF5BB909ADACB397D36"  
00415 "D55D2ADE537D7B1B942F3A815917A388B12285689B40C3B45C04A591672A2B957F924F1CF"  
00416 "6049B96A38304B8641DF6A079614C9F9984FD9C9F0A82ED6E440164AC484252B45147929A"  
00417 "58B069E63663CEDF0FA44FD67580BFD23BF77C61B95B533107A25684FDBAA4C1285AB568C"  
00418 "9FAB387C6FDDF5EBDF5AF1CE094C59630E19F621CDED14E3847BDE9B0E0EC64C159F23C779"  
00419 "C4B80163554900393CD7F1B885F92F183839CDAD7801DC7FCFD08B7E08D23BE21BC80F8655"  
00420 "A2F3BD0C483B2C1D23D447682CCD9CE098AECFD7BA7C48F5BDEAC8397220A7852077A8D793"  
00421 "3FD353C25E37DAA473E996A5920DED80C26DE4FC3AFBD6ACEA16DD3208C9384021D9B1FC1"  
00422 "C88431B59276D48D8BF5A0A37D84B8C579EE0D9BC", // PRN 10  
00423 "446D385D09A3464D1ABDC6D6CCB015280BBE7DCF00078634371D7F33722DD720E8BDB7B8C"  
00424 "D97FB9505A118B469762ACA10C4E135ED782FC905EF30F9FBA82F11AE679431F270351F70F"  
00425 "0508BA998AD9A71D2BE05F55348B275A0E537C63699C96DD8AB8E0645CA2606E35805F66CE"  
00426 "586C840FA65E2EF036697A18DA49C36CB22404497577107924DC02FD87072490D7EC8D5B37"  
00427 "6D0C3370F9753E0E7C781952C03D844490370D6853FCA49E12139805C24FE9C9914E3CF914"  
00428 "64D332C934570DCD5D1C3C609C6984E28C9BFCBA3669B75452478DA4099CFE0F6311FF782D"  
00429 "1D0123A120F657CB536A3810984871B69D25913C05117A7E51D199A56387C2777E349748B"  
00430 "9C2C40DECC03DE7235F7237CF43FD421039743CA72361DFFE2491953A3C3F0091FE3EE064B"  
00431 "838BD73E9E6EDF8FD90235CDB374892BFC42759792E5636D031795C0DCC70D7DB86C161FE7"  
00432 "059CE17B6926F5D024D6B1B9DD5608839DF85C7802F565525B633BDAD0803CE14649095C"  
00433 "0A7E870F66888C072C410E0E0854238BA7FB259901CAFE3570D398B621518193BBDD75258B"  
00434 "CB53AF2789EC66DBE2311DACDC1F9CDB6BDB4BCDCD15F9C2E77E0D2736CD191D2B83389A71"  
00435 "B1804A0FB5F2ED08C38343CA48EC6D1830BA3466E53FC51C4EF05F4D10711A78AB215650462"  
00436 "E9A26D9301EFDF15A3CD5EEE14DDAFEFA45171BA3A757CD6FAA5887A5D5A98045BEC7FEB33"  
00437 "1D01A997282910B6B946CADF0AEA57973520EF0561B233BCF798805B3B048751EA086FA71B"  
00438 "CAAB15F8BEB837CD2974EB3E54D3B6D3F99A51F88313FB64218610CD1831E2E6E18EADCFB"  
00439 "E72BE5818BA4EAF3780BAA7D8C4A55DAFC10ACA3EE7AB7D1337506C485C83EB0C41748C668"  
00440 "705B4D41D926221D33C31BE7D56B42BAC7E9240FE4C2287B5B41FE88F9B9E19970D0E4569D"  
00441 "BFF167EC42BAA256667EB055B69D15938D23DABFFC86A917490B4942A220038CDADBE24C"  
00442 "335DEB895F3472F0D22778BC79B169D1EDA9FD18B31DA66F0246359DB3EBA36269EC5F92E5"  
00443 "A1A89560C7DEEFA121A7035EC2501C553854724C5176AE60F87004F6FAEA907B91D1C58837"  
00444 "C9FB265D9C32F66257BD78169B1A48ADEBA72B5699D4C2D14F67BB347EC8A96B04278AD9F"  
00445 "97A9B40DBB56E7593CF4B3C905108F10968F0AE6A2E7F1D214FC7B45637A256ADA5EEB70E0"  
00446 "3C02DA2ABF86C36E581A31032DFFFEF4343C3E59EC7A2A715E9C199DAFD3BF8C9B431CD016D"  
00447 "CD33E590A20EC8B657C08E5C0CBB4F159A07206504780715A252D1F4F5DC65ACCC43E3CD"  
00448 "C255123A77792685C9B68AF281863AC89ADB5F778733E95050ACF781FFA386B3EB3CD633D6"  
00449 "E8E6D4A42EA85F005AA80A4597D40738A8E8823FE3C84B10CEF044E9651E45D1ECD5F95C37"  
00450 "AACAS8D2BE4265529DFE6CFFFBDAED252254B732EDC93B4D610C50DC81B576C5A5C5BCAC"  
00451 "0BAFD7BDD7771EF1A19C29D7A82E448AFA07E7FAA8651C11D1BD550DDC69F4C9C6A4443503"  
00452 "480B47B512155E5141378A94CD277D4BE11B2A383B1539AC9731079624E81D3A3404F5325"  
00453 "F5D2EFBC0929F510529C84B1BC3E7C08F661640E55AC49452C00256BEB49B7A52DA23EFFD8"  
00454 "5B8C303C6EEDD2202768ECABBFEE87FBF1DAA992C8439602B1645C462D3297DDA1AD94161"  
00455 "7B508E481A28D1540212CB281B5388A3691A26AD7D3CEF52FEEE9F59F477EC98AFCFE38034"  
00456 "B762B9D4B5D73BB312758C4238EFF94BD16D9C39655CF07263CD6DBA96D1C0EA425610C1F"  
00457 "C81061FAAC5C0F8E6AA9591EDE2FAA1AAB8F76EE18", // PRN 11  
00458 "C514F282E4B5E02A90A2AE126DA52A9FF5274735ECAFA74CB36D607EF299E0227E94B6CC2"  
00459 "B78B1F71D6B5FB8BB4C78C09B3A6E3ED6CD590AD526F74CBF869085F2BD3A9DF66D7DA9DD0"  
00460 "6FC6748D8FBB78D98A4D860B362BEC6295F34A6A16A58BB21C9632F8B3A64370B7A7547E32"  
00461 "9B39BCC82D17B82F6E4ADB514447F62AA800C97B748D73B0168ACF2FA898C262820F898171"  
00462 "3D00CEB8453E39E452F5EF41452A9C8CD1F1260000C4CA9AF4E6E629B13DFF203187C0DED"  
00463 "A08315AF24181015D2E68D6AA4E7E938FC4FA9A8B80CBF49E3A9F27953A05FC7609146EE39"  
00464 "E09FE485015168636D1F452FF311D4D8749C31AC8420865A87A707E540D7FDF0D25F63CF73"  
00465 "AC906E048281BFEEA46263B4350BAFA8B5A27E0CFB23DE145622BC2FE34DC4F4E1B4DCBF60EC"  
00466 "D5426C27B3559B73A59CB1B71587C685542F5E6DF9FE3C5E75A8E9600E89217343D29C67A7"  
00467 "AE80F9A8589D82AAE7B2A1B8D6546D3CE2482DABCE7FDD3BEB2F1009297A77503E7155ABD"  
00468 "4692F1244A9AB32628947A681388AD75A750AB92C81ADFBCE0DAF166326FADAE6B7907047"  
00469 "3D67A8FCEBA7F21BBA932168CEC2A754247CA324081900636D87D001A605C4E5C31AEA44A"  
00470 "8DB9693617566726C7120AD15D96D917F5ADD0D86A5A9CDD7527644EED58A9BED014D9F4C0"  
00471 "F62FCAD8D7D5DA2899D57C5510EF15B7482CB3E087493959AA12D73D55E194E48CD128FBA2"  
00472 "788829AB4A966C325D60227074EA55BEB601128D5FF42371E74972B802D3117E1F2AF201817"  
00473 "7D4E99490396DC365D3B52582AF2085EE5919DA7E84A2F3A200F4F6E76F11CC95F87BD2EE"  
00474 "5935CDA0D52D9EE6FBA6D59E80427EEF4B80372091C65E059B566C38A6B3664974396BD038"



00475 "3A35EED0494C2BDDC167725C4160AFA58D408978D5A345EB03A55127C6BFB08FC0B4EF7037"  
00476 "482EA724D9D121E26F2249C0C7A927946A846A67FB7C565E86C8C59A4295ECC4927A8FC88"  
00477 "873DDEAEF6F8B629BAA1819807F38AFAC501EC9F206346581174A037B1D7EA56628C660E17"  
00478 "BA0A97F2E189299A7844FEDC562BE678B0606CE18493F717559365F66201AA598C68F99A8F"  
00479 "28AFD74AAB0E3619BC342AE9F36D3811EEB52C8DF6544C96B679CF434B9C64FCA0F43DA73B"  
00480 "9BC8FEBD609678F5E691AA706E3586176ACDC25652BCB68D98417D3FEE12B23724EBDF496D"  
00481 "6AC3B35EF706EAC11BB0090066F3C421C3E298B00821C90DDADE472331B9B2CA1F1E53E6D6"  
00482 "8BFB059EE0811EB7E7FDBEF394E3E35E59BCC4B63FA8E09851843F28CD941F9A97E5B72C8D"  
00483 "CB9D32F37B79DFFDD1B9D7C22C10E6F65DF07DBC7734430B811BCDD450A10F491B1E0DA3746"  
00484 "868622E457774D0BCB1B29F9C1CB161DEFAE78354B7F718D18F07ED6672665D35562D6F05A"  
00485 "2B3666B667A7C57D2617711F27DEC15D0A4079519D0ACC6B3591EBF0A61A12C42A6820CBA7"  
00486 "55C07FC531610C4192A0BE9BF241BBBC5ADF758FAA4DE47C6B766AFAD728BB1D2C672EE0969"  
00487 "303E42AF789FB295E766E90319D98F0618C1B4B8BD31CD2C5EBABEADA30C6E43D03F380F4C"  
00488 "2B9C3AEA20BC89B852BFC0D42AD3C9440AB395E9E251DA758D317E39792973083B20B79BA3"  
00489 "6A14057E1DFFADCF903B10F8D3D4D35222E4584D520D6FF675D3EBE9FB43ED1B79252648A5"  
00490 "E0C2552E388AA3D1168790A09854F1151A94DB4E5AAF1C941FA61B196D869B16051B75F56A"  
00491 "C1BC769C6E7A7EC173264A2E56BDE4BB9952A422855D65AF1C209D561F8210FFD7C2FD6A54"  
00492 "0BE788EE3D2822D49DBD2C6B3329960A4835F07A34", // PRN 12  
00493 "0C0184CC1854C2013817663EBD9E23F3B78DB8009E761F1B38C2DA2036F26819D2DC68AC38"  
00494 "C18FC15EB00BDBF5A11316C7B6075BF9DEAE9B0A651527DB3E3B3039BA01398B3923A2C40D"  
00495 "C4706A0BC186D2F8FFB124EDE35E8C427FB6FF05A32CECB81E275D1AE370AFDFB6F731E27F"  
00496 "4B22B877B1990FE479239526C655796BBB8545051527ED0E532E0440048912C64D0913227C"  
00497 "2F6657B8B0FD0A07E49B7E3DBA1D0E23C9D7F43280A324DF16BEE2A87B87C21692FEACBC3"  
00498 "348C9186DC6C0CB7C1DBEC60AFC4F1D76FD3D45A24ACFAEB8E7C95145D07427CD6700301"  
00499 "9AB57DC233D566BB9A20AB82ED4130D9B4C0FAA6F2EFEDC8977498315FED436F67F3B17087"  
00500 "7040265EBA49415B1B2904C8277A11598432B973AA63F628456AF2320D3478FE669E2835C0"  
00501 "0D13EEC7B6ACB6BB94F1677C9A05B2D0CB87ED3E529AE1D2E3354B1E30C6634E10AB3BB0F3"  
00502 "9815049DD5321B11B80CF1D1F1426A413D0D11FAE969F3BA266F4F7B60684224A627938D0"  
00503 "8D46FFFA2E1D684416405A02A2AB2E260ADB23498AF716EE58B03C8CC442DE0EA201C4DE39"  
00504 "FA7038AE4E0E1544BB08F149F57F03FA910E09C27303038EDCA21A618A54DA87CA747763EE"  
00505 "6D4D1DD4912AB7E3BBFC999939F82563B944CFC5F4460D41A47573DA92CBB28439BCDDBEBF"  
00506 "F16491C404B356E5889D33A034680F8648F2CBB1888FEAFC410FD5A3C8ED85B90EC875D7D4"  
00507 "466904F5D4A4048D8DAD848313232B80FEBAA09D4D86F70045C0C2499A93610DD088A6263D"  
00508 "E5AD71067B9B3C82E20658AE98677889BDFEC65C2574E4AB7C9059F1C13CBEC17EC630F9E5"  
00509 "B891DCDE1E81552738D4C8612F3BEED570EB39B24EC94A153465D1326765D9B8F23D7E4AE3"  
00510 "F2782F6F9738C189D0D20C4DFCCCD37D9A510250F0941423CB9055C7BFED84520AB119FF92"  
00511 "1CF81CBC6967E9CE1D759E5F4481ACCC5881F9328093B63ACB7D04404D8C0FAE8747D39A27"  
00512 "0399000CB690CDE17A46E81DC5CCED183BB863FD7A535CCBC592C25B059496C45DAD179B83"  
00513 "EE02543B5EF93213B10E2BDD480F7C7D007CA2988942F65E46B886EBCE0F05498C655048C4B"  
00514 "32A081A059A28DCFD484AAA586F0291F322B25D186FCB5ED2DE7AF7EBE58CD8314CF846AA3"  
00515 "9C67E2F284162524A14BD2765642B31F31EF4704548C13D5F5A132E983203EEB91795BCA93"  
00516 "4BA7EB26DAA5450745333446D2B8BADE99788476A3417DB7FE9E53F4906CAECBBA98465796"  
00517 "9B4592F9213612AE6812C215D5C6E2B8C9BB76B9CCC937BA4D4C6368CD4C491BDF41B242B"  
00518 "D8F9C9DE2D35907F08AB80BE71EF90F84BDFBC6777A020DC97BCA1E963050FF991A4F0E1E4"  
00519 "68813AE1CCF05B819012377334AD6B5A41A87A7D7C4F0AAD241FF7D3AFFFF40D7E6D5E62F9"  
00520 "884863E23C3A8896260A6E9E3BCCE001D487E3E75DF4D35959633BE719FE592C4659970C0B"  
00521 "675969CDA4EE3B05237C02CF2E36FFDB7CAE37F0BE993FAC46DA6F56B15FA74EB4511AEC2"  
00522 "C861C2A95253259268A9BB95468FB02867394AA8D83EEB6464A65CD15DAB6A696FABAC4661F"  
00523 "439F377560F3D2F3B668A7648BA4017DB5083BDD0D76B0DE99FBC06D7938D7D2F0566D3DD"  
00524 "036CF6A5E336359F0896B9B1E3B685D963A0EA70CAC175574C5AD41E20EA1332CF103BDE6B"  
00525 "1C3B3F117C21E8FE7450866C5036C244856B5FAC176671270EF7225BE7CC36D24498C9E53"  
00526 "06C2870FA7839DB1D7646A62AE0623EEF3199E8DBE32020B71F23FEF2F2BF3C9E0EBF0FDC6"  
00527 "29D97837B97D76C99398EAB044AFEED5BFBC35FD4", // PRN 13  
00528 "8767E0FD01FC0DFEFA6640DA0B47AB5B2B2CFD25C8943FDAE6AF46C3898AA1FD2E2DD0FE19"  
00529 "D1A9BD0353A16ED03C448B7005D56D108FF2035E948E3C4FB8F0F928A5797D491389AC431F"  
00530 "271DD1C8FDDA2B1FE623D410190FA81F74E042C0BE9F97F99120DDF442D5A4ECD1EC9789DB"  
00531 "DFA8C8E577E0AC5A1BA7DD6D5A474CAC3C2C04E71A14DD2244A30D64B17E4C9B081EA8AF01"  
00532 "75D7D0A6E039B6A289EFEAA20763B1F106667745EE2795CC588721BFBCA6D659D81192A02"  
00533 "FF2CE11CC878DCDE73CAC9B92A972CC9340ABA7A03C3384ECD28301103F347D35FCC9B12B8"  
00534 "5FFF170894F7D1E4F34D95B56A23C486CD9E7520EF8CC0BD928EACFCDC5D230D77EB9B19EE"  
00535 "274113E1F8FA79FCA93EF54B8690859BA6688FD29444E03E2F14033F076FF2F237111290"  
00536 "99051C59E0126A907B57905A2F1024D98736C3F5D1AA11C25F32319546F91DCFD4940FDC7"  
00537 "C4428CE78A998694E6B19AA30FF470AE4582666D350ACB02CFECD0E9EC39D1607912B89247"  
00538 "9EF94CC67797F443EE6BA4008F46C0E61C3819C5F82DE7A7802E62E7050C8AC506186878F0"  
00539 "8ADCA8A76743CD0D15606AB7BB08511D2F8DB46EAB9B46AF359135030A9EB46A6137A941"  
00540 "9D97C7069147BF5065712B1F03BB8B328CBD1E583F59B662B64C0F06ECEC8B71AA2213CE29"  
00541 "F8968E0F3093AD04E739F01DF51C644FD4CB65AEFC7073018DDAA81185627B613242260CA7"  
00542 "A9270FDBAC3EBEF62865D0341B5392DD5FEA4827D60D5AD1C9AF0739CF5BFE0F658E72C436"  
00543 "BFABEFC403B87AD4B64F3143DBAE1D2CC25130E65D72ADECA1A8FDE09EDFD723072D53879"  
00544 "2F50A067BA78D7E4749F0344AB1AC1CC0E8EFA1A42515877C4539EDE35556D444516474628"  
00545 "BE9D8B29DE428BEFA12EF99C985F1EED4F937BA4794776C63033C03EE356991607473835A5"  
00546 "0718E473D036F3C522A9DE2E49FEC2E686F17A5023FB7E82E94B26A4AC6236E0B656A938BD"  
00547 "7A8575D734F53CA419836110DF78197D7DE4D45B6589CC31F5057759B8402F43ED9A423A7D"  
00548 "69DB0EB9357B1E51207161E08FFEC8FE286722159C570ED4AF63EB554BBA582404C0E978C"  
00549 "EAEC1CC1404CD15BA22EE6D8C98C4278C1714695927C2389D74BBAC72C2516AF1E5D0C5206"  
00550 "6A656170091B68A7E4C8399ADE63605E9DA1727C189C541A404B7335DA27843A32E470FBAD"  
00551 "ED8EB5EFCFD30BF15AAF37D7CFE4B545DD09DE3E7C8D0E77427B3C0764608912616E5DA2A3"  
00552 "B885688F25E83DC8D632BC00015C2C8E6923A3D9E7C7344CDE87D6F0DEECA2006F644F7BDF"  
00553 "A0A524759B2B16313501A1CB8D4D762D3E9D9C8D32154F69F0DBABEBD26AEDA947485FE0B4"  
00554 "0D1D7731D7E4E41D1EA5FC645DE31CD850273B6EF9C374CC8E8149960B349BA16FBDFAADF"  
00555 "12E86EF05AA38B6B17EEF7B55D27D7CE27FEC916DF99F0715F1DDFB93C7E1130EDFA44663C"  
00556 "F19F4008D5CACA48262C1E8209D0C353ACD72A542B93EF18EC7E15FD4C9B1633161D94B285"  
00557 "FE8FCAC30FBB1E140E9561A35715AF982370AB4DDF25BC30850E5F059297289C0045BD3B3"  
00558 "912843A9B0BB88CB7E961639849B8A0A75E3E903182F0CC6C7ECAD4A38081BEB18DA72EF9F"  
00559 "9439DA7DB2B4D59AF6CB71808E604E56E1D4BFEF627226FC2887588BD486D0C8052068D12"  
00560 "BBECB27E9DCDD73DBC93D33EE66AB63936C2273525A517280FB4BC33A724D20BB7D988B93"  
00561 "8018EAC4738D03D583A71C16671B87CD60BD6FDD101BF2AD07492A794FB8696E84CF24B970"

00562 "7D7706F0AEED606C3A7FB5B1DF7FCF86E65F6DEAA4", // PRN 14  
00563 "CB8EFFE09F09C622B34574E74D4B061C284F96FB7937481FA7C204F449BE9B2DB0957E428"  
00564 "D177C525C67BD75D75989EE98D97C4D0DEB5E72E2B0DD62E66DEBEF0D22D2EA97D42A3374"  
00565 "B9B659804F1EEB01D7743DD9AEADDEF356C7C4FE94077411DE58DFA5035271015AF65FD1D7"  
00566 "8C0F17632C10EA02C537706A7C14E3A1D7124FCC11F579333A68A1AF70BD3E86F7B1D13A3B"  
00567 "4EBDCDE34F36A28FBD0C1E758D380C2A7C17A487191A46C04C23F7906418B69E2527F329358895"  
00568 "E83D6A34ED4F9C06BBFFCD5953C51814FE0F4CF5A697EEF5559886D92F811004C04FB399E0"  
00569 "6DFA845DEF183160BEF8546F6E8995E403FE0E464008646330CEC4B4149DC464F229A47E0C"  
00570 "D9F89C83A28FBD0C1E758D380C2A7C17A487191A46C04C23F7906418B69E2527F329358895"  
00571 "2919D220EDC36BD9687C7347AEFEFC8CD24E91299900AE4ACF018FA2829E215CB13CDDCECC9"  
00572 "18DC85BBFA317FEF7D2FFA9E3A30D9D1F1B9DFA0AD0D641E800850B43D7E0B187F2ACFA74F"  
00573 "C5DBC99471A7511B838BEDD6E2744C3AA9AA9D46060C83FFF803C148C35209F840AC23D60F"  
00574 "D046B461CC2D776AF51612DD166E1450B8A998052EBB25D1C8F426391A8571C88C2012A894"  
00575 "0CBB04DB3223CCB39021D20142FB5039A6ACB250994AD7A99B1A43494EFFBD8D3692F213B4"  
00576 "C66CE9AF3477B6061398CB19C099F27F48F4F785E9E741988412F070F39727FB8B74071E0"  
00577 "07210D939B0EC5D6148A5931D38A030374E5C0B575AE27D49DF49A77245BE310E49262C56A"  
00578 "616C757A80C5989F27962303825EAA3AA014A4C9C9E5B3CF93F1A031F82D6A0C877277EC41"  
00579 "356D1BE2154F8027940A1CB2848A6A23B905BDE9A907F7C20E333586389014126765CBA8"  
00580 "D5C8951CE09DF085B5F18A24E1E5D638651687CD2A2F75CBAF738F65A8B24E7C67E64FFD9C"  
00581 "B8A34A473F11996E3663607A49EF663C4D87A0F65D92067132B5EF3CFCA9B49FF66CB6E36"  
00582 "3BC1C7EAAAC392605200A5EE704D1E754CBD1C9CDAB75355BCA41DC129722F8552152200D6"  
00583 "F6248609DE9EDC5603562F5C82B5414BA305CEFA60020E1CD93FC756241FE7A20CB4B550D9"  
00584 "80A7266E7B23B6FFCFFC4461F925A14EF6584B02005E1D795ECFEF93312B3369C52A8A2ABF"  
00585 "99AB5255E809C05AC722F37A5E95AE46B226C7319411750562BBD0296F0CD1326055577BE"  
00586 "AAF270E75F4433D5312038854C30AB23BD51F1246EC46599FEC1B628137A7A702606112CF"  
00587 "61CEA7C3977BCA6701C8E5F63132C15EFF01CA1EDFA5CCA8FC0E3FAD392F21972F857B53F6"  
00588 "2CAF3BBD75D556E6857067015E11C5E217F5D1C1AB1D603571F9EDDE04499E23D39DCAAEF1"  
00589 "6263A5EBAB9BEAD8F61B0ED942EF261E0ACD309A903E73AC4E4569A65FE6490F6DD6842812"  
00590 "DF61CD858D9AB54D3456C89CFEDF2621426E3F9D13FB94177104F6EB4E30B018C01F78CB4F"  
00591 "A9C89F453B87FA7D4716B96B0BD2DC532D4AA7D8FB9D264B095BCE39BA017E8BB90D600DD2"  
00592 "C58647B9D95F63783B4B90BDB83A175E5201784AE000A493A387E16F5F0C5CAEBF62A360A6"  
00593 "A927B3FE4CBFD310A3163BFFF1836FBF9770583FA4F88E5A6E7BE5796145890A41FAE8BD18"  
00594 "39110EC0C325012E19C69C3868C7507D04458C074DB3440A0E018A23CDD32ED35E7A28DC8"  
00595 "AA8BD63C8CD0A9F4876B56C04D5D3D5CE52A0601ABE2AD3CF3106CFEA6D761335F420EE3D2"  
00596 "77DA358DF4F6813D543424653B256E325346F677EB7F69B2FAF9536DE80E8EB018D9784845"  
00597 "09804D00783708D48C4F7100E0AF891BDFC75352C", // PRN 15  
00598 "93EBCCD22C36C3BDC8B579B31AD5F5C9B2B08218BCCD1B29829A365DF4A8D5259753381B67"  
00599 "86877A5250DCE6A2B182734E363B39560D3DF44E88CD6394324765087DD1A77B3D8121CBA6"  
00600 "1F665F9B719E75B740BD930D6FEF316325B8B3A2FDD93AC488714AD9AB525248EF60B2649B"  
00601 "70DDE6849BED793264062B38EE74270CC190C6A86AC4E106781BB3D09F0A6D7AA4DDBB0116"  
00602 "D3452CE041BFF4247819FD8D78304088355D8DE741B1A6461C74A4832ACF4D4004BFB22F6C"  
00603 "9BB397D7CADC6F0C5AB74919CA346F9A54DBDA80F8E1303A1D43239C5DE043C2AC21E182CF"  
00604 "49BFB9BCFC109BC884E41FE2D8669915701992200BCB938A745FB0994E6AD01187C4A24212"  
00605 "DA421A178E42216287DB1F6A25E8370E4209E09057E16253BF477A061399AD1B3FE985D6A1"  
00606 "0D8B05D94B67E878649862D68F285EC0BF06B7006589A590AE7B116FEFEC871010993F47"  
00607 "68197473BF22AD54B2A8FDB0BBF2B181A2A1CB9FF0D0989F2F68B339A8D8B27FE1781F3DD6"  
00608 "F7CB972D73E3DD702FCD0CC3CBDF66A6237460E3B2F45DE273D2BCADB96C118DCC19F7C1E7"  
00609 "03BCBAA04DA5370A5CB3B78112B8E95B63328B43C51557AD84B3564749C2802B52EB334583"  
00610 "94747C176E119E1774BC905119312963401C1DDE02964DC9F1D836069CFBB9B33791C908AD"  
00611 "CC717B08FE3C0AE98C0D5DC0C22876595AF95A02AB33BCE3502A92E532E72D0E3455A3D7EE"  
00612 "796B3CD561DB902DB8C0057F2D72655E95DF55DA562F632A4F9A2CA8B1D2DB8F5745A713"  
00613 "18C37043D3EB08F9950D06C1301F9E9FAF4EF560B238BEC483DF0E017E3CB27A251156221F"  
00614 "183F3E9B0581838D04BEB5DFD9EFC26B979BCAC7DF631F1ECCB777E2617EE7386C8BE9834D"  
00615 "2B96D80074BBEFC8BBD203299AFAE8E6F90273B92182CE7EDCA9154E0A33AF9981B8CEC9"  
00616 "4E6A7C78C4716B47790EAF771C3DFDE276E4E617CC0DA50F05099B9BF587127D1D18D5F89"  
00617 "2E0090C8D033F21E7382738F1EAACFBA493D776821DBDFCCBC24B1D5F877B157AE6BF6D307"  
00618 "AE9BCA04DA5370A5CB3B78112B8E95B63328B43C51557AD84B3564749C2802B52EB334583"  
00619 "BF166EACA8D16A532BCF69526FBA8B61F71D2C5F38958EB11F9ACC438CCE14E88A11DC431"  
00620 "CE4E20B99DF0ABDD2B5AF358260A64012FA4A08C6840D365AB95167B54FD479E8527B5FF66"  
00621 "2161F75FBA7CF790BAA51613DE7C52A2A94E4AB78E051F32D10233161A4EBD8682389D091"  
00622 "B5C4509171650ABA574F1FE954698268450E11D2FB52C1AB3E4DF426D26DC6D5319FB762CC"  
00623 "CDD552F4B1BF4E82AD1F78606C644185AE87827F022C9508805472136DC48E71D1C4594441"  
00624 "3485EC1CC72BC092D6F9990148E12C2A3FFB3D8C9F91C18CC9086F176733A83F2CC9BAFFB"  
00625 "D810A99B66DC3B399891B2045BA402749B55E7E363F5D062427A9BFA8B599367F1D07A38DA"  
00626 "16074F75A4E095A528C26DE4083B044055F73D207A6571196263D8BF5A53B7679791BDE49E"  
00627 "869F0F58431829396A34415CDF4CB986281840BC48857361281CB4BAD8FF5B86ADC76FD3A"  
00628 "812D7641C5FC88A21F137A6980B42FD3858E2A057BD4D3FF57A5ED3F5C41C71C70036E1762"  
00629 "C0079DEDE282175072A073A90AB83C9DB47AC07E6BF99DB04BF276FD876F4E4816B029433"  
00630 "3BF3403DAC2A2077BCAC3889B55E76D62F58DE7D1C49816D03DA2D492D1F3CDB2B6B186439C"  
00631 "BBFAAB8645E9745F404538BE8FC4C698765A3C67D01CB54D404D2A2E78098FC334FOA375EB"  
00632 "41C3031F75F9A773612445AD3D98BD09C949616D4", // PRN 16  
00633 "5D55CEF58C74B4C681E813564F1903FDB56BE053280D438BBE5E2480CF279CE47FBAFD179E"  
00634 "270D838F764C27F496AEA9C0800056F2E0456EAA7E59CB20432B65875455975AB37186D6B"  
00635 "06FC7D0BEE743CB974DF92B19F4D2EEE82F36598F8E400DCAACA4EEB110ABE352D3A0F58E"  
00636 "FF8D51DBDCE5C23E84D1166FFEE5428541ACA407B9578E4295C3F79F9376384B630D25F112"  
00637 "71C6234B9AA9C1B4C8015AC2C73CF768B9EC30C61BBFEAE1426F72FFBC995A834C9D4FB835"  
00638 "0C301133F123313370C33FCFD711D1847CFA84E782F4B39762A02F9476D5F45237A29E3E26"  
00639 "9D16DAD60033E0907D5431C1CE36D37BD388024A020B612E1DE42492A3759AB82AB393237F"  
00640 "4A1AFE8B217D2733916B7B610342C0D182C17BFB4556B071D82E6C6E554F2127654C15671"  
00641 "FA1A843347BDFF9498C7B5E7636318C3EDBAF7BC4BCE123698A05538D5D42F01970A1A7"  
00642 "5F0A46E61520911D86A39D892700AA23A033EDC836EBC17CD2EA274252F53A4BFFD5CAD88F"  
00643 "8066744C7B7364A0DD0B5FFD96787F7D18357DF728FB66B212DD4DACAF7218292216808E7"  
00644 "B81E35144B4947E436A301C4181CE53A298250835AB98DC89D0A3D6240C57C699A5BC1EC4"  
00645 "CBF8C111149CF6802942256520BC7FA22968556E2B390E63C9DB7D0AB79939924ABF068BC3"  
00646 "D44264D607F088A38FA67F8E25933091BD3EB0A3590F3E62E63CC6C66AF66128F7C3F97B76"  
00647 "E62F1AB8274D8C147FB2C8BE207FBAD7E5CD355230EFD48DC9FDA4FD1C970781674E901D"  
00648 "35D7C651EEFEE2356F825ACD348FF0FF590E8CBAF0332F77A063581E6F19F53CDF1BFE9525"

00649 "85E93EA13A83BBAD06070850EC824397A9FF13227F4C60D65B67AF810EF1BF5D5EA95A5512"  
00650 "F95AECAEFA25BFBB24CBE6785E91A4874E81E465AB91B1F2E990F37AB145645F2681B47592"  
00651 "990CA78264DF8902620E796D4DA9A260D1BA67DF9BB0292561FACCB2478AD7F7D7CAB3DF17"  
00652 "9897596604F2B0F1E3F10CF842C311F49C6B5EFD5D73DFEFF59127384D3B4D855F8C412E94"  
00653 "9E3A941782C186413946DF8AE33C9640704CB0D930C12F5FA1C4CEC5883CDA699E29401956"  
00654 "284CAB028548FA974D6C56AD667023F9C417F067810C4C7E4A712FC3417A92C075423A57C2"  
00655 "08AF683DD695AAAA5F027CE535412DFE0029B292D32B2366DF978294FD263DAF8C6A6F7CC"  
00656 "895BEB1437CA189C2AD9B6FD5BAC06DDC3D075DF55BAA6E75865CE375DBAE862EEB7A51D0D"  
00657 "DB5DFE03234CC8F14EA99110DE78A3463546CCCE5F7C623028134968C59E247E642CE17D8"  
00658 "933B6701DB7952FBCDC6CAF7F2FD24A241D82D8BB1C0F5D55BF55E57BE4AFEB4200C1C5793"  
00659 "8F4A6B1BE9AA02FA86AA811577BCF1D6BE545D6F12046735AE963DC7FDAA9A2F7270B8BB37"  
00660 "B3BCEFF7C73F59013041B6CF9FE10FAAA86DE369E868FEE7A37F62E2E9CB1409DFF11FB809A"  
00661 "76AEEAD0F8E443582319F6F74ED0AA3229BB7A50043636E881BD139EB7442FA35F5DE1D6CC"  
00662 "227B921A1343E34030D2F1F361D7651C487282C90757765161BD781C49655A5F87BBCB68D"  
00663 "08C04290B4A2A94B61BD0C7CFAA1821B7F5641B899E90DA02690FDCFA4FB956414523041E"  
00664 "14ABD7B1AC7CDA1148A03B6EA17C49503F01707BD3373EACFD4E248BA0F8D4B845251BAA3"  
00665 "5D41DB186E10464F6FCCD2CAFC9CB06FFE9DC97FF41E07084B00B7B31FADE0E093C8968D1C"  
00666 "EEF9B10986821F1741F1A71F5F20C809F54852307FBF33A6C33C6A230EC7CB2CC6C44A3A5"  
00667 "3AFA49477037E55BE691ABDF6D4A852799453930A4", // PRN 17  
00668 "B19B7C4DE003906A0001E0DEEF57CFDD2B50514CDD6F57FAA2859061678DB9F8B5AAC67442"  
00669 "D8B737F635DC297E9A46C34735AE4D2C607B06964D1FE264B7786B8B0604781D4814E2A81F"  
00670 "3A0250DAEDDB77C121B962D0E7109A738630557F5AD47F95A0971A58704CB5B53AC378781B"  
00671 "48D67FE5D5349440261C58C26D9370DF37255CFE845F7B0F735CAACF4921CC9D55B3431C6C"  
00672 "88878D7D2C8436C9DC1DB94FF0D510BCF15DF0DE42AAC7640741B4893C960FADDC4261E6"  
00673 "284681BDBFB0F471C864FD560DFD87DDFEB4EE7A1267E285BBF2966F6A8BB408A105034220"  
00674 "F2565E38F6A15A45B7FFA82DA63965BF44A182FB4672E90BA7933A0120A449323340D0368C"  
00675 "F057DF299A8CF60035170D197104B394CA9F2A05983980A1BD93CF3AF8799E613BCEFF26234"  
00676 "DB2550A15BD53FFA306D3F616D78A6204AD1D50FF4F9972B0CFEB48B76516155791C14E0C9"  
00677 "E18457900CE13F672882BF423400D2E6FF94CB16BF3071F44CF2DFE8317A9B761FDC8FDE90"  
00678 "D70D5827521B070492AAEC90187EE48D93374B463C381C7D80178F54C676479BC92470FEA"  
00679 "7069CBA95BFDAF099A994AB8244CA786E7767B8400464BD08D5C560D61F27A96E3886A4E82"  
00680 "1E0087E7A56F96DC6F6E85F1FDFB5C82FD7BD48B63EA794BDE01439FE4C3D7F08864263A05"  
00681 "8B1C1ADAA5DDE6B99EB2A619A12A1A76A39D8B1F04AB55B67D15C7878651BA1F66216587A"  
00682 "7B890DB0D2BB53BA1C6A5E3226DB4FEE7D561842EFCB5B459598CF5AD3F98217CA720B678"  
00683 "1E83C2297C96F4F51CBBBEFA708AD4B3F682FA0C67C82A5F5797030E0724AE9715A911CEE3"  
00684 "47E80F774EC780DFE2279924F67ADF9D07FE4F56DE99E06B58099E1104DCE62CCA82D100D"  
00685 "90BAF44F26F19ED7AEDEAE609C04C530CFBEEDD161A2C324A382FC8234D405BF900F0B6CE8"  
00686 "95F9BD534A6474680A7D63C0CF7B4AF1D1E93958CFC02FB379AC09B205708A68D11683B928"  
00687 "2564C06290F5E6B57E3E7A75D30C9E21DF8F27F85E1FF347B32105A000BC613006BD7FA6D8"  
00688 "45C8E6D4053FF012D0D85F99737D045871BE017D672C8FDEB16F944E3020BA7FF0351A303A"  
00689 "0D3280D71CBDE42F55657671A79CA30A7A33E744E74B34D27453654DCC8E87911B3701FBC2"  
00690 "7CD16CAD9D6C9D1BB9AFCCDAB28DB6C2B42C5D6BC1D1386AB26669985B626BD54791930CE"  
00691 "D4A13716BC9AFBF4DC37A04184FCF711EFF515B92C943E53879B5098BCC4E87B98000BD0BA"  
00692 "2E1CA3C7A97AC95C1AD145EBA3950B154888272A0ED49289157091DB2C55D43AE77FA3384B"  
00693 "CE92E5840459B405237E194B6F5788F16B217D8BF7539EF8809FC7AE1BA931DBDB2A7D15F1"  
00694 "39EF4FDA60F02EFF946277647D675309AAB8201B5FCC1ED7AE066FC0E4F32D260DD9406BC"  
00695 "EFE7DC5DFB38D66CFA08E5CEDBA636EFC23B3CB7AC38FF7C390542C9814A86D2D8F676720B"  
00696 "7E2FD93217F916F10FF34A1AF34E3B8563319E92A3BAEE34258B4D63FBFED854E08B1A2F4"  
00697 "B560FFC40AA1048F081E66748E05EC3EA02837272FBD9CBF1A658B86B19B7DFC63436185D0"  
00698 "2C6411142291645692A92E3EA6A113B5436B0BB78990942146956795DFE003A4AB6D9ED875"  
00699 "FE7E5B895EBD69C43BB7317CA5C97BCAEA97CFC6068431B24673FEA931EC1DC1A3F72625AD"  
00700 "531CD89CBEB0E07F47B26971059FEEAE3986E23286D13BADEA3585D73F1092E41AD041D591"  
00701 "20D3D25FD3BB0AAABABA738049C91B22649E715F3710EA51C41C1705C4F1FFB1DCEC85F0EE"  
00702 "704240E8A8A20526DC01C7C839B21DA979B99F44", // PRN 18  
00703 "5805FC5E7042027201F3517AB8BA6F5C3BFB0BC7657337ECCC99CA480BA9C3D36033336AFA"  
00704 "C0BE172AF430A98D8462D46453E935805E594C18C0EBF735D90D5027067006193854E7DB64"  
00705 "24688B0715A7B72EAF93C2F763F3BC340EF4BCF90B03AF20C28A59F72114F693848289873"  
00706 "BE3869590A64658C0A3938A414BDE86BE8DBC25165217B1605EB0402FED48DC09139751DC1"  
00707 "DCFF532CBF599E9AA13FA72142E05A1D8E47A4044269A0B414976670094C9785811628718"  
00708 "C6FE5201221A5B22E29DAD564B6EECF025DE7783B7842DEA40D4F72D083D112C608C596F25"  
00709 "E09EA85BC6CBDEA39300774301C1C36790D7D9C0B59741F6B53E16BCABF8120112C81D70F3"  
00710 "0E44561EBF3A3A4571F207305BB3A292C46C46E285C246EE9A7442F869711F52D921EF379D"  
00711 "F7CFEC4CE64BAE4D7AE7DBA3CAF47EFCCEE78B477C02550D1DABE4FF4EC67A38A705F60706"  
00712 "D948C017E333B645C0D80AAC1CEA393CF3ECF711CA3D59ADD7CA02DB6C5E3F259B4ABC138D"  
00713 "24C47A650EFB96AAAE93E69F08077D0CF38B230F199C3B353229C529278DB1F5846086BC8F"  
00714 "5F54F9631423105673FE92799FFBE5D6ED167658BA5DB5AAAF921926FFC27956758CAE8B2"  
00715 "367E3C5C5C6CC81DA875AEB8CF3460275C4EE6EF43570B4F571994C7690FDA3DE311F2FB7"  
00716 "7057732B196FFC96221DE1DAAF9B8CCC21609160457365877519D2B332FA3A85B9801EF43"  
00717 "5D04DAEC3643CB85413A036CC1AA0968A40F4EB5A76D2D0C3DFCC53C97BB58396E855957B19"  
00718 "41FEF59048562171F6241AF278F6D6B580B4C94C5CED65CBC0F810D46E864C2B567BEC920F"  
00719 "6D1DE42CDF32300805E6022EF184B43C1733F809F98B129C937930B984F5539F574AE144ED"  
00720 "AF7BABB379EF1A1AE69B7B4FF9212BA817A4F223FA8B182FEA231E7029F80DB126641725C"  
00721 "36AB4AE950725335561C6FE37F960E37927585026B8ADD92E4ACA3B4500F0CBE076860D16D"  
00722 "BB90ADF6A127FBC7F791B4F93D5463F8C71EEC1EBDEECA881FBD3D2940C2EA0F8E4DEA107"  
00723 "4EB5C0C9EB34534C9C3992E988DAF8979FD8AA90F540F028708BC6D0EBBFC00FA946DEE5E9"  
00724 "D01CC272A4FE61E5886FF1347340FE64F650996F65E5702A61FB53EADCD457A5ED66DCF1B"  
00725 "019A7104842C51B60142267392DC51F86F69367C25B999B3153C1B7895298B6B1B43EFC8"  
00726 "CEC3694F90BBF1A417C48E5CC4983782F874A7C64E05BBF462220950DA9E8F56BA324002E9"  
00727 "1D934469225E57309C7F562407DB900267553B85C431A463351F20D57DED0B3D210269809B"  
00728 "976C07878C324F74758D9D507A4CCEB4AA30CC9CD05CEFF32F1174C7B0F7AF93875F66E3A7B"  
00729 "12F163920825885CBE5092CB7728A1B74B2F3FA9A01264214761DC5101BB0C18D65BF972A3"  
00730 "587F95B3E13F52EB6D1F334B15220ADA5A59497558EAA0F4ED358A995F7444B633FEB900EF"  
00731 "D9528B51FD82F5AA30699900D3EA3F3644E4FE3F6F2900ED45DFD997FA905D91521CD9C898"  
00732 "AF4CE09B6EE00A15057C17DF4C656E00DD876529CD75A169977962DF36E4CB6950BCA37A7"  
00733 "9C98F108AA7B02AA488AA4ECE05F702B468C20C81630F51F884AD74A76491DA378DDB8290E"  
00734 "CB8F5879F019BFB416BAF1D9361126F3514557403415F286C8E77C0BCE4FDF0F4EAA96D1F"  
00735 "22F189971155F749EC19687657B245D392BDDA62935A44BF6C82E0AF3593BE06745099F06D"

00736 "C4A66C0768F17916AA360988D5E578D169CA84AD1A36B12CA4929223BCC52AA9B47FB138E9"  
00737 "99CD6423D1171AD3BF87B5D405A174B0398A689DB4", // PRN 19  
00738 "F99EA1599DA410098772B05B5D7F9AFD485A04CA036BA9C71AD5DAA3EA5F1C757BC83F3C0"  
00739 "028DCB93DDEFFD157EFC2CB29C0F8146D6704D34091EAF1EAA29AB915E652309491F0BDE63"  
00740 "2172914C132CB9A533E97F89F567A7A9B0776B400FDBEE71B49AAAAEAD156753F5BF564CACE"  
00741 "2C1285C4E65B508B2803A11B22350ADF79645AB6E458B1C9E2031C4D4D6361F77B5329C35"  
00742 "30F6EF727225DE668A960FEE1F6D7588BC126D0E2AD23C2ACDED06AA2331E9D0E7F99234A3"  
00743 "91926F0DBE610C600964E89BE1842CB1F1551D5CDD4AE3FE45B628E9A990C872622C9257BE"  
00744 "40A8A124FFD6B135BDD2EBAC05E2DDE5B7DCC13C76C27C8A3E778063F3DBB3488CC52836E9"  
00745 "2A89641F4559F377273FD51FC45550C727689CA4D5CEBB26263F4268502658F40D4EC2BBC0"  
00746 "DFD0E41F16A399F02711BCFA96C2456EB8A77FEBE852726B0DE4F927C13CCB329F8B9BDD76"  
00747 "D01DD229863BABF42EA7D7F6204BF13F7355610527A38E89F66D3D73522B421AF013197D64"  
00748 "9D64EAD17F6A6BC8148D9989E786AAA94FA19077292015EF7CABA4B1545CA9CA20E5A816D6"  
00749 "EFF7FCA31FDFABC611253FEE80FF0D75518001C2DD6C604387C2656E1362A3EFC1AB85461A"  
00750 "5B55C6374733FB6B17090E6EC1186EDCE2C49EAFE84B139C66A2E8308CF2370095B0129B6C"  
00751 "E7FF6E17D14F79258BB3E555EDDEF12E634A5AF748BA40735B90BA1E7DE7EA00D2F4A8B3BE"  
00752 "D32BFAC17D10050B420DA32EB6D14035CEEA1B7951592AD75FB751FCFC190452FA88AEF450"  
00753 "80161C5544AF80B0FF93C62B02BE5D1966F77760D446F2625C3FFC91A07A5E852BE406F1"  
00754 "638B521579FF299227769DC4B88A805E168F08ABE606B5D745433584BEED712453785B1C01"  
00755 "D2B9F1020BDB7EF11936F0BC449EF26DBC2D5FB35A2131BE9C0DE2746DD6D0356294D0E09"  
00756 "076C9A77A076F6B659B7B4F54A06BE2F353A3A0EDC6A5123598DF9BA1CA852A5D1FA16877A"  
00757 "78D88F0B367E6C4471758F0663A7498CB1C8F13B03F0E2A941DA89C717A15143F4ECF594B0"  
00758 "DE2E43B657DD87E4BF536ECFA3820127B1FBBCC0BF2E53B9E0E1CF10191BA9B43D4D33FAA3"  
00759 "79B1B543C593F1BD5CC05E0B7511E8D4925BBFC5923AB0E2FA90D05ED580F56071A62A3344"  
00760 "08D3C454C892328AE1DAF16CBD530ADA7B858DA3763BACBC989F056EE1D46192CFF6DDDF7F0"  
00761 "66D2EEDCEE5BC6B72551245F8D10389984D3B3455A7B9664475CF6B837D72D42E49BA2F61"  
00762 "298DE9BCC070209D7F1E56BCECE1B21FDE69293D2560B8940223E0AE966459C249751A2BAB"  
00763 "98323F184532F4D6FA396D5C0AA2A4A41C75C638DC3DA5822AD7D3D7D51FE0811ED3673D87"  
00764 "5039D3625362D7D00356E8461FBFFC49FE1B0624F6D30D550FE1F5FE4211341B9D57D21023"  
00765 "7902BB7C62514AD872C1CA87976809717817AECE1E224733079478A5DDF5F2A067DDE111A3"  
00766 "FF911D0B511A6AEE16046039B229C00D6D10317F66E1202449DF20CF28BBAB4112948C65DD"  
00767 "7342D09059EE2701B36E2ACAA6A9948F818B61B9A3155729EEA670D4AEDD6F0AC314DB12E0"  
00768 "DA62FBA01D419F7C48B09BED87F7292FB0315A597CCCB7B51AF776509818A73A0BED7A656"  
00769 "AEBD7ACA2646605797940CB737548E83653E204B6A7A5AFE2EF16EBBF73484314C5699EC8"  
00770 "D24E899BAA0E202F37E3C8D8B5F75ED34417E608EEE4B26B453C07DFC6E62FAC215B6122CB"  
00771 "14A07739E8FCF816398DD49970AE8A216BDD8C4C478647A8D07EE8BBCE6631C2292047415"  
00772 "4B3C83942C842A72EDCE7466565D9A6BB030CE684", // PRN 20  
00773 "B23CE558D0DE0584768E83A066B9BF6A12C78D173AA3262866339B86EB8A8D5208159C587"  
00774 "3CB29E9C4C0BE918CE4E8DA85C85E070CAC29850CDE48E2DC5E0609F419CFA46EF650B9022"  
00775 "273AF14BD9EA28FA856B7BCB580D8949C1F732F87EB34BC1257205BCB6820FDF3BD1E7B9C9"  
00776 "9E114B3C7607D01235E111E3AB1F30E8A9311F0F30B601C833FB0D8C4D76AC1B02F85E93D4"  
00777 "C3D0205F735E0223410CCF56317B6A12D07F3BCD42D2023BA0676A00E7268733C41C8C7DE"  
00778 "D2555FBD2BE09584B33063BE41BFE93340955788603FD52190B35FFB8148CEC733CFC1BA49"  
00779 "F609ECD3C4512A99308043F3C377EB65870535388376C2C748D18A748F5828B3362666C49F"  
00780 "803828B4A019F26D6A9362726E32B650A85523484C4BB2ACFCABD16E754EECE0466D1B3701"  
00781 "75C37C8CE9DE236FC58ADFAB4FAEAAA12E048BF40CE084467C8835A1E209D2CD67C5BF01E0"  
00782 "D62CC78A12130F5EF2E3CFCEDEDF86D489C4C5D9AB99484CDD469F2F8EE5E14BE43B9EACB92"  
00783 "E72237FB54F9615627EEEB193EFF88AA0A908327C61C50D20C613C933D736F6C3D7F60969"  
00784 "A5D3770275081365BA8A57DF600F10A1652167AE3E18AD3715532B7C4BAD97980DFB0C4F69"  
00785 "D9137EC7B21BC72AF16F0BEC30B94A4A5C2F55074ABC6233CDF4D8D347F4FA942944D9B253"  
00786 "9D121B24E2C7FCB56BA877371AD7701ED087269DD643C4B62E4B40D3F6572C0730E465C5A"  
00787 "DE50FDD11BD30C07EFF35AA6218D0B73B22095D320448E0671E1CA59BFD52A517A50932A0"  
00788 "6C4897BA6F7E7038A39C92C20F6DB18D6F6AFB3563D10BDF48BC194B79FEA6861F1CB97562"  
00789 "81E969968B65161D3980C4CBD036467850EDE9397321913D57C1EBD7743A44A12E8C85F8D2"  
00790 "D1EFE2A8754B5ED0E93EE9649C53F28129130DEEE66E58181787137B262E1EB3FDFEE0604E"  
00791 "D1AFC798B198044DD50A3E87A754C7EBF0852953ACF3371BBFA0FD5979901252EFF69473F"  
00792 "C5655709EA193DCA2A2BA63D706C84F49802684F80C576706FCF1BCEB2892B9179A72B7CD8"  
00793 "86AF36C7D3ED09270A4077C481736BB69A333AB4BD864AEC1EFA944A591608A2469C85AA64"  
00794 "31C47F657AA9C2043F373A6932E735B56BC8177E1E37AB03BAE93A7B304EA1C5C52B244FC"  
00795 "265EBD6742F1647E29CED527FCBC93CA7A56A4E771D7D2E576C40B2EC736DDBB4FD74111"  
00796 "68C5C568C446D1E454249E6F7529AD83215EDB62F57D25272F351CB2C40965B1DB65EF5835"  
00797 "923EC2D2FC7D8AD56B67980A3DBB77E6C4415F54F3FC00A790DBC739A2FB699A20FC634F1B"  
00798 "1C177C37898A81D7693B21020C6B592E5A6359AE395A841B434B675F1C30A9CDD0ECF987AB"  
00799 "C2406178EB2DE941393AB4F96BCD43626978AEA548173413EC9DEAE7B53B175AFAAC5A1E02"  
00800 "D6EF7E27AC41268C1FD8DD83D1AC55ED45D087083C0D136048B7F61EADB94C88CBA5503000"  
00801 "A90F91A39A8C9E2DED567F7FA0DF605AD7870189600B941B99240A15310E7B3FCFC5ABDB5"  
00802 "6D50CECC06B4303BA39EFCAB2F7E9F5CC8F542E96034145115AFE3788DE0A4985EE1929D49"  
00803 "35EF975E760DF7EDB8A257718DFDC5A0B4B87024C59DCC6FEF6F2715582023C3C80EADBC7"  
00804 "AA8712FFAF80CF5E159CF5418A1CB50845C97A7F19DF3CC5FA344562DFDD819DD736E9CEB"  
00805 "7BD57AF324256934CE6DB0CF295AA0ADC5931D339027CCDE1441891067DEA40F69F6EE071"  
00806 "87F9B78DDF6D59CE2EC3BA6DD68F06851B10AFE9672EF84020DC53ECA25B3D5E527405AAA"  
00807 "DF60593A8080373A387111EA898499F096FF46FC58", // PRN 21  
00808 "8515E8CF42FC33A8CEECBD6E1B65BE307A5BFEEA9CA59FC51A382C24CCCF012C38DFAD3FF3D"  
00809 "CA3180934DD6AA52D828439F1052A959DE23BDDC3EA3F2E8CD70F7A765F77549BDE6CF8E44"  
00810 "4762DE51CBA3431439D1F1E42F1773FCA2F90E48126A000E69C0964B1955F586CDE1895419"  
00811 "083322FB4E6528F61768F8BE4CDBC37FE9D02B27193A0BD79D15F3CC9AC6DDDDC82B02201B"  
00812 "EAB86D13EA560F7EB5328AC0B534C52BB9D266F291425EF1607A19ACE89AD681815EC4FD18"  
00813 "88BF5449421D92E473C21BD44F2C419A0B3E0603EE3C5C82E56466C5A76B76F81A6C394BA6"  
00814 "93991A5808334EFF1E996AA1B1394A4DAEE00995F02E9518053FC744BA391214DF2464DB9F"  
00815 "01A5A3870173B3E8FB6B9B698A8DEE80745D7F71B70B811867D3513197B32115882348FC7"  
00816 "04B2AF8BD6FB3BCCE30361C1198B84865162734719D88BB62B7431FA39D452E0A1B0A7E455"  
00817 "34BF76B6AEC48DC25B7487FEB4B5FFA9C56CC7C83E39AC196987747D664943A600B106F0C3"  
00818 "D6B2D0D4425073EAAF89AB33BF6AD76B151CAC32A670A2B03BA31FBD8B48949D3074CD9F56"  
00819 "D39553708CB0EF2BC96170E2FCAA3CB2811EF1C67FEF6FE85CD450FEA37A04A07EA732A696"  
00820 "EAE8E76D497F5C350CDF72F8970FAD94C03909F67FC895785D5DB6C3A68641D68895AC8544"  
00821 "C6A1096DC4CCD45A68BFED7ED86961B5E035E61E9C126319A7B3C1468457C423CA4A2EDD1D"  
00822 "42BB90C8BDE2A162F999A8C3A99BE998B9BD7A14C101CE78F08857ADD9A96059DC22592B90"



00823 "488D77D6F0A6D459CA65FC947D742714A42277A79745C73CB157BB825E65872D9BEF5066C4"  
00824 "44229E2A10A97AF6B0308DF2793A2284F911C764BD1158742EF4109CD8AE6111DCF8A40BE03"  
00825 "409FA6C430829D0C9BA28B466916C67311F8BBAEA2D8CD2DF9C95094E4FC6D43F4491AC5FFB"  
00826 "F05BA6B7DB9D7FE4A6DFC386D136B1710DA5F30E67C8DE97A981895476B8E4E0DB5E1AE630"  
00827 "C8034F949AA4095DC81EAA0F645B0B440138AA7C5C7C5FAF3B57E1504A5FE477CD61A355D4"  
00828 "AA66C39FF89E2D0C9BA28B466916C67311F8BBAEA2D8CD2DF9C95094E4FC6D43F4491AC5FFB"  
00829 "BE5B14B1F911700B4A947D53645DCE170401135D0D7E968F03A9B19D2ABBD94BEE623FB64"  
00830 "C1DFC4AD9040197C319A79391016D6D44B8EBAAD5B2F1AAEA58AEFA21B5B3D9ECB66B3B706"  
00831 "26AB31CEE5CA759499073B4FDD96457392E58C0EC871B95C7B77F74B81C4CF5B1B243BD35F"  
00832 "4D540C1CE878E80AAF6C35D363B9D72DFCD4FF62FA8339BAFF4DE1EB3E146DB93E7E444403"  
00833 "E0D4290AF501C841B66302A345F5541B1C797D630B5B7E788A793192CE1F362B22B91A8371"  
00834 "84106D852391A241ACD93E3BEC192C4FE52586BA4044C4FDBD20B711D2624B6CC13E0C5EF6"  
00835 "2F2B2E00CA516991C7572732DB94DB809FA0535D2E8376685506186170E3702903A6D08996"  
00836 "BDE20F74BEAD391245680A029FB439FFB96EB829F0BFE58A6446B00B4E1894BE1B5017F706"  
00837 "156072EDFC9AB25A3BED2DB310DA1A7262ACF24A65F41FADAE3F81A32077BB9C864ADCC2F"  
00838 "92B95EE56F20CCBA24DEC9F5F75B36EF1A101D0F3FF6D730DC199869F8F26DEFC6F51451BF"  
00839 "EA2C0EE77CB17422E2790A8B29573405F112EDD704E3F46C7CD8ACF0E495C6BA9ED448B0B8"  
00840 "A9997850F9A813FB23788681D3CC04996E931CB00A879E2FFA1BA6A63F9D3BBFC2040524"  
00841 "4B0F838A040F125AC2B4999A197A2D53DEF8CB167FB7E37DA3862CE3ECB57B081DCFD98894"  
00842 "E45C6019B9229315108E37DCFC57C9F80201486AEC", // PRN 22  
00843 "4368224792AD5CD4ACD64D6956A334DB4614EE5EF8F98EDE0B4A5CD3D06819A485EE0037DE"  
00844 "64ED93BCC7FDB47E90409638B3789BBC63134308B97276556561902A9CAF4F310E93BB08E"  
00845 "F1E4B325D414DB5DD7CBF660A880642A27378A7427F2627BAB3444149181C54E9D3FB7967D"  
00846 "E33BED9AAC255A47482835D17E329ACF3F08F23AA1DDEB730D9602243A6894989EE2D900F"  
00847 "F6AC0579864C2361EA56CBCE2F39A75A12100670AF3FF1A36E98D1931D17CA68D8D031CC52"  
00848 "7EFF7FFB35AD6E046F4AF1B2B3F66210D978F8607039BBA99AFBC6A499A5A948C5C59055A1"  
00849 "40293CD9A97306BFFCEE37CE77C114478D0F73D1C8C9954780830D56878EB786967DDBC71951"  
00850 "ECF815CC4266D66F84195CA1AE0375E90017FA5C66DB8329D668BE6C0BFD85742088B7A325"  
00851 "6A3277AE2A1F4E72BBB82BB2A59B08FC59FE5F24EAD5DF43D95917A144B4DC6587C893314D"  
00852 "5C2C427E9A2D0E6922F0B06ACA646092E0D4A12779FEE4B76587F26AC1EB89BE0BA46F2"  
00853 "A212ACF50D93D3345AFE72AEDA1A0F855C45AFE14161E22DA3E223CAD6B47E591896F4283F"  
00854 "10C4073AB524CF50D0ED88F979458BB928F1C179E2C5914B4CC52CC000383D154830A54332D"  
00855 "4C0B652F98B91BC086370B04CCB1AA56C9A33B9F0A65C33467BF45C74905B67607B83FED"  
00856 "A063857AC4E326B4293483C8841572174CC8A384014945F5C3072C5575A5BE322E3F2FE3A7"  
00857 "E30F3F21118FDDAF23645A40E0A359A3572153F5C3ADBE1BBC4EBA5E063E9F73510C5D1D43"  
00858 "26D3D7A20384724279C3B4795C68BA332EA706E8127672E41DE4BF1E8B5EA60B39CAE2C3B7"  
00859 "B21A225115BAE051FB0401A69A7FC56AE4B7E31CE2CD575D82C8279E8B8499EEDFC134503E"  
00860 "6798056324B17B8AB045FFDC2A35B190B39FF212C17C36537A1F99F2EC3A4CE693DCEC7DA8"  
00861 "0861FC6F3D064882CE5581F9781289A085A300CCEFB6E80A86D961408250846A374ED00F53"  
00862 "79F446FE6E551F608AC3D71346B738C75DCDF63E19EA79AFCADD5E1E07E2F7C0AC709F6428"  
00863 "FEA1DCDD835E3F56B026CD8DC9D4629AFDA9015295C2CE55DFD9C83223821682B5AF876B1C"  
00864 "E156343A0F4C1655448FEDBC992602BED5DA4B299B86EE2F86D02C757C6596B32D60357B2"  
00865 "1F9DAAA06EB3BE2E9117D2C99DFBB5328B3636F41A06A020E3CB47851891AC3825369C037B"  
00866 "0C0D8412204FC3D5FA9585F3218F01295375A09463AEC65F591A594551E19DD00D5738086A"  
00867 "31712AB9FA7BABD4E9C10E9F4DA82C1EB2F3319E852B96E61EAD2F85BB6765DAF874FC8AF"  
00868 "42746F14B8FA489DFAA9540B4D0470CB31F99B10C335F672C2E06E8A2378B9DD28E8F56BC7"  
00869 "A4C017327404C79F1E8E67BCAEB5C4C793C176F0A3C1833007FC44103DCC6E273C9FD1909A"  
00870 "EF16562B5756B7F4D69F4F2EA57622DD794777BEB2F34296691A1FC38AE44FE60A44AD89"  
00871 "4CE92DB77FC0246C779C13E8AA16648843BA68CFC5E4E02FB564B0ADF3163C5E167C275A8C"  
00872 "15E58659E2F70A3AF792565422D3327CCE25C9D488EF3C32B195066BD27BBD4EF1A20AC6A"  
00873 "A22250BC0592ABC734477E1112DC3913F6B7CE323FD1F7B3F8299B8C9BBAC23487F735070A"  
00874 "FD5C3627506D4C96B0A1427F7EFC07EFC62DDC5455A77B57D3FDOB8CAD30A346F8C34F42D"  
00875 "87827FAFADD54264A323907A89334C0E7F98AFC9060D7767BC9D8D261BC0BF64FCA389BB19"  
00876 "2A9B4B00C8556BD254CB0226FA17F3BF21E2926CA52CE5DDC401E74A92C7989552CA66675A9"  
00877 "BB9E9C198BCFFC488BE1DD5E47DEDD94015BE09B44", // PRN 23  
00878 "30F77B7E40E42A5245A26283CF3A29B02DE3398DE5179773F2CCA4978059495B9DB1306"  
00879 "9B001707462C003460A024DF3AD33C063F9608783E79E38F706B92AB117029EBD6156CB5D2"  
00880 "78149875B95AFBE877A21AC3128BD6D0B11D31DE7A2BF7CCB68911309BBD2215F592D56AF0"  
00881 "F700D962E49CE46D089548C7587B0598374C3D7BE80F7F20D6855AA28A27AF668A8AD3552C"  
00882 "FDE3CE0B35DCEFD5240C8FAAB89D013D0DDF5B66E60328AE32BF4EDB8021E43762E96E5999"  
00883 "CF23BF2457B1CD75D1457D4926E7C2A324BCFF70A7CF59047DF3F08EAF2922CAE6A7BE1BBF"  
00884 "A3BDDFB864AC965B7FF093CFE74B53DB332B411962DA01D4E6C62CE6731E893574A097328C"  
00885 "A76066ECF3733C987F98C31AD74EB4A863084266869D28C07F9C5B5F10385EE44E463324A81"  
00886 "D523DFCFBBAFEF20F3F71FDA33B7EC1B2A35A98F01D13693CECF23E00CBDBA2A2D09C8A86A"  
00887 "819E82D781AE88B9F207889EDA1EBC54CADE6A69A564ABF48F950FFBE7A86243C544A78362"  
00888 "D9BFB6CCDF20CEDE386B9E3F8D7EE0E4A58596F095AC1B99E4F787472A734252D6C00B177F"  
00889 "D2B99A85D49B9C0F32D728F7D9BA4DF96D1BF6C365F5F4EED7D38727B339A223263B58560E"  
00890 "CA6778069DFCAE8703AAD1BFAD59CE298C6BB464552408946265B1E19C1FFEE000771814FB"  
00891 "5E4688140E1333B270F05EC9345E8BDF47B412F295AE3DD4DBF33D6D2D62582887D4CBB9EB"  
00892 "55F5FE3C1BC32DC689548B66893356704968C143AD3CC2E48BDC154620ECADF336E218FF1B"  
00893 "22521D183EB2D13B15A15D39A42A25B46F1AEC07A14CB805AC382DAC8CF96829B8F1D241CF"  
00894 "5CDE5D849DEB242295D7870432EBB642A45A6C6F1C2786B88C883B9B47343E822978155D07"  
00895 "9AD9F6C8CD4A21479B32B2112EB0121C3C74E7C23284812F74A2B413B9A18D6372AA3E09BB"  
00896 "04A262E7FCFB6FABB074C7C603C89FF607F83DA32DA50F2A19C38BA0965BEDC7A96196689"  
00897 "5E17A20A4D8028FAF7B6947A1964601DA45643C5FCECEBDEAAAAA3EA5420F2984CB606E9F3"  
00898 "191EF323D644635EDD801130893D84DFB19E661E04188C6CE596DD38C11D1CE5DE1EDE0D15"  
00899 "D8FC427842085D300E6801CDE02768E44BAFBB302684FD945C9AC722561B0EDA92E8D845E"  
00900 "B99F6FB1889328C74D6D9CC6DAB59D55463A613AF33226660C0C034B896075D9EC2F51165A"  
00901 "754A7D2393B15955BE327587943ECC1F3BB97180E751941728B94948B6648D2C606D66B9BA"  
00902 "1898FA28EFA38A413B066D715F7DC5F0C755B6A3126ACFB606EC6B12E332FD71E386ACDD1"  
00903 "DEEF1533D6386AE5994BFD857380874B14ED836D9DED47DE4760ABB05E80DC622C8D7882CD"  
00904 "74B2720884DA747BAC3ED9A6EFA2287D5F7D6AC7821E79C79EAFE7A68D2543D7CDD6B63A1"  
00905 "3FBA4B894A6D0B3E14C44876364C9FC93DC066CE953E98851A49B6EDB38661C9316C8ED9"  
00906 "02EEB894321C84331F6646A5C1AFEF832F38AFAEB89D181D3DD1338DAA236BE0E87A6381D0"  
00907 "0845CCFC4DBEC3D41F8AE1E6F39167A499945DAFE77601147EB29804A8B98E7EEB31DE20C"  
00908 "84E9148BCCB15CD0D21E3B519E0A2E4F70F57A69281E651B189C08D575D5E997E3A435B4"  
00909 "CC86F938BC79EF4961CD03DF8B92A8C507DFB2DE4D5EB0ECAC2F3D712A46CD3EE01B8D4DCB"

00910 "845D23E53A2E01DF3FFA100078F7B2467A45BBB4A18D14E0CFCD089433959F07F6F60F651D"  
00911 "D7F9828069D340D048FC079E4F8B92824D54322611A3698FAF181008DB5E3E327EDC431A8"  
00912 "F6E481484BD5A429D0F5DC55AEA8A18F719A7865B0", // PRN 24  
00913 "A7D629A25F82015CD70688CC461244EAABDCBF2CFFC2D73A3247F4264101FEFBD5759FFB8"  
00914 "3E8ABE9EF586662DA3CF7196385C05D0E5BD50C71D3CED424AB805AD632F2C1930868ABEDF"  
00915 "493BB4E3A9DE538DE35146320EB9F5B71C40E6F32E9243E6A25F503B68EB484ECB73C4369"  
00916 "30F30E2776D17A2E5F9CAF157F368E0FD9C26A9E9AE30A30E15DDD8A56482C85B46F973D00"  
00917 "0292CFCF0021577E5DCE8220353E23DD8507D0F561BBCD96F6FAA451E9C595E00CE0F99C8C"  
00918 "40359B103D621CE0C837A3D1598D814BE6E22B7A509DBF1ABD2EA60955364791A591041464"  
00919 "6F9928B5676ED590F41BD128A55549170211A0630BF5C5BEFEA204896C7FBB40F0C72E40DF"  
00920 "19C95F3A0ACD641834E5BCBD26E91315DA34708E2025E0B7D38F9513945DE44AEE57077C8C"  
00921 "C32C707D9A39BF098E4928420B3FDE8C74F39F81790D827A628B8F2776251F51095C1193B2"  
00922 "FFEE8C93F49DF6D4E3C2E1BDA52F7356394BD5A675543493E04D602F332761AA1D57DF3EF8"  
00923 "213E4E9B8D4BF6C8BE8BCCF661507C394CC9D5E9DD4D81DEAEB6D638FB9225247818710E47"  
00924 "24111F522610075AE757AB5ECF6FF32CF087DEC9A047DAD8C613B22F2AFD31466574AA87C"  
00925 "F0B8545B38C53A8F4597D5F1BB6837AF26114D72C609B624252722DD867572D449ED7C170F"  
00926 "99D4ECB995B76C8DBD14F50F3DEA077FBE4F838A03AF1DA6757C70C5F3394092D7EADBE1E0A"  
00927 "859A8487F70D9F7EE523AFC08DA5A5F8E9260D6330773FB7EBE01373C719129E483DE3D1"  
00928 "32697EC4C108654256315A38D7060C6F52B4A444EF33E87946C2537A3FB3B5FACBB0F3292C"  
00929 "7F2498F0A11509FD9678CC7C678D4623DC07D8F218E4B65F2EB844D24D696A31483E1EDD10"  
00930 "63ABD60D7A110F7992E3E46F8C7D5C81AF02D598440D79D649AA415E5F3ED033A8FDCADF8D"  
00931 "BAC48F9905BACB37E31ABB0174CC08A9584E25B94423BF395E097E2D0B7ED76B55C6231F18"  
00932 "0D1A7B969F6B874B9255138A1C3A2847EBA159C7EA281433A39D2F88D724B6350E846B365"  
00933 "BE838F25B904E7B02B12ABE543961D33D991D0DA93310CDDA1EF551D7B6A02765DCF0D870C"  
00934 "2BBF578BF8457CA1B599D0FE445B1E9C39E9C8756EADAC6A10F62C84F60A27268FC5B436A3"  
00935 "23AC5BF4A9ACFC55510558C5A354D96A5D74F69687385C123C497543EC14C40117ABE46C22"  
00936 "FE54CD4D2D6F4823A7F3DBFD3A1FEED428B90945B8F6B2C9A162F9C0287C758603E2C78456"  
00937 "11DDFC37B815A62A1A665C3FC003DC9F064CD5D950DEE02F48505CBE1A1E515E6516E926FF"  
00938 "6DC0123E1E4F94587F1F1568DFBACC3728BDC1D153196DC91D9F6DA84ED6B70D0DBA7E7DB4"  
00939 "A71B224E5122328D1DCDB9FAA98E88E9823A49E7560DF822D67C908E9AFF7D429E84350163"  
00940 "E49AFA0415337532B6E928BE2001CA202C5D975A3E54808AF8E110575AD5FCA28185B2A7D8"  
00941 "A3019B97AFBA933BE2F9CD5F279266EBEF52E0B92114AACCC30237B712FE53DB1B8596CA72D"  
00942 "C82BEA99DF8FE945D390B8DE1C00F24D0D9C619651FB16A280102CA8BD7FC37F868E4649C6"  
00943 "36064F4D0EB06D06595A0023E6C5D32654EAB9C8EB01186A32C3E5DB9BA6489895FD50A26"  
00944 "CDF1BC27D599F7A8BD1922B30187E0DDC54AD369CC61E958661C4FA76054C2C6658AECAD7D"  
00945 "AE23A9F36DB180E1141ED6A1084A8EBF8975325A1DE4A679BBD0E022BC35A07B40C52FA1B"  
00946 "0BCDFE17BBEAD099088963D7937CFA5B51ACCF722FE4F9DF99DADA8CB05C17B6A6AE884F1"  
00947 "C7DA09049027D6EA570BAA064736219CE1DBEA7DB0", // PRN 25  
00948 "9BFA7777848D427BC9D39876B843378E100DD8B7ED269AA91097FA433CA7EEC7D6210E7F10"  
00949 "0C449876F83439E3E494941D8CFB3E11ED7AF04EDFFAD2729FF75A8FB14A9BCF386ED8D95C"  
00950 "046A326A9A92D3F316C758F6939F14BC8878B917F3DFB5AA691A872CA44E908AB966D5F672"  
00951 "FB90196D71136ED26591404202AF5858D44C8E0B0ADE3441F554040E458F5AA64AC3D5B323"  
00952 "99E453FA1565F4E71FC9EE5BAAC97552AD75D0154FDE48AD914F6D59800A7FEF91BFB6465A"  
00953 "296E217E554631FCAC48FB7FDE1244F00697C2155A3161169225109AF5AE218D51D70BFD45"  
00954 "2433BED94FAFA5920B35353013BE2CD0C32B8AACDFAC6C6B4CCCC2C222AE845FD76630CE963"  
00955 "CA22F5214033FDEDEE3296112F9C127865137CE6182BBF09D1489A1DFFA82D9B391A5AFAE6C"  
00956 "80414698703EF03BA42893786FB10C9DBD21E7292E197A2959D605EC466CD202BB4E7BB1E9"  
00957 "D5B9B307F55FD8637314E3D68C763CBE79982FCCC0147AE6D4C96F7A1C251BC35D8EE3A4F"  
00958 "F6DD9CEBF771878B80E3AE481619D80B7BB7EFCFA9C20F11A0FE18505299EF1BB195EC7A93"  
00959 "263420D183BD89335411CDD66B4809BEB30AC649CF76552A6F08E9D77C2E9C5AE05FE46970"  
00960 "5E2B6F4485395C054F401DC6CB9F581E52978BD93CD927D47472DC52D81960F98D60A061C"  
00961 "5ACBF9F5E4434B31BC3C55CCD71CC2F2DA6385A3D040DE8B8D5F5CAABB6FC301FE3D5DEB95"  
00962 "A00F01772505201B33464F2BB1D45A2F55EC907E86803A94F9340960A5E99C727517C520D3"  
00963 "38A41320E7E8BAAD6C4470300DB904B1848AC84F67F67C6BDD6AB71E2531C5C662AA9DC2E"  
00964 "84634A5ACA48C5A5CD1CDA454C6689E2484274EB1A2D1CE2D94876F4978987331D16AFBA20"  
00965 "5E1D81CF287E5E9A6DE000FFFC99E50B0B124E718CAC8FB51A708B42C0FE669894383C01D"  
00966 "020E45B301DB2E585B177FCDD1A688264A0327614A483BFB689DAA49E8F59BD2D3943BA06"  
00967 "F39119D46E5DCD0DEF78BA40798384BEB03292F231A62F7B6E456F3542425599B2BCF96978"  
00968 "626543266EAC0426B3C434AEAA1E2FE20B54F70273A5DEB9BDF92B8B4DE940C48963DDC5"  
00969 "69FBA89BA6940B7644E33D105C5BDC61C3EB7D823D4CF3FA0AC7196D6004CF6654BB4E460"  
00970 "2974FBFCF6B74FB5A83540254EAAAACFF41C5E1481B2C38E7084B944684B21B7C661F92A"  
00971 "28508F449D8ADE8881D40AF4A161A08C5846C78FC5C562CBD36ACB83284223B2891F739478"  
00972 "04B6AD40BB99A4E7D9ED1912448C99E5B572760E53DFC434D06DCCA06DDAD3BA09DE063378"  
00973 "A98943EFA0CD3E75B1B860E8D1350CB920BD6FBE30D437B96290F49E37BEBDC2FF9F8A4BAC"  
00974 "7CF8F4571DAA9F788A3A411AE231D369D35055EA48D03AF54D968EB71F7BD14E51A1E617F8"  
00975 "1B492FBC37361C601CF108DEA9E198BD1B576F94DB9A15975CA55D356C209E5CE1D8329A"  
00976 "5D353223AE3C59BC55468AD4C2756BB54B6FC66FCC3B0BBA3962EFB4ABA17B1B08B01C296"  
00977 "A578D5AD63F605BAB75C90924AD456D53CB72436F3A2086EE61D7C3479E0381A52715292DC"  
00978 "B8CD28337205724A0CFF97928D4EB54265F3110714C360A0479AE7D51126C5CCAC09372AE7"  
00979 "139C20C46AE13E4C41E89AE12097C6A198A979DCE9FCBEC5D1AF9D9189059E5E317B7B6E9"  
00980 "8C29F38E29A35FCC3BEBE3ED0267C5503BAB0FDC46C7D89766475F2676C80CBA71110F815A"  
00981 "84CA7EF91ACBA63AF793305DDA345D527DCB5549C9CCBD5B201BF9D0C82683A8865DBB4DD"  
00982 "D057002025D088B7121C118B3A6C901BFF83DC1114", // PRN 26  
00983 "18A25B1D741562F456B84CF500F138FBB34AE1D18E9BD6FBAD05D0AFA14BD1FFD10334E6AF"  
00984 "3ACC7AC2698BA01465ED5EC26B13265E5ECD1C79359095F36B1B5333EF87E78FC44571ECC0"  
00985 "0F294243FE64C4BD7833FED71F3B6471ABDDBBBF4C374519D24DEB11ABD2EBDEFB8D2CB3BC"  
00986 "517AD92CA598040D0F5F8E5607CD879C75E15755087EFF796EF432073AB647C9DCED072FCC"  
00987 "9DC74D78326EEC1BDC9CD3272180E38DF556715CD1D37DEB86BBA06B81F01C26A3E97B9475"  
00988 "B68BD32BB642EBBAFC4406B7625944603FC085B38E5E4C9ACF0D0F92A65EC9BAEC8AFD9A1C"  
00989 "7805854B19EF516BCF65125603F131084CD13BA6E72201B13C1897D0ACCB8C4266884F3439"  
00990 "F6833D22C08A68CA0EB7B636D866182E6F42B35EE9E255F3FD30014F4EF3484EF31FB1D17A"  
00991 "6FC0913DC1443439494BDA221336D05CEF0FF168A9C54830066A6E986672EB92B2CA01C7E"  
00992 "CABD5B5840049B2161F8F52445FC27379B0706866AF20800C521BD0DCAD90E8F4367929A9D"  
00993 "21F1921049FBFFCB2BFFB69B27A2A4FC9E48256CE24302A57B9B04D6E51469D5D7C7E41DB"  
00994 "6113693322E1DAD218FCBBFD47892E8BFD28FE1461AFC769286C6F75165CF945E5CD3C932"  
00995 "2955261922B6C00CA2C006070EF26141BD8B44798110A9F20E86C2CE88D6FAE122FD7208C"  
00996 "9216011D7D91750A800576F575B8EDDA2545B3D60310D61F7351CA1C82C90E18641FCD747F"

00997 "E901351524B9D0B92CCBE86067EE5F126FF4B2C582B280CC1549CB3A18D5D9FD25A09A67DB"  
00998 "83A8F3F7DABFAA3C09CDF00C6F16D722EBCE259723B97017885B70CEBBC1C164E1A5361738"  
00999 "B1F3FA426D077AA97AFC56F4F147D3EC04E52AF0CC518AB4CE8B3ED66A1EDEA49E904B81DF"  
01000 "00EEABBB5EA8437C18C0282EA1AEE445331C5B0414C2E3CEE637933753B30FBD2160E93C3E"  
01001 "7956BC032FF839B81179DC5C41CC81E6B8A74EDAD4F6EA0FC2077D3AFEE342D75DDEC1539"  
01002 "F5C00F640A56EBA718A77A88E2E8070AB9663037AD66D9B6314C268806C2856E233D37BA73"  
01003 "E61B8CE393F9DE86DF5C76610766DAF68A8DADD37C3146F59115E8AC18330977D7BBF677C7"  
01004 "55B79C902A50DC457FEC314323927BFD45DD4A5AB7AF351228986F3B03E51581C8A8D5E4F8"  
01005 "C24ED9E14FDB5D0D6BD2A6F563D66A31247D4EA17CF6C158BE7FDA81DBA14B9A1549D50926"  
01006 "58BEE5542BCFD42C8DB540447AA13F208223DE36AEE06BCC2F280333D81964ED05A275EEF5"  
01007 "6C21F0BBE0EAD9D6A8F2C70660C0CB04AF6D84E0D18D3EF56504CB18EA259BDA23AC1D8CAB4"  
01008 "2BC340448283D9CE770872F3B94FE85A23CD9A864F76FA82EB637086D12244DBBB9E7B295"  
01009 "1D3B0B6C345E513883F261D3EE312A31017BA33A08586F58084A8087144CDF634517F42AED"  
01010 "F185D53366FF728C0E7C0748A776F3932CA874D3C2C48B30D01A5F579DD1A038AE1E20D98"  
01011 "6515309A13B369F644A856C995C2F6942B7E6BB624A0DF34FBF31CD9F96C486E530A2F3FF8"  
01012 "A43D7DDD578D72B1E93338EAAACC244650F8BB37AA9CDF721B1EDC334755B63CD34AAA83272"  
01013 "5A88333A4AF4ECE71416146A4F05232968680AD41C89872E14AE1CE2FA65B7A3BC099FE26"  
01014 "5173B0B715966225FA64FB70902BA0988A30697F110B892D7D6C763FE80D6E904A2CE47B"  
01015 "ECED5E98AB2A3BB70A73D6AF7175BABF8CF8F38F2AADE3BAFBBEADF208415A2C2FD4A1ACC3"  
01016 "140E185A2534AFFE6C01EB30FF0137503349EDDC1F9BB263C18F427E2B119BFC692D5FF444"  
01017 "2B0C8EAF146FDE964C2DEC3E1C9D5699ED01A7D78", // PRN 27  
01018 "69A39F2889ADC98D94B5C678A3D3D8736BC17256367EEE3FE8EF5AC6453A0691D894F1F320"  
01019 "2CE84EA8BA5A341D1733C6BD0150CE9C0019168E67768B33B64DD4C9F0E2BCCCF2FF7E370C"  
01020 "2BDE4F403C270F36937119950121A840913CC1D663F32F01B665840EC9E09E5A9C345EE001"  
01021 "F84B2C1154359BC066D29E0405941476C5B7B11CFF124F5324D4D090E2BC49C3E8D83122DA"  
01022 "9268FF58B523EAAAE4493217609FDD5A5C510038CF383B65FFC7671119556702F630404C0B"  
01023 "C686975E853C86AB392D25C853964693AA7F4A63BEC12BF51762DB46295D2062FFFF24000E4"  
01024 "F9FDD21972D747ED2F37DBBA2D6DECE1009222EE06B846FB7D49E8014DACFAECBEF2560CCF"  
01025 "5E06EAD04B113D1FE8341F200DD8CDB3F36AB99B566B16A84F8C040ED8827F0E9F0D4FA340"  
01026 "B0ECCC98F5A3C4805A990F3B410CAEBAFB409E985A722E4C442F59AF3ECEB1AF0B148A3E8B9"  
01027 "58D9F6C0E45BA4EC8553332DF170872E60967E106BD27AAE4A6C15D391C1B4DFD77357B4D"  
01028 "4E8B29FC874947A75CDB1A1ACE512C5CD08405F2D9F21D50197DC151B5E5A735F759628BCB"  
01029 "5A70B5CC9855A3918FBBA8E6BA0F5B6538ACCAF394A809C68F81041BB81CE71F7DBB2A11"  
01030 "D3905225DC9B1F0F0D8B7E5268020039E03FD97D42204E503EF98C26B5D9DCCF5424573775"  
01031 "E536C4AA26BF4B0351530857B8F640495377E9A08AE82F227BA487FA1D945EC233EB1B42C1"  
01032 "8FEC4BEC651D4CE1313C9574976F092A5E52D1B37B5D4B33800F5C3803BC6ED2759AE3F688"  
01033 "E7120D3AC2F24729A08B2B4C8D44EF807987C156DD3ED871B0A6217E9783A78BFE56C819F"  
01034 "2C8FFC2D2D83945FE846833E5ADE3F840B53EE67E5D13BF0B7031B6C1D004D274F43C1982C"  
01035 "43FB58011389D2C7E9111B8ADD89C37756F1E4C4A931D828ADC3597429AF3A00C0C9A71A4"  
01036 "A462F922102CABA479BDDF25A088BEFDF630E2AED4351A90B9567FB6DB5FECDD6F168966DB2"  
01037 "EB6161DA84479DBED9090026C3CACC19B91EF5680A40073FEAFCAE161E3A2F0DAC9C0A5A7A"  
01038 "A31D938425D5B5A7BF6D38FE797FB53793EED791C82DDFBB2D224455349F26F147FE0D5974"  
01039 "7AB091585DAE61A32B04DCEFF19D907C1159FF69443843248D75E772B558E7AA0FFAE2A9D2B"  
01040 "6B1E9F34B79DABBC98B2FAB34F76FECC635E6E756F7724106D08F296019FDEC94FA7B7CBFD"  
01041 "780F2C0B75B1D88EA0D5885237F4D09F03B2E8E1A98879656AA33AB23583DAFF96D8FCFF54"  
01042 "8302F323A8BA959710A78A803185D88A9C971D9491613E5BA2044D0B640FECF219CEFE3774"  
01043 "45E5256FD687A15CF66475C0270C9C1A7189507E881DC6FD5B31FFF9949342586E24C7827"  
01044 "F260CDAE356F874CC7C913ABCB40E892E495CD4388AFB84AADB45069F37ABF7DF966CB8EC"  
01045 "86C0CA5A86E79350DA5D6BC2B23EEB51B9C9C352FD9D3FFAE368891D8FD4E2861B8002BB16"  
01046 "DA0C4A96E6A9B8D8FF553CA1E6D85B16F996FAB677FE58608BF58AC1AFA9482CA957B03924"  
01047 "43B84260A4147625EE657C7763B4C9F4C00466394DB4F22DDC16B72C579FB80B3861D6B43"  
01048 "64007F8E56DD8D498F3732AF3A7041A2D71498ACB3793C250CEA60DE84FCDD0C227A08F903A"  
01049 "0B5AEDC936FDBB7BC6AB8F7AF002266C8740AC3D75B0628EF6A05C0481CB283E3645F7D8A0E"  
01050 "D74153EC6D587F0F00ADB11BAEB947F03140446AF31C2BD41F1D34D75FEAE64B91C6007C"  
01051 "ABC91A07DFE8AD753CE183C611178B8F63B822F03AFBD41AEF17F6A1D58F341D172D4682F3"  
01052 "8FED397FB01B396352C37397991D2E1473784900B4", // PRN 28  
01053 "39B27D6129F6C84173251B68B742EACD5A1BBD49DA6A4DE039831B66B054A9BB13299EE828"  
01054 "282F1779AAB7FDFD82F4CC3296E9777A03C64AC4F08DC8882A9454B301264284F7226A4AAB"  
01055 "ABBD8C336CA2BD1EC6517A9F54A776446C5A12FEAE387629FE5A66188C671C80A6C83B0832"  
01056 "2C2068ABBB0C3261AAB8280F4E36507788CDAC4D3F0817327FC0D19B0BCA35D89819A9DC2D"  
01057 "5C0CED0A8A7AF8DD99304505EAO8855CDB514357F8A9A7F3F4D062705E7B26632CA617F64"  
01058 "0F4D99417CE5C90D432D6B8955DE269460B9DB3C08432FB7BA08F5A658DD79DA8E9AE5B075"  
01059 "1152847EA6ADA78D0BB4A8BD9F6E221990CEAF771B5AF603F463D2369DF9BF556A1CE6D0D1E"  
01060 "A6172BD05EDCD166255810C1FEFE029EF21147DFE8EDD5B333F6C7274997E1BC61F253AA37"  
01061 "5B9CC6B584EB8DE11CA00C9530DDB1F218870611D4037BBB88F257614EB7F2A6FF395376C"  
01062 "5DAEE4E83C5D11BD6600DEDA47D1CAD95B11E5BCE680D2F18CF92353A11840DB88C38B363B"  
01063 "E8813E8004D2BC1DBF95E86C64E060F735C404E8945EAA3DB31A27C63AD818D96EDC6B0111"  
01064 "CEE3B3E9F0E1B7DAAB22364FC1B36D25ACF77630DBD1E3CDB194FF8B8D1BE8ADABCF4D0760"  
01065 "BF5CD406E982E0F743AFA2C4B333152A74177DD79E64566BF34A10515AC0E050CA135D7357"  
01066 "E1F94F16539E9FB0FBA0BEA1C2B09D6378E4E33EB98B5DB2A2DDA5D9F29CC68766AACBA09D"  
01067 "E25B338BEE82700203628ACBDD4B828B34856910D0F2230260FF41EC371D08BE96EE2E821A"  
01068 "1F3F328B631B53BECA81BBC74A92F608F2B040548BB5E805EAA846815793F1D01D121D2E34"  
01069 "7ABF1FE62B4E00E31C8DAA01202340219430E88938D859D62B709D43934265B9DA9BDA015"  
01070 "37F54B439111126613FFF46ED0A39B2B19C22B8776618B065E15A515B752E1846F51F8469C"  
01071 "E32FA076F47A5C49541A19D54CD63512DDE1126D7850B4300610E066AD53E094F26DDB4098"  
01072 "0B47D632DA9B4524EE5473B521A8F79A4B84CAE714B65E30824068305858CE9F39BB2EB068"  
01073 "21C3C07EF2E030DA4BB3DC2D7A29A69DED7EF7B03825C663F5EF44B9A1A174ACFBB164FFF"  
01074 "F0419E13A8DDD108FCA9B2A8CF1A5A2E0471046FCFE32555E72D857ADFBC123B190B48FC9"  
01075 "1CE8F0BF86CE2B439DCD3AF6F2CB26EEBF393FFB34290D2B7EC33FEE285547BDCC922312D3"  
01076 "6D9B25AE5DCB0A03EB0B603A6F75A5EE878266221C9B606B185250231E4B5C8B9470E5BAA"  
01077 "9859A963A51B83ECC111DA4779F012AC23AD2E2DF2965A39E52BD684A0E8D2406A173465B5"  
01078 "7BA6DC095298DD716BFE30E42CFDB0F2202D5E97772E3D64F1CA9B628B6EF2217A837165A"  
01079 "1BA1A8612638C837EC3854AC557C326CE88DA10F9DA027B204CFE634A695F2B59BC57F987"  
01080 "C613131DF2F0F6A5F6081C96098DC03F82B3691AE9E4F14C32324CE495FC94FECDD1CBA8884"  
01081 "2223DDA27999E0B1762C581162FD1839352E8FB7E4FC143E8D3C774BE9BFD7BA4C3091A04D"  
01082 "08EFC9A98B3AEBD1FBC65BF1CD7BEE784B58A877D8AE9DA9B3D459521A53010D4224DA0"  
01083 "181F661F7CFECC2E74BF3C8E959D84F91904F4F8DE602B214A2E3C42B93BF904058FDBC375"

```
01084 "28D467E323039F101209D60C56BCEA60BF9C3714A05437A978F3601FA625B31562B5C9B5DE"
01085 "2C6B84F560FE35296E2715D905BCA71BAF6B577E7A03CA4A5DC816E79E21C9B688B0857887"
01086 "A25AB624E2233FE8C9A80CFE9AE7710BF44A72AAE91885BA9541939A34519C457C41964D"
01087 "4FBD492E11B2890BE2BE0E069D274621FABEE60934", // PRN 29
01088 "4545989EE35583C07E91COFF1F5FD605F39D51A0485105A71ECAFA9C7EC0748F7FE5FB5C4E9"
01089 "0F1A59B1E31F74912BA831BCB9C95A315008F4C7C72D8AFF51DD09D9E4738E72C325B12E0"
01090 "3F7D90915066AFC3FDDDE335EBC4EB34895FB9807731EE6E123A28E815E5927BC9FEB02AA96"
01091 "0308978D8DB5E131A22CD601D88069D145D4366FAF97FD7B71F4B39AB73A97C715BEE20A17"
01092 "5931D7108AC1FCEF63CBE82D9CFA3569883A448220CD92C433ADD11A52F72303BB4A5D1DCB"
01093 "DCDEF605913D8D424F513EB8920EB864641B5833A39839822AB8ED011762756D6E16540D54"
01094 "4B5CC8D9F8B671F19DE2BF7ED74CDF8EC9592CF1FF61095D0749A6404BE49AEB7434C75379"
01095 "DC02A87D213FDD6E0E6DA751657D7D221BC5EFFC8211F6C86B0F77096A81B1F60F51C23A5D"
01096 "525C3411DFF07CD5FCD54A60FE7AEA7E4C182885B2EF08BDE99995121FCDD5DDB0335DAC07"
01097 "DE1AEDBC9E988D7D90CD81D30E6C8E598BE6743D2853D975F236E6AED206846FE8F6AC9FF7"
01098 "2C585606FDF6C25F5CBEC0C47567F389D5F317DA01F4A801FA2E2CA60344EAC2CCDEF92746"
01099 "08D8EA0A0D6323A1510FBE7BDF1EB8C0F580DBD3BAD17B268BB6E232B109504CFA3DBA305C"
01100 "7F273F8FBB7D8880303492CBED0B8AE374ACDF582A945A1B1D2000E71FCA445CAB59EDA897"
01101 "7E0595734A508495F2A742E30A006D383FBB6F83BFF1D0DA35805B6CC6A54BD1A7C3BFE81D"
01102 "CBF47021D6813B0E28000AEAA0654D085A9D5A0F0E0BEF802AFCEACE79FAB3F85E09E968BB"
01103 "ED03B00AD50D07D9290F7E18904ACA4A62F0472E64CD16429547DA8A35CB1D9D532D42234"
01104 "83E7CDDDD9E406B4920EFDDE2134B169A1DFC3F2AE543CE78A121F9F46F5D7BA6BFD99CA"
01105 "25FEDAF0B2AD447902DAB3282AA9339C235FFD874C5E893E7E171CA838678C05936F9C9F79"
01106 "4300DFBE63AA842A3B9541B07EF486C4823EF1EC56ADDA3143B89052D624EBCA9DC5A3A68E"
01107 "19F7E50E4DD71F4B111B24459EA8FB211053AC3CEE16F0D2C75B68E6F264A4E2874DFA99D6"
01108 "C6B3330B634A65D1B3617AFD7C32981D73C8607996F107211A7F37C6D6FD1D72B56432D66A"
01109 "8CE99E30E16847E549E54E7A3EFDDEDD39741A55E94B1B735238093F856D8690A813E0ABCDB"
01110 "630747EC74A508495F2A742E30A006D383FBB6F83BFF1D0DA35805B6CC6A54BD1A7C3BFE81D"
01111 "1F18313B0643890492BD13B49296827D7444876A98DBF99A6FDCFAF7883E02FD4ABFAD610F"
01112 "DD2BC8306377EDBEDA3FEF03041101A489051431B91BECDE69DAA5633C8E99839DFE70F4C1"
01113 "56FC2B1D27ED307757E0335EAA699C3FBC43CF47E520CA1404054F38D13CFE0339D87FC144"
01114 "0504AFE91AF4088820F7A1D8128ABA53DAD634250CB0C467592C27D1DDF7C64E7E53085673"
01115 "75FF73C02951DAE125866C53F845852C5008A69DEECA708E5A621FFC81C3C932319D15E867"
01116 "AFB67B1B1E07B261ECB56A013C441A7DB94523F103E93D95A7A0FD95141A6342922D1A7130"
01117 "ED95F4B1FED18F3FC61315A1FF0CFA66382BF10DD284AA4B54B80FC2415CB8AF7F3E686AD0"
01118 "664C21EE9F7D020FC799A8A44A5D51D8CB4BA010659C3070EB3374DAF2168562D89FEFB10100"
01119 "67066371D029C2B09107611C41B202B0815DA2071256988918DC6E69B35F46B8AA4F11DE63"
01120 "CE565AAA1172D7CAC8296B9334FAEA08D3BFFB34264D3487ACAE3C9781C1664E800732C387"
01121 "F9D65C152948FACFABD175C0FBA3B45F9892F8E98F2BAAC48E304EC9CB9D2A41487751B8E3"
01122 "7F3B58F66710EA2A2CE02354B85E56EF7AEBE52388", // PRN 30
01123 "F2BC6298F0B96E3B736B5A6EFF22ED079C8AE2FF586E0F4D53ADF17300D11CA7B8C2E79C0"
01124 "877970ADD0579DD90187CB8095278FEA5F4BD3D9123D300100AE63635463FB5019926B9B53"
01125 "BA7CC826A14EC4B5AA5E9B6C72685B4517A4564028FB6D5584994F8999F9AE7E59D4327FD9"
01126 "5C2A45E83A63E9D9F04B554C8906E0E2095448F75DDB68C8057F54630BAFEF06232F760928"
01127 "2B1EC50A0EACFA93E8A7EF3E6FFED7F6E4A2EB8B82D22C5B9AA7B8617DC13B69386742C09C2"
01128 "055AE5C27E173E9E0DDF1D23E4928D7120B8C171A4F4421DDDEE6229656295430B936FAE1"
01129 "8254A11612B9ECAED23869178CF6B1E6ED7430958039A7DA988C1DA03E0DF5E1D8EACB7C72"
01130 "6A9108E5472329280184861ACFEB110ADAB5ED03ACF5EE7E806AB0C850AB7E2256F8178833"
01131 "94177AC0E0F5874E27CF6B315D0F134B282EE527E32A762C1AD06991C733A7F2E23776B01"
01132 "EAD68363A361F28F381F6190635A6BB38CA9461749E8E3117CE4F767007E2C38FC0EFFFF4AA"
01133 "A693664F30A5704D369C9BEE225890021DF8450EF25B28CC123EB178D113B07B99ADC4AA47"
01134 "FE0078EF3D70E82600B09353CCCC8BACFF198EDD2B3C551C212D005CF33763875B3753014"
01135 "B14F96847142410DAD4046C9C8340946D922B6EB6CF2D2BF3ADA92A492D2C43397716A4BFE"
01136 "ACA01BF2051E029D5C664E4D9C3B7248A9142092DC25FD0DE44275AFFE90848463E7E68A7D"
01137 "4CECDBD5325AEA355E111929BF700179AEBAA60E26AB2C6ADC3967AA55FC9A2BB1CAC49098"
01138 "C16CF85C569E2E4A146568C41C815C76629061A483AC30DD27ABAA1469C1AC6C527BBBA5BE"
01139 "8BB0541358FEAB34B1E3EDD7E37EAF307AC213055FCF44E65882CCC9A12AFE5F7FFFEF4262"
01140 "C19AC07E2D91ED636C3FB1D36D7B99F77E93A5DC2848B01595D8DD3C7DCB95075AC5216889"
01141 "7126055E8EA8D1028AF2718DCDA34496ED6FD1D86B49BA42F77A714D65D594965A748EBE83"
01142 "501D963F7B8D3EF55804B4FD5462D12571B77384E3D23C77ADDBE7F4BF667CECE0717C7156"
01143 "623AA6AED9C1D2ED3C46403F750951AD8BD192FCB1A7FF95A590D085E7A733DB1CA0A8216"
01144 "31245100FF92AB1F63F53A6D831EE041966571EBA5914D81F838FC3CC5C449BC39709A489"
01145 "CF31B7CB91ABDE6FC8BA6E4D41E048D5C59377EF4DE7BBFA2406C6F400232E6B502A10802B"
01146 "420A11316D93638A92FDD0BE4A368A223FF767B11ED73E6DD1AFDA6508D4CA6EF3517C336"
01147 "2CD1604CEC574DD35FABF6D5004208146C93BF179ECAE54D9794FD13C34053FEAE2E7286C9"
01148 "46BF2B2371D9B8AAE3F0114BC138EB3C304102846E65E7908B40358BEF9F290038FC645979"
01149 "951AD82D98106440CFE2261C5BDB6BED587C19EBABC54469D6ECA9D9DD59A92DB30868297C"
01150 "ECC90AA8228E28C4D1B074881B939C60D8A4530A193290C5E5399FADEBDFC42CEC5FC2738D"
01151 "B18FFE9FB20B7D443FA3F1FEF91E65279460D27F6308B6B0F8B450452DA56E81D37851E8D"
01152 "589F9B0822ED525BAC86575FCBA54E67BEE4512A7F215820E1885F79A5406772A8241FF965"
01153 "33C644B1B5200FDDF4CD1F270E6EEC77324DEF0150183CBDE7D042C33DC0C27AC6F17D670D"
01154 "379AA778E6EF48D41DCB6C9F0F7C5E57BD0A7875D4B2D8E31AEBAA58AAF78947683E373AD5"
01155 "17DEF236B037F29C2A09AA4D596609B7AB761FC8D7F566A6BC8840CA626BD36B1D50AF7434"
01156 "2BE0B54475D0AE3E30CAE146306AA540E0D63AC32525DD66D5DF50AF71CD3A651E5A14519"
01157 "2E1F05FD31AD2F37719B62DBFDEB84EDFE0D64490", // PRN 31
01158 "9DDBC6F5221BD70BDA886C59D62F7A95BEB41276DC3B5B505D883DCC478B697388421D5C21"
01159 "7C8DD2E56A4130A28BBD7176163980EB820899CF74F3074DD7E74A77B54677A35EEEA93C3"
01160 "2DA4EC4D2E4196366E40FD0C2A471575ECFB8FC927142229E63FAEA97C51E12809FFD336B"
01161 "CFDDEA3FC03A1FE99EA685DAAB6F27DA6BACACD5D432352FC6FCC323EA9BC72FAF2569A7C6"
01162 "D55E126D0F0963EDD6CD3CCF97A033F74201E4390F64F06F89080C83096F8D6E90AFED3BE"
01163 "6A54202A59ECCF101E5DB790E997918DDFB1A160D472490F3CCC2B0E1C10AEC61ACCF136"
01164 "BB6A8735AA86C50D2AC986721C003A2A8F1056BB9910F5A8D62D358DDB31589280BD14319C"
01165 "27A13D847D84C02A039B869B52669DDAD47CBEE6892E2A881A020867405FC1756DF2AAF2E5"
01166 "706A16F53E1E459BCDC430C26403F60B979A4D74F78CCD34E48321E67A8B896A162C873CD"
01167 "EE8D58BE234C34FA20FAD472F0EB570F5B2BDADEDEA0D0483176163E20E47EBF51262B1944"
01168 "2016EE5867BD13F45DA7CADFB2BFBF14B4EAF986951B024D876150CD64B7B0A188CA79B122"
01169 "F274FFB3971BF88DB59D36437AE8BA43770DD69E3994F8CD673EC3E1F0A51E798271AA7793"
01170 "CACD7E53E27A325193A9B9FA33ADA3BDBDB9A916DC8BC6188C3D4C38848E3506E38621DB57"
```



01171 "C2B0FA80BE5CA34C295159163F23ADC249BFB1ACD94FF0A42054C1A2C304F3C515B8B44F2A"  
01172 "B1A7608784B959FFA905EB5541D14D5F12B4489403645A15F004828AC8FD3D8B44C5C070E"  
01173 "8A64BA086462E5CD447C58E0D96130E9BA3400CD7AC266AF3E3D33DF214D3E98C6B7585C55"  
01174 "2979738879552FC758F04B122889E81B887B653FD7D4A8A5CCECFB0850B58CFD4BE990FAE2"  
01175 "23803EAE972D0C65A97960421D3FF5B2B75079AEB42A92E0D0EC2F04967C91CCB84FE6B39B"  
01176 "6B136CB64424686DC6CD4AC39C26BFC8E1478AB6429D4E9B4877EFFC8740FCB5D9968ACF8"  
01177 "D69C771E9E48CB542BA5BBF0327A4C55CC83E466D6DE0E639A54576D85A44F305958A325CE"  
01178 "98A45E2C227AA9E85BED652F3EA5CD31AB99E299930F5AE92B12E10D597F69272ACC8185C1"  
01179 "E83412A67CCA39DB06A77329475A3A87A1D4586B191E2E8E7C3EC39171CE6E775E7B369C8E"  
01180 "ED0B2E74A7233CDB5905F6AAC099820A227B6FA0FE5057CDCE6C8175F4CF27D70EE76CC0D7"  
01181 "69A324AD66D013A3F699C7F024C22C03B353177C3D721EC56E472AD6B383A653FFCD078774"  
01182 "C728258146BD892B209C0A6A8CC60069F160D33F673C139534362B58E065F7D117FE95F"  
01183 "4A6DB7CC75FAA408C4FB09B7026DD640BB08945B364646D9154D00CDF5BBE432390607904B"  
01184 "C3D70A0207F696152CE6FC5A0BD79625E437197B929203CF028384558071B625E0F8801EC1"  
01185 "B13F6DM17EB8135687F08071DFC3BB06C8EB4ED296C821CFDD87F2779B0B913E623ACA63C"  
01186 "5FF023C58EAB547BF1A67EA020F317658E1328FC845F65464D2E06ABB63F678A94614E3EF"  
01187 "E281E41ECD6C85FE10ACE427FB1960BDF06630B20CF0401F450D71D3FCFC0C477682FE342F5"  
01188 "511DE926AA2E776CB9C798058A7576E1F76ABFD2C86B8C741230D6D95ADE980ADEA22547B"  
01189 "88E607AAA8C3462FA34114DFA46046848CB10482314D1240516AA7EC7160215A54F69EBC39"  
01190 "FE4CC97EF710BB7EBB55F9AC158B7F9C87B738FD993D2F0E35E4E9969FE5FC4B301BFCBA8A"  
01191 "28D71506025BE7635A2FB48AC65EA29B829DAE037D871127F80CE85BE003B4F94E958E493"  
01192 "D7C2236DF970D7DFE8495CE083E79B777D44F96C6C", // PRN 32  
01193 "33282734E9BBA791CE9CAA5CD9F80DEA9EDFCA4E3F5E32FE046FA978612D25F79FEC5DCD02"  
01194 "6666C3CAFA0B5A682C9FC6C97F54BEDC056A6D900DAD88CBA846702B0BC435DD5ECEP1D3"  
01195 "356E85247940A3892CB61BC8FBFDAACBC1DCA9208DE593DEE6944822EC80F823B2ACEA3FFF"  
01196 "66E4CD6F542CED77C81F29F144C9D0FCF3B61C4887C6B42EC97C83ED2C5DAF28CB8C9B352B"  
01197 "37255F54835EEB978966A14AC91D58857CB7FED677EB87F8556B343BE0056744655B6D9761"  
01198 "5F7F32BD4A537D8D586F477EF6B2871AF0524F71763748C57DE63D62D6824EC5075498D492"  
01199 "A2C621FCDA2A5540038A50E6765ECD69CDD45D10153171DEC8B5AD45A8A6D3C609E350023C"  
01200 "32E1F72B8D5E1AF803F68E288CEB2F5CEE3F10B7BCD7FB15D11DB344526BDA913D5930776"  
01201 "EEA446337EC5A50F9E4CB230BA735B656EE26BD6BB5632B907CB81BB65EF4D85BA379E4C1"  
01202 "64DE83CDCC023FD458606E09552303B8535B63E37FF038F2AAFDDBEFAC8BBBCF04EEB41BF1"  
01203 "26C20C533DBBADA8FE10771F01BDD14AE113760DDE4C0D71FC7C29EB8D23D668E401B3A026C"  
01204 "46913E2F08DF8434FF64B48036A37166F32CFC8747F13B656BA6D18300935195D0411EBB24"  
01205 "E1F47298AE9F9617DAD3F39C4251566CD92A64860949BFEC05EE2EDD7F2DA0FFBF3D3C9F5A"  
01206 "349130B7A5BC58839000431D00FDEB344C0DF86073F73C6833C091AE132D121F4568D011B"  
01207 "A0DDC27DFDA5284478F5AFF037D89DD8EB9FB0EC066AF20740DAD8D29C1866DA0998E580BE"  
01208 "BD311BF6E3BBB31D89D2DB162A680D2AFAC4903F9838B616EEBAB56C2D0386C65CFCA946D3"  
01209 "3073684FA119B2A6411E1786C723A466C7070E58248ECBE1D86C6D6CAA199944D9944BD6E8"  
01210 "B108060B612F8FE724742DC158BC1E5E3F71AAEF1864A7989CF6C8CB2396D010E1485C3238"  
01211 "F6385D68E20D36A4D3BB8A5A265E686DF558A73020588EA16B4B0D3F74916038A9954F7523"  
01212 "2087822BBD8831222F7B94968C5A324018192A6377661AD1E15FF05AFF75FF60B60C5841C"  
01213 "08F001D85A237443EDF3E361030FEC44573B044CE66F7A036EA78EB83E1863259BA363D517"  
01214 "049B13F61956740B1DA3CCB5C9EDF190C89A2FE908916F9DF9B2BDF0DB50D032FAB059309F"  
01215 "3DAFE1723D8031C93F9CF657F17D1B214492786F0D297D3C4C01A25DF2B1C3892B6724DC7C"  
01216 "CA704E9AFA0163D37F91DEE0850D9F2AD4860BBD2266FD7EE4E3D56349B90155BD843EBC089"  
01217 "9FAD39C67050B5F8A1A9DB38EE821EF7561FCD0D6F833DD3C9FF3C5D978BF7E70288BBF665"  
01218 "F4F20260A74DD0D414B3538FCA93671DB4371F2CF6AD94070014BCD5ABAC89157836CBA9A3"  
01219 "4FC8DB4090A3BACD7D641C5FF58A5E1489E1D2F410E65650AA30D7FEE4AC3AB37D5AFE208E"  
01220 "F3023E88D96F4760120DB5FC705A005976BB27F580CF5C10988BD2221507F8A6BF84B19CC3"  
01221 "BAA186050870D886B147F98E56CC470291FC96AC5BE10704CAFB10CCDD8E3EEC9D58E48B"  
01222 "C655A0A899C2FB39073CF28252FE3FBC6BB61189B154EEAA37A797A8E6F329C64DE5836EA4"  
01223 "4F7FE04837E343963FE89B0624779212F1E5E399427ACC8BA83C10F3893C7551FFFD62BE8"  
01224 "93DCEE02AC893928E81A080CC2EC8B95C4FEF1A6B65723408F12F6E3B7B384B149F4002135"  
01225 "EF114B67AA3CCD406A78A732374CD7C72814BCA01F4D44909E6CA1245B25E27A67F2350D28"  
01226 "5E3B37218495CDFBB9C7FBF2C40FE31C57702519B5DF1D41D85F48DE4E46639104445FC2C0"  
01227 "78FCFC853BDDC3E4C5822B2F2C5AB40F5906C314", // PRN 33  
01228 "6E2FCAD2FBC021102FBD4F992C597E4B91D2AC56A22E8FCA938D9C74533912782F75670F31"  
01229 "BADFEA83FB93A0F6D449092FCB9236F50446A4655586FBEF31323F1795AC3216315BC9F4C9"  
01230 "2DFF583AF7E977A996F10989067975A53E153364B59662B6CE9E1EB65D6D69B9E9FEFE57ADD5"  
01231 "8A7921506C2BFC4D51A01F79A67D41436B519D314D1953C6D133305AC06A45428C0D6FD922"  
01232 "8B7B9228C3A2C03EBCFAAA09B9343C736BF99022D1CA0B5C25A6179EC0ADD8AC95E8D6C00D"  
01233 "6273AEC61E1196741DA2ED9436F353919A8D25BFB8234DA7A2229CCF0FDC4C65CC6C0374A4"  
01234 "ABB918E1B5BC7B929F6554C568DC6A62D5289E45284DFA975D5AF56609EF4DCB3CA54A26F1"  
01235 "00E00D48446C4C35D1D911CE7EBECC58347940BB411F475ED18D8A1315C069C63E59CE2F69"  
01236 "E6F7B48B005107A69EBD9715303563F6A217E2ED4CD373648DA437E10754CC563679093094"  
01237 "7650E6C5D1B3D1AB853A52A89D208301C54F8E7DA8D419559555859AD1B4D096A1805DA251"  
01238 "B6767DC2C0D91791E119E71CC78A884A24F5061CA1A999DA65AED69CA02C20B6EB163B2CF"  
01239 "435135D3B34FE6EF52F3AB84DE084E37958BAA4EFC8E165E2F5AAD1D84E9F7669FEFB6E60F"  
01240 "1B9F69C79DE2926B7A01BDB04CAAE8E6A16ACF9A119A8E577919CB22B58EEB7EB289E708F2"  
01241 "29ADF81E2350B211AB48043A491D4CEF1721936B71145F07A40947D2A89A584D5D0A708F34"  
01242 "80F0892E331B2B73D8B7D88B7E85F73DBE5346D801B1904CD41A8070A0DC74073217AC4156"  
01243 "54F4170F42314AC0AF4213CBD28094F84CFFAC06BB0114B2C1924A4A1B7655965468574845"  
01244 "864ACE82B8839036DEF45E88F341A5DA37A511D583C51DDBDD51577488A29D7663CCA7AB2A"  
01245 "EBB53A4EEE7F38B8637C96CDB86B23092FE0CF669858C5374F6F078FCE4E945FED340903FF"  
01246 "26A6AD6C408E390F08ADBFC9A9646CF224A5410D0ACE93380314541982A8AB6443FC358B66B"  
01247 "027A1572DA0BB7C61484CA65A62E4496053C947D5F4744B69B4A73F7801453FE2055722487"  
01248 "8B8FBD318E8B572911D09839FABFC267416B4E2245FEEA6967677C1C57A3F72E76BC297CBA"  
01249 "73AA729B890E90A4B29CCAE4DF5D959BEE30D836C859F4767A06B4A479954CF20C7D0639D6"  
01250 "80EB61A2CA11B007EC9772293EB8E6525A56CC0F380ED4F088D8CADD0B95BDCBC84C0C746D"  
01251 "4DE3307AC7050FA2531FD24164D29294D492A07024FDF1B988060FD766128120226E63F497"  
01252 "3278DC9F07F6599A1313002CC9FA7CB44A72C9C9D740AEF0984D7D4C5749F9071138BF93B"  
01253 "CC6DCB601CF5F5F60E4BF2361A083D158BE93134DBC1D4FC79FB866B0B215E230713137EB6"  
01254 "FF388307CF2F07CB07E974C8A874EB20B06C843579921ED0839F9454DCAF71BF56387E01F3"  
01255 "DDCDBDE6C15C58BEEBC137B673F122179100613D6708B6B16AB750A3A311AC85E795FDBA0"  
01256 "2C2FB2A9D9734FEE1922D984DA9F76EF65EBC9F647B280B0433A3C2E73B08A3669C78CB985"  
01257 "F65E97A6B4D64D2B3A29184EA87163B06F267FE94714977525AC061BC54F7FD95CC0C4285E"

01258 "45D75772AA3543AC00EAA32626CE2F9DC48862657C1E156E08136F3B1EECAD329072EE4EDA"  
01259 "B16F385D736AB806157409496ACBFEFEB3F5AFC91E2FC5569DA2C5DF20FFDEDEB6BB57D18A"  
01260 "EF1DF037D5A16BACAD645772C4C016D147EC11EE9996D2E791C5251CA8FB719F2093DD2E"  
01261 "4505B7BF320204556D06B91D547C9C6B0B19FB57608934D4A4F748FD07F3F57CB108A31031"  
01262 "9F7835C082FAED651ECA42EF823AD83D6B22375B20", // PRN 34  
01263 "22C6D5FC65DCB08CFEEF580D53CA6571787AA81CDD29C4918F5EFAF89E285A37DA51E01500"  
01264 "BA0192D2F45573533F540BB1569E27A8865FF94923B81AC26FAC1DD03DF79DB5B5064F84A2"  
01265 "B15E0BE7CC53BA875E477159D0352CBFC374B07573FEC884A699E93497514E7375E49FF5D9"  
01266 "D9DEFED637DB89158F30B27E802EEE4E806FD61A46F8F7D7AFF89C9101A9375F73A2164C18"  
01267 "B0109CBA1CADD42E13C9170378E28382EDFD85B80A02469D9A78637494BCDD284557ABCD5"  
01268 "756225EE3B26C7BCD597E9744FA1674C5088D3301D779B1C3A922A0723AE1BB253EF2BFFFC"  
01269 "99BC8BB4CE539BF6D2D0951F6C76BB001B48E52387C95E49FF1A9D1DC12FAAA2B967754113"  
01270 "FE59822A1E5C56A6053F73A2CAFDB0CA0B58312ED54B4F9D1832FA089C5923133F47EAB56C"  
01271 "56EB7AF20D8006EF8D9674089CC49D37E82A1FFB88C97C3267569B5AD1A57CD67A0C94219A"  
01272 "AACEEF99A11B28D01EA4329568E42A7E717437B030D3B649DAB105C700F30AEEA7B82A9759"  
01273 "ED54F890C6E9B2C9F3F1D1A7A14A245817DDD4E2343BFDC5DE774EC60C5C41362805281C30"  
01274 "19CB2915182F525576B3BF537FDD5236300DBFFBFCF8A8C9489B1A119D66181A79AE93F7DA"  
01275 "BAB81A1A3E8D5A888F5144AE0DEA33ED8B7B6392B789EF9C544F876D179DD822E390676F"  
01276 "BD579FBE27B4A2752D4878962008B7873765B9BDD3FA581FA192C0C422C15153D7FF16F273"  
01277 "2EF68B66042B5053E458518EB65C66E3955CCCE4AA212ED4980411D3E4BDC6918B30BBC400A"  
01278 "A338D5CF8235F2C3B0C3DC6D4DF10A8CE1BC117338DA3987965A5EAB6C2E8E368F59C7D"  
01279 "9C77750717B4440AADFF57D0EADFEE7C1A92EE72283D425BDB0CFA9F3BCF602634BD8526AA"  
01280 "80E0247BD0A043D277A3E575C1D1EBDC0565330FCB30C63AD02F48D485AA43358D957ECBC6"  
01281 "991D0358AF953A41C67019E96576BF88122CA76D2804B73EAB88007A121BED76F33AE0E0"  
01282 "433EA74FB2C7AD672D0DF5927D87BABFB30989EB91BBBDDCA40ED8BCAF7684E8EC9D1017D6"  
01283 "147039D3C3423A9A0081A17B70F56FA300E8F2F97C399498F4AE85FF2707B50E3A4492BBEF"  
01284 "19E14834B261F700DF4E685DEFF476ADD919D5A15A7BCA06F382E1F0649B6934D70A804173"  
01285 "732552872BBA44A5A42D64844532DCE8ACD5D20039D397BA9EBE04EC4742F4E2DCAD29F89E"  
01286 "0A9FF572579237863890DD13E7168CF2B4CA8F6A36B49A5734BC85F811087B9541605F44FB"  
01287 "EB3313D3B56EA13676CB69F4FCF14A1BD285455BA516F20B2B0D3E8922B71C0731F2BFF112"  
01288 "4067D4CEF2111FB754CEE5AB73DF629797FB5E012515609B1E9A2085643691BAE5E4A2A52E"  
01289 "904651DDB340090EEF578675B778D1E6EA868FC1106F19B0435BB464887DA311545300841E"  
01290 "10441E9316656698C879089FD009D3F8F49097B6AB6AD2D744AE79F1D15F0DADCA70C11773"  
01291 "74ACF5D07979CED4F3173FA4ABF571F6DF2112E8AB26C257D6861923CD92CBDE33683F27D2"  
01292 "CD542ADC623230470FF7E950475EDB761E57ACEE78318FD60325B871936A51FE23A63C9B4B"  
01293 "7DDA872556311877DD6A8EE053D6CA28261BD359C0C997F2A184270847A13FD3C952741C5D"  
01294 "1C47ECE09DEC3D8E0223110A627084BF826548E60886D7EA15C585F5A16A813F030EF3D150"  
01295 "FE7A4D271856623D96742E848F7797526238E56C102191600AF7756C562A9B627B1F05B56F"  
01296 "B2C768FB657986BDA95816E0842759438E262FD9BEDAFCEB594731E9A04512A22D1EFFE104"  
01297 "EB8F7E502B94FD846C71004E534FEF2A30812F84A8", // PRN 35  
01298 "E881D91D5F8EBA0700605336EBCA6A1B1BD1D58F39650CD55F056FC8360EC9A2CFD0CF440"  
01299 "EC206E98EE8E6B31558B838EC9D473F75CE2FB04155330C9FEE2C381F872C3A79331C536A"  
01300 "05C766124E69A81EBE9425387279FF9FDCD28CE6569215A65F678611C96103D710E4502952"  
01301 "1D6EF7D5A2AFF378C9C26ABB52AF1ECBBF615561932791A23A3BFF2824CD3810EFE7143A79"  
01302 "B4E14132E306C48D9B2FCFAA475A7D2AA0BAD046314E5BB4839B27FF4ACB7F4B3CA8C97CE97"  
01303 "6F11DF6DDC71144A600DBF9AD43B6E471791CCE636D67CED546A377D786CB085D0E4E91258"  
01304 "719A9233F342180C9A827FBAF77C8B5C1BF72E9A81E4E4908EAD39772DABCC8CC0BEEF081"  
01305 "2DEDCEDD7B4110C6A0B89484C2F0F6CBAEDC219BEB8C5903CD03AE727B1849735F62B385F"  
01306 "4D5C5B8C527CACDB3586AF4525BE1CE424A0879546505F86E324EFC4FBB0B9E86EC2434328"  
01307 "66DC9CCD4D9FB258886EA75AB006D87CA5C10B5BD0E0CC4DFDC6B152D65D5FB268DB6AEAEF"  
01308 "C92CB49ACD4F7E88DE2269533D54AAC447124EFA9BA62FACD2FA914885C162D59BC6BA2FAA"  
01309 "3389DB9AEA4C350843B3361C9F343955AA616E661ED1A137EC0BF957BD7DD7105856499367"  
01310 "E91849C99F0F7138CC7AE120CA36164658BF3B67BCFC1BECDDFFF49606448AC8B38619EF4A9"  
01311 "97AAAE688D3FA77F169B8BF537314C8E865F395300E5AEC2CBCBC5A8B5AAA31C811E3968"  
01312 "FCE32B82CE5EB1EE839134EFCA68E4CC4C2957329A7CAEF675AE2F990CB108CAE8EA5EC21F"  
01313 "6C442CE1B81ABC203327087146D67275300868B95052261AFCEA3EE854F38DE7CE1FA564"  
01314 "9AD7E9083DF2DFB3ED4186244045390F0C843777A677B8A82B69EBF43D8FD24C9BC0792A23"  
01315 "36581F3B45EA6F8BBBAC89A0314570084D683C76D32C2B0A6868A428DA7D72D9C6D37D99BC"  
01316 "05D14ECE27C6C1E3F849332C1CD67381B6CCC907E1A911D3C97F1B40CC11822257491AC9C2"  
01317 "2642C4FA2FC8754E6AB2574AFCE65062ED5C1E914F8DADD52ABD7A9D74E545C362880F38A5"  
01318 "67BC3E6D7BD7156C87F698C870008C2AFC78F94354C2DB9833A0635E2C63E5A47A828610E2"  
01319 "A5E8B2F914306C966D1EF8222BD8FDA2DCADA05991ABADAE250B942F8FC82AAC026696D62B"  
01320 "FB27C8731149C49114EFD2AC47250E0454E33383082E7826569AA4D6F5661A9968C1A784E0"  
01321 "A080D6BA7F8956E37AA55C07257AB5971E03069E74D245C4A2A0BA0BA397376141FCDD486B"  
01322 "2B3C865E97D1CC327E4E3320A131A91B61F5A78AEE77D1DA75A1F121D777DC4E3714B0B515"  
01323 "DD96F477E9232472353FF7FF07B0C23D1C88F7DC4516C36B3D1820AEFFEE62EC3DFB614363"  
01324 "0E1C532A4FC9740369617E80312BF3CDD7E3723F287DF666A34FEE38B3A1461E1B5FA0390"  
01325 "7125359025AC125D6E86FC6AAF0223F83F252AC0C6D578DDAA8A2841DC35AE6B6DAB2E7A31"  
01326 "EF4282DF4E20D471A3A8B28EC6510453728F981D7BC2F1BC2A8E024FA65BE18AE8CD272B0"  
01327 "5403CE05C6006AB1E612CDB9F8A71C50D618E9FE46B243AFF2529F2A0FBA42C7B990BFBFCF6C"  
01328 "4FCF59DE230EDD411BC93C398C49E76FCE8EBA32C93752B38886162745ECF4F80EB59AA3F"  
01329 "3660C4B1D4B9850632548A6D5A9F5A8DFB5A74881C0E93E38AD2FD1F0C23E66AE4631F25DD"  
01330 "442E1A010D6A04FB73559C76CB30494352B24FEFB1E529BB6407BA7B68535D06C7207B2126"  
01331 "A585BD59C43ED1E36AF6408E33EA1CD6F43BA41401048AE9C5638077C8AA8A033F0C18E589"  
01332 "9A6E9400531731486774E30C1A6292FDA7D06A6A48", // PRN 36  
01333 "74C4DB6B3E8A4BDA889FF8264534407C16B05809DB70004C8EEEB10156C32686ACC5076D21"  
01334 "A2F583EAC92954996BFEFEB1A28DD699B55B5B94A122A1349C58DDFEE9D4270368DEF85D"  
01335 "86CCA4FFEA81880739A934BB8831AD75794986D60A92AA6DE661ABAB1010F3999A57A53F38"  
01336 "68026C64307C47370E0B82FCF8C02A3155E787DE359624510D5008E229E9DAD0005CFC9D9FD"  
01337 "BE5080BED01D7342E570360E411FC27146147EE2C72931D0577CE6D7183386C754A6820BA9"  
01338 "7BDF080F6EBD063934E13B005CADA4D2D8492E143402C9D1BF8A04077C4BC7314532474704"  
01339 "BEF9D24E4566A1409F085C76B65158E4EF6B30141DB42B0A96454E451BD63117428273D362"  
01340 "9E2E5FDE4D9E53FE5F9EA2F111A0FE5F50D2B72DD26E941A57213EED915147473D630F755"  
01341 "6CA46FF8EF1F0C88E6272D6E95FD3741AA675B0ED839BDE162F9A308B3B092663B1963221"  
01342 "708404CE8BB54F61BE9AE987344228BB60921726D90FCC07D5F64BEF673E06B969511E111F"  
01343 "0293BC5E7186B28AE81CB2803363A1266E7A78B47F5A35AAC2FF0B3A348DE73C81D74E78"  
01344 "68683918FFAA42A91B89E1E2AFF4BBE6F11867DE292A6FAAA689CFF87F4A629FC354ABA202"

01345 "EDBEA3F08609D1A4C350A7BB7376A897AC15A1B0CFEA0D35DA71039A97B8C99A5367DE46C3"  
01346 "38D75A845A4CCTCA292DF57F2AD8C5E20D95670E3748E8A1BBD3D728980DED4FE0DEC90884"  
01347 "B7A528D0B4C27DA315195B7D8D8CCAFEFEA3D62AE34B40F5049D06C3AC92CC59A85C5D5D0"  
01348 "4DFE0CC89A7DD7A1B6AC4F5F3D2AD8858B83C761485B77F6D512DC2EE1DA402FC6158C644F"  
01349 "14C2F6274E8113B453F9130BB6C24E36BF84C2942F8B28A3914110BB3D48DCAA3ACB1D7A89"  
01350 "769E0B9048566671760AD83BA7F7209832AC10FDDC299F7F90A3744F72ABFBAC19382EDA24"  
01351 "D0BD5859634EC3790C7390AC2866B1379ECEDED50D5127324A4EE6AF50D9A6826DFB3E27B9"  
01352 "922E4A8652526A6DA29A4C2B9B8465E3F99E72A82A27645CDEFF1C42E77359FC913A907451"  
01353 "6720BC61EB171AF90AD427A574080970282539A05471AB7AE2EB027DFD3D496BD27746D0AB"  
01354 "A30EBD9D13AD9DB6D405CA19EEB90CB088D9CFF93B20DE07F86B978C7FEBA8AC29FD4E257D"  
01355 "23380B955CD79E93E3135BD14AA338FCA16560A123131DDBA8A2B63923860EC3EFC2E03390"  
01356 "15662C3DF1E837C879969573E0DC4A63EB7C0D6E7419F079D4CD05A1A9791330F1F1B7EBAD"  
01357 "8CE76933D54A224571796A3A046116964B98A7EB5D9FD8311CDD6BCC9B63F2B7302347C0D3"  
01358 "407219228BBA73B8B53DC3645EDE1969D345D8C23E691A58370B351BD6C86EA6EC9349D524"  
01359 "BD05D07E868567D9214414507ED507F0E3CED126F95630FC6A3B40152A318278B8605F30C"  
01360 "389C1826478635753FCFA9D227325463581DC51EF4AC47E1C49EA005D5AAB9138706671D64"  
01361 "082F6E2BC39A802685BD8EE8001031F7C56CBE5FC68CA5B14109249C210B4ED7B9B2ABB4D4"  
01362 "E83C9EC82911E9CBECE5DFAC8D4A1A6CC81B5378D29E8E339CCB31B85D9DEE3080ED75057CD"  
01363 "5EEDBEDAAE7F3CAB4BB1BFF54042E28C8CC79741FB04246ED66B3E34D83F08D98D7070D3D6"  
01364 "A032BA442E46CE38F45957D7485B17B51C562B94985FAEC8362A8D6DE43FC0EFE1529E6628"  
01365 "99F1D21BF643C22293EOA0F2E1D40EAC1934A0421C1617DDECD3D40A3AC9EFAF0F8D94FB7D"  
01366 "EA085C8179290FB4C917C7D6F8B75DA8790404F8191C0EEFCFF9DA2208E7E5D382B5A9D1E7"  
01367 "2089B3C4976036AD530BFD31F6D08C2CDE5C2E68C", // PRN 37  
01368 "13AB0304EA5EC4D96147EB5EF8AEB8FDEE70A62E36686ACDC275D2FCA1ECFCAEFDC4F8CBB"  
01369 "E5BFC2BBD343742EE011379103102BC11E824732D147FF93CA6C9B0A503C49BB2B621B47D4"  
01370 "C5971643EFC01E760EE958A9764397AA551423EFD49AEC50E22FBD9AE99659E36341361286"  
01371 "D8452EF70C662DBB3A421898FB34EDA3314A1C7A6AA9E9B5A6960156A549C32B91246A9CDD"  
01372 "CCFE0FF278C3285A5F20081EE6FE121E3CAF922C75915DA773D0673EEBBB268998861C3018"  
01373 "642A6A1B1FA1DD8DD54991254067B4E9BB95D1AE807B8A41B88B772B5590FA9F0D71093854"  
01374 "C213283093D5EB9D4C7B1D57DEB27B1E2A1FOCEA306B28A5DD3D1B23EBB7494AF61F771052"  
01375 "798C6FA104940EAA5698AC24910DF692D5C89C29F318E090F30CB12397B88702418C467EAD"  
01376 "9599670DA2F993D5B6FD7A9A43CB89274719A2F1491FC3C3E1528EC00A6D3117C5D2E0CBCE"  
01377 "BB7B91881CB7540809B5B9BD4DD37BB6B94133504981829410B4F348A5A7DC553F089C5F75"  
01378 "9F032F0F45C13991C1F442F2DAA5C43C5A6815383091ACA7101735A0A3D8E5E88A454067F8"  
01379 "D92C4090DA1035B7880A303125ED48A5E4EBDD9308BBD45B77BA6B84093210D9B011E3FC53"  
01380 "13F2CF1B20A9C08D3F4D8AD7003B32A49850808F1B78115D1E727A7BD387D2D4C4CBAA4CB9"  
01381 "4CE71FF25229B37BF1D671CA50A04EFCACFBD980263EF16015A97909BAC6FE4F981B96AFA2"  
01382 "692CF3603C5C655BBF7AD4DCB06CC3BBF8BC819CCF20619535D399BDB5FD0D40CB261FEFCC"  
01383 "D4B0B0F73684FB0D992ADF1AC6046BA8C8E0F16B6E691A23DFF7014AAA67C7F84A9429064"  
01384 "139809E1569145DF3CD8240D49BB9D9FEFC2176C697D9950F11D56979DDC01ED19EBC96908"  
01385 "93D6D8E76041BC4FE35982D5C535FB8883654FA3A2253D52F06B62022744442D880EF9DD4C"  
01386 "C2141DD25ACF2EE3B26103D6A16710DBDCD43731A59DAADC1B40092D9B166E822A8A7AE0B"  
01387 "4323B8CEAED7AC3BF3CA6F24BD79A58DDFFB59FC2E5FF634C83F6F7B03441DF9B5763FF827"  
01388 "D4E74764C3E0EEA21ABC1AFEDFC2BB9F73E3FBA165A7DF3ED9997A588CFFEA8E3C193F505C"  
01389 "741C1863C347A80B41B7668D12B0DC7CDC6A61923C102D3D9071D9937F36A39484ACF1A24"  
01390 "97F238E78A9D8C9C1B96CAB0AA15B643B81F6C5439C2336055C8DE4A392B8C110753E060CE"  
01391 "92325C3B67F0610D79D46F2E7FDA96538D43B09DED1D647E8289986EDFF482FF7E24F40FD4"  
01392 "89E4051F7D0C9267E934ED16B13827A8C3D47C192F5C5FCF6A8E82521B7062F6A23B463262"  
01393 "E425F4820AAAD8E23E85DF0448B00533FA29FCB815BC83AD29B32D12F614AF4A6C60F7F133"  
01394 "9E639706C4C671B71C99B55056832544A23C996C6BE5230AB5D8AA6B89A0E05B4235CEB352"  
01395 "C70D378F740EA6BBF05CD97409EBDC9F1F98E428AECB124F656EBC316664376CA4B3804"  
01396 "6E5640B377457EB82A5CF0E81966865908DB4B1E9B482F8740A470E4F5F0333A11E1D67A35"  
01397 "0ED7BD9C8306186E83FDF6FEA9A09BA1AE266A24AB6D3C192A4DF53D0A627EA2A2847AAE9D"  
01398 "5EAB2DB183A45632B2145C3A2FB9C4EDCEA521E4DFC5AD34FB6E2CFC2E5C38B30996542C09"  
01399 "CE9542E2F692B863CFAAB27FB549BA99A1DD77CFAB7E26057055737E6D08E5592D17DA8322"  
01400 "B0DB650AFD6ACF4ECE23F1C9613AA7670302428512238E608B6AF8E0862452319E3ADE7199"  
01401 "28AFD059A58D3E8B19F3C14DEA608F80D3C93E374C2170D2836E2EE9F600A010B97418B52"  
01402 "CEA1592F4D6C4E91547EF019AF4BB4F1ED010242DC", // PRN 38  
01403 "1193230653E4BEC0CC617609C775D5DC198C2F533BEE5C3537F381DE8AD1995303FCEA246"  
01404 "8BBE8713A7370F59EB4F58397BF63DD3AEB17BA59D1D906C853B6BB55030F2E0FC04C0BD74"  
01405 "6B75538EF16B44A6494E65FA26A294D25E7567E7ECFE7A830896D9CF2A8D97DD5D6FFFC52"  
01406 "1907C1677F195AE7539C564DAD1D78DF994A3F33DF3F0400EA6AD3A742FE3EB8A18555CB78F"  
01407 "5E9ECB5F6E4A1A13E2411902BA48D0061D318CD1026FD379C5E4D9ECC3FC5AFDF0ED8D99E35"  
01408 "FDBAAE7DBC9E228411D203FE2645169B3E8144F508F88D380EF8F9D6B6B7696CCB9041BC14"  
01409 "5BE25FBE39017DE8120DC3989669DE5B587E5F2C5C057ABEF2195F5E0AA441E05E256B5BDF"  
01410 "CF636350777392CC1D5B5BCDD59C82BF8F3F885C86F5E6ACED0F0BA503944A9E2F15E8ECFF"  
01411 "BC5BD2C1444776A5F47DC54EA0B1C9739E21E2ACD85DF739DD5761C17D71444039B3CD53EA4"  
01412 "7D67EB8C505C39690D09E52127C8F05521E8BE8ADC6479EC436B96DE2CF8784C71B585B"  
01413 "1EC1475F16D92DAE9C29BE7375C527B1CFCEAC5FAEC37C3D892525FE1ADAA164E1E90680ED"  
01414 "F0876A0C50CEC8123CF2EE2CAE3C35409ED5C5E6840E004C9F69918AC8B39899D0B846590F"  
01415 "17243210FB32D3CDB4E3F8AB4C7C42DACB17716CC5B330B009EABEB31FDBE086602F370EE9"  
01416 "172D66A8B4991CC56A0AEBAA3A623BB37EFFDDBE273DCA7B0C09649B1D0CB5F811F6A27ED3"  
01417 "370D1A1E922A70214AD71C2668077E59C562DEED43C282333BADD38CF3B01E3C1F8A55B382"  
01418 "2A7D41991CF4E4ED36D2D7AD71BF6BF100643EDDD23B6C04486DF1193F552EFE21BD9B5A6"  
01419 "A1D93D4459CB147647052906599533AF6D3CF1ACB1EB89EAF3004B2E4CCCF6D4CE8E3D6CB"  
01420 "F8BD90E6D4637335D593A3CC8BCBF6F6423388FDCC9B11A159813FABBA39263F337B3856B"  
01421 "0A6B3102D920AB5D216FF4D370C836A85B753EA0E43E2D9E9442C9A4E6A061A0BE7CA61B6"  
01422 "7B0F012CC43E42ACD298B3057F9CD3089C59B39200E8B005772AAE2C2138172DAE25B7D54F"  
01423 "CE9C6FAA410C50079012F069A9A7D6581CB6B2A3EA909E4F2C1C57BC9F127BA4EF8E392203"  
01424 "71EEFEF0254877B19B37E0785583683E5AC8062B438889130038ACD1F52098E5E4AEFC3717"  
01425 "016321C263B4155BE8418F22E523F514B1F8BE737449A48EDB9F3B38D55E9C26DFD4C8BCA1"  
01426 "11D162DE23984DF208D5AB1B82623D2A4F4E3A104E3F84F65A020BE96711BA3B3BCE7E6630"  
01427 "F8336B16EAC987E8BF10D97C62E9B21B10AE466451FB70D345A88A549967C367F6DC1C87D"  
01428 "817EB7B4B0A5032E1A72D2078AF3CEB8116B1BFA0BA64F9C1352577BD9D0C552084898EE52"  
01429 "AC1613617D842209B5A8406DA72EBA89CBB1893E69BCB0E0058E4C227ECC507108152C248A"  
01430 "CD033DB3C276E77D80AA7FB7C7D3ED6DA7CA79D469B1F2668DF1278FDDFD20CD02668C429C"  
01431 "CB2D64E6AE7EA6C65A52B49D2A18EB48CDB26861AA872D74D51BBF34C56F5455C98E02AEBA"

01432 "0D3E658FFBBB82960205A1ACC2988D274E74A3C885F1E1F0660EEC75EBF839E8E3D5057F7C"  
01433 "39DD8369F9BD6EC41A918BC8A42291384E2688F20A7B78A96746263991D57BE8BB2A516D9F"  
01434 "BFA202EE46CCA591086EFD8C97B2DF9CD939D99A87A0AF0AAC65ED40B6551E6F84E69948"  
01435 "3CEF2C9099693B895C475C52C56598AAE7A9848CA6AF2E7970FB80768704209F117776AFC2"  
01436 "30158067B52AD83AE5E87F81014B0947A183F629D319FAD66FAC270A4D115F46983309789"  
01437 "7A0F185687084816F1D2CE615FFCDB27D060331F0", // PRN 39  
01438 "594886432182182A60A8457BB3531920208BCA5F9B84F988DEC9D71119DEFE0956740D55AF"  
01439 "CF2FF33EF9801B2C9EA25E0565F587553A9DB639F3843EFED712A9FE49D67C197615E19304"  
01440 "23A24D464A113C76A8545601964EBB9AF89E236A2021D25D5A4F71E592716BEA4B4F9A9BBD"  
01441 "46A835D49379873CF2B0129E236DBD2E510F2CE164F08307C4EB11CF12AD26AC9ED8863B0E"  
01442 "F1FE28D2FBE71859BA4340B13DBF103F6A1A855C4D51717B4B6435E6A34F7122AB6C214028"  
01443 "766BD398C6A03AF19D67D7E9CE96EC95FF00CCB56851289D52FC2F2BBD6CF27F8D7BEFCE68"  
01444 "17A52FC06A468C0BCCC6E27581F11A9394226D7EF8E1E7FCA24295275042F14FE0BBE7C1BF"  
01445 "A5A1161FA57796F21C5860D6CC3C9F2D28F5C3EB0F18D27D9C512A4C7D3A3C8B55E0452405"  
01446 "643091F679A935E2B3EB74C0562DD08A45339B1517339C19B88D1E6B13E341C7B932D3613"  
01447 "CADCE8B349EFF25DFB15B9306B0520006C72C01B8E1F5D625C76778DC8ED95D62DBCC65550"  
01448 "0EEFE96F90C8F5D83E4A9FBD2BE87594B6DE62DFFBC056824A4E4EB61B6E83AB9BDFCC980A"  
01449 "FE3E2EDBB788BF3E421E62644BCF4A8023213A8FF374A14B8A1EEA61830771C6E2B48AE0A"  
01450 "C68B14601D8AAE182A840F98198A8BECF4EC119EECF5D7FB4D2920AFDB10968F70E65D2C8"  
01451 "D5C8102DF73BF6AB44BDAD9B8577FAE45C83D76A3C6E425664131FB3F38E5946B47C0C9444"  
01452 "416877D9FD7F3923C5EECAFC435CF17A806C13D79055844164A8216A605615312EA5792592"  
01453 "AED524608BFA44DF569B5C6DA880DAB9C2B7CC0171C785ED77104E08B61C7B7625E80EF034"  
01454 "EEEE7E13502CC6510A8EBEE8D7C2D6DD85A6DF2D13999042C0B3E78C0188A741B399E212FE"  
01455 "05053B9F166EF9B876B17AB413072CE8DCC6004B99233B73F1E9C494B23457767C2197B101"  
01456 "D9BAA41BABFE33224B2C1C112E1C3C5F099320603968AA648B2792E7D0F009C1C7C33D6CB5"  
01457 "B94CCA72F7CA5EECC827F34457DA961B39DD98A64C0DF1A1618DBA84681657E108C862072"  
01458 "3F31F04728814055924B638319F13011EBDA0289A0BEEB732ACBCB71FDC411B9F981969023"  
01459 "F6A3F8411D9D9681DA28DEAFC84195A442340A537244E69989DD7F5A0F9A33D9ADEF27362"  
01460 "321D4CD3A6DD3624D98E8DD1709C3AC48167D91CAC26705A549CD2E0C62621BC275530EBF3"  
01461 "7AE97458CD7881AE26497F6C2E212462AED47D5A0F6EFB27BB657D47BF6BA303CD66FBB1C2"  
01462 "73A83EC36A4A2EFE2AAEOCE1FEC1B8740E83A73DF7A3B24CA22BBFE52584824445B2ED95D8"  
01463 "B8A0BFF28DA14D382B8B54ADF00AD82B3CD5ABBD4935738679DE633243B4C6A6CFDAFC43C9"  
01464 "08492CAC991FE4000124E0745CBFC21FCA91FB90E6F1183E5E96F67240DCF790D87A569F66"  
01465 "03B185DFB5524BFC6959F717891FCDF39C7B66210BA5837A5645C62777D83C07125DEF9E63"  
01466 "71991DA53E79A7B5767B6FA08450A806B2E2148B50F9FBC79FC66C47844E3CB6EA74C7DA86"  
01467 "F313BA8C54F0FAC364F5C0D49F2AC10238D44446AB58EA8FA3B5A5F14ADF3B9FF1AA225BA6"  
01468 "62F3BD6028EB1066D7DF265BD210CB4EC7DE71CF028E354FF81B76E3C621ADDE054CBC1366"  
01469 "FE069ED6F99BB74DE007B9E2BD31D1A96536F2260FC137AD11C5D2C2108C2C8F3C893F4A6C"  
01470 "4A7EAD2E0AAEC7931FB85C5882933C128A9C3191C35FE3CA022427B6AD9447A09B6EA0AA9AD"  
01471 "70916E9127900DF0457C6B901FA2740353988E26E4B9BA1711E451F375029FED9AC0256433"  
01472 "3761603E99A1B725136B5494538C88FA2357FD7D9C", // PRN 40  
01473 "9F4D899456DF38418890A61EDAFD40C7B43FECF9D700EE269E58F40BF2CF47CC1C1EC7F28D"  
01474 "78909A53C7B79C4E6AFD275BEA739A3617E8B3C3C77ABD4EB770B9E663BF154E77205ED1F2"  
01475 "C5669CFA18DA365B988499547B2C36D4A56582FC610E0E4D43E54CAB563B7E5DBD04CD0A"  
01476 "EA83DA72645F1P26422D416AA26CB2AE11C1EA270CF7E7AE3679FC2D69E012F0C950FC64A5"  
01477 "6C7B766F909526EFA0AC383E4A28CC5C52A26CDFD2C3FD2AD820EAD3885DE63114DF5D19A2"  
01478 "EA6C2BB338A2171C73A7C10AF41A9E4F0E49142B363AE9D306378B3211CA3B0F21EBA2E72B"  
01479 "C131459E9D05CECF7C6D6E540587073B0CEA1EE973B56EA2E671353210885EBF3A6465C1D2"  
01480 "AD12C89217F1ACB1B209552F1DA1543293D6D1D273C4AC5AA5A4B9D2B368AA82A7FF8E1025"  
01481 "9FD1FE52BA14F45A40F72B75AB7DA9F73DAEF5642CC1744E6550ED6CE2E547F33598034E89"  
01482 "70A774D7B1D58FA2D4248186083BFBB8BC1A34C744B6F08AAC374C28B1C0F9E8C2B220C6EF0"  
01483 "B5C0400D3B725005A711EE66FEBA353778ABD4552DEF9E5309FD0002FEAF1F3C5AA3753CFD"  
01484 "7CFD62907CAD3B1F68071C01886748201050E39EB0DB18240C5D598D213CBC508B8F0EE770"  
01485 "8CED659CE85BCA8BE714DA5A607E023F9609EA3D790501C4A9969FFCDD7B3E7317F6AA6B99"  
01486 "CFC32FF1B74AFBD42A9773EB0615598C943BF55C1E8AA3B85D5D2EA2892CC33FA809D71621"  
01487 "B02D141C74723297318B8CA0F4F9DEA01096234025573995DD2F915D35F7EBB6F7F3A9E83C"  
01488 "2D23D7B639042FC669BE9778390A3E4404C04FC9991FA6860AC811BC07BCAE69756FE9DA"  
01489 "B27B91B91C51892D61E94DB2B486C25D13C225810620023F1B9FED1E89B002B7B11A609FF3"  
01490 "48867E1CFD827199A5F0D6FA267F60F89B09B5A1D8FF62744B386709990589D0557F5EA477"  
01491 "974CB87A86F2B69E94B758607454E3937A539C85569B5E68ACFE4911CB053CE207394B68B"  
01492 "962A7487516971740287D8E8C51CFD26A14AAA3D974068D227089093812DE910649512D966C"  
01493 "B97FCDED09F4BCFF7749E16E3391B75A4846F65EEAABC5A1CC89B1D9D52C541DC65CC468B27"  
01494 "493F76F6D68AFCFADAAACF20BD979CD63A42EA3B8961EAA2F731C6250436A5A739D3E015FB"  
01495 "32E07D373FF650CDCE417EA85A6DB8A9A24FDF62487855DA1C807943154C162AE17BA4E477"  
01496 "531100431D647F2D8AE76DEF070A0418D2365549BC66DCA95722801F3A77E435205E919B4B"  
01497 "349ABACDB2BD111087B05C05FF01E4BF04F5685665B361D37634855BBA312E96F2C6314590"  
01498 "E298770C3CFC4207A0FC944622D4CD7280CDF5AA241232B6B5E78BBE5C6684319AF624124B"  
01499 "E84CD2F1512E89D6CAAB620009206AAC993A9F4D6A67859C20A8B9307D8F8E370F2E600A9B"  
01500 "363C2A460EDF5CA007EE110F4C088CFBC2EE82D623D5EC42DB739C2267DFC81C830244A6D"  
01501 "EFFD4966480D6EA8D258FC047500062F42446230456F9B5405610863500A8AE2440A00CD16"  
01502 "E5B6136B0BF1611345CE7D70319962896B3F5E69583F8D305B760E0ACFFAD762DD06C54EBF"  
01503 "8075C1B0ED97FEF59FBB48A7A65C2251DA406D5E9E5F67586C64E5F8FCC805DA4F33BB583D"  
01504 "E2D5F85A348AFF0215233EF88C310FD1343934F5AA2A2AAF8F58DED34D0E35D210FF01AB41"  
01505 "14E18019A27D6927FD843F842AD6AA0E171DDDC2BCF6B49EBF5D91C35665D8F88C0924A45"  
01506 "3FE96CC06617D81DCE66A3C813F1D3DCD5C37E60BC672AB416C2025C78E9A1A6B1CBC4C022"  
01507 "49AACCF55862C538ECC42D0169B7E3F4B188F2698C", // PRN 41  
01508 "74A3CCDD552EDF50F80D3E5FE68A22F85EFCB41BEEB787E5CF0CBE3266EF310BA5E1B29A7F"  
01509 "FFB0945121EFCFF811A1B7DD768D3ACFC21C4E6EA1EF06E851F2FB37A5F8E0C49F59A28A4C"  
01510 "C2380AC0CB4043A7493320AF4E34A2A20E36ABDD2AA49088FCFF24C84A4D53FD6D3B29178"  
01511 "F921768F21C5827D64C125DE47F7435C6BD000BCC1C95D0F82D61581049CDF2A34A27FD9D"  
01512 "DC409895B9742443D0CF74F8A7DB13CBAD88E1F1D6F52E4486809E20AAFBD82B297B038BE"  
01513 "774BA13649ED922FC67F6DB9F0864B742F6FB60E849F9A253E8B0B873C8501F72D3523EEF6"  
01514 "2C14303585D2F1717203DD2BD4065984190EA2075B5385E1D29226236ED894DD10FF82F1"  
01515 "DAF16AF55F0940415685B3A58D1DF7CE0B0F069F33E6FFA29FFE6134285CCCEA2DBDF1B901"  
01516 "A700765449D67C24766A042D43695D1F5D241974E62817FF9259879813F37AB1E8E51F6C65"  
01517 "43244118E411ABFB6448E9EA3CF1F636D212D65775EF9FE046B80F0DC1D1480158D603626"  
01518 "F7002290058E62DA252328927BD20C05E5D3BB0CDFB6C60C4DD8BF4A3C9413241339AC186"



01519 "B054B34B24D9ADE710A79978158F1E2EA5AF6FAE9077CEFA4E335473D29BEEF00624FE6A9D"  
01520 "EF0BA52464028BA25D9A9C1BC85901F507AF046D56B17CEE0F23433959CD7F108548169780"  
01521 "B699EE6119BD0500B55945186A81B89A9C6B0E32F1D5196C4787BDBB08A6F1D79AF329B3830"  
01522 "4B1203E9B8093146B15F1207C7EABB020B72B2B92EDB95377D698F10CE7737FA3DA501E0CD"  
01523 "09F3FAE9504ACC414E63F60E2721E759CBF2CA1E2671A98656707F4629626219E4F16D8A76"  
01524 "5AF3E75DEC1B6AA82622916E47872DF33689D820DBEE6EFF26E70F61ADB2EC3D8FC9A0AE30"  
01525 "F12E3CC5E9A8E1A820DA15952401D3910F6F500D4CDB7D85401AA4C8F418F7E2A166B491D5"  
01526 "CA3B415378891B359112894274F25E9DD2B391A96636D468290AF22FC6BD5C2E7842AC5A62"  
01527 "BE24D80E6AF79C6F2118C126D60720601272FA89F966C812EB875052A0BCEC16D0E6C973EB"  
01528 "B683214ADAC9BCD0541D32D33AFE16EC2BB424EC8BEAEF0BB8A3DDB18EFD5F58EBD84B2DC8C"  
01529 "7AD54B86E98BE762E73CEC7C1DFB290F6CC1B63F5257FBFEC9EC87D8DCD2FF393B1C1D1B"  
01530 "B2EA9BC10323E96CA3EB8EDCFCCE055680109C03C5C668BB8848F38C4C3369A4E5C0515C72"  
01531 "45B0C092F0B8525434468E775F40C4D4206B6420039BE1F695DE802DD718B9AE4FB41320C2"  
01532 "20F6B9971A4CB8E67C6B0CA3049C2772D2B3C1024A502B48362CE9A1B84BF4C9D47D62409B"  
01533 "5CD9BAF1E18DE57E9BA9CCE077266D720DEB23F192C70145FE201561AA2A46B55F9B6E5BC8"  
01534 "FAF888FB74BF6510AA7E6BCC40DFD471D6CBD2FCC28B33EBC763E8CA58D5D74C1996D66CD8"  
01535 "27A6A6B9F22AE7C24B13799A7D0A9C9DE940CA217FDB870A1640B55C69B2E704EA21262DA8"  
01536 "31C7E23411D81773A965DF6AE015FCFA0D5DD389D0327C514968C2F72A0A0A35243616AA"  
01537 "D673DCF57A684E1C9A6B6F35E12C09553AEA89DD40CFF53EB16C5EB9F98B6D04674F8DE9B6"  
01538 "AC125BB80BFBFA71267A6BFBDE2B64FFAB35E5C5F27B7CA273B901D28997E62298E610404C"  
01539 "07569E906AF093A96BA9E40475EFEBE524357B044B964136C989FFFC7024DAEC0CE3BBAC48"  
01540 "B5A0CC79F20AA1E58DDEFE7BCC9B3C3A04EFA49769A50F81AA8F3D152757F33E935CC6ECE9"  
01541 "959B47BB90810BFE6B108C8DE4643B24B4154CB7EC3889F5FEF9045A59E82300275EB4D379"  
01542 "BD85A2991579688125351C1CA84EAEB7607894508", // PRN 42  
01543 "C9ED53081796789261D7D209BF086C92C36B574AFC6ED55C3410E0723C799696D9E7DDDAFB"  
01544 "114606EBF4256461FDC9B86BB21A134F735837E6845571F6C2712AE42FE113878AAA18C76D"  
01545 "5383FBDD38E43FD1314166DC0BF7CFC3F07ACA56740A582173D2DF20F7E94F0CE29693AEAC"  
01546 "6B3D1E4B22BC63395BDA6B49E30B49261FE045CAC9AC29BE5A2D91347EE4543495CCC1A38C"  
01547 "AA59593BC3DA05611BDC74806284B4ED73013A415E6B36BC2CE31181971BF4C7B488E78E92"  
01548 "1C6AC1070772EC36A9BE74DA8AC4FF68B6210A392866B689110D9BC2C7ABB2D82A38EB9908"  
01549 "91F264DDE5FEC75C6B4E539F971AF2728343F1641F4B05EE8EF3A520DC170B3D8D562A7AB7"  
01550 "2C40999503476CE1C258F486ADC9E44FEE47D5E81BF1259C0AAD3BDA4A9E17CA9C74AEE77F"  
01551 "ABA25B3481C0317199D585BC75BE5A774F8D5CFDFDA9FA652C4B14E29AF2DB5ECBE682CA72"  
01552 "4BD98E05BBFC82320C8883A56F687F840319DF2D8BDB3FA8553BA3FFFAE0D4EE33E84B6222"  
01553 "8F924A86C5FD5B3E12672DB6377D807356144009F741325714B5B6AF08033F3DD1AF558C0B7"  
01554 "6620F77056A18896D2047C967BF69ED972BDC5A0409929CC2E2DE802ED09EF87433B978A85"  
01555 "0ED7710508889EEA4558097A7FD83EACB365371F80FB6EE1F7A6D258B97BF1AF9C321A629A"  
01556 "5AB7559FCF1B8231F816836E391BDAF741E493298B49C001783921ED1569C149C90D26CBA9"  
01557 "4F8604BF9E49330CCEDD8812A95F7B0E71DE95869D2515E524083C7B2E06C4B5E9DA12BF"  
01558 "CA74049F03A352C15DF3B79C3F2015E86FA527BEA76102DCDD8C63263F0C7E43CD9D49F05D"  
01559 "8F6A124350436388FAE37DEEB43909E415EF9F5877D04CB5A86AFB74FCDC990623C9D4D034"  
01560 "816E39B692B68B732322695C97CA93CFF0FE7FACFC1DA29BF0D0C08D93B4485D666AE96B09"  
01561 "C8B011B31077F5232DCDD9B3B8DE7BFDB4CABF31D2E92AF686EB8C2FDFCBBE80B5F5591B3F"  
01562 "7AAEEEE0FC7A46BBDA652C8E857AEE81F499D9DD537139B7A6275C64DDBDD17BFF7594DA69"  
01563 "335C01AC24270403E88D9FB10E6E80A4E30E2CCE8F8DF3FD7A8DD567BEFF109ECFE3B8C843BAF"  
01564 "BB5C11ABE38A76804D43A44327FB4D4F986BE3C52DF109E4D72A092F25CE7FD384B0404628"  
01565 "936256877CDDF543099D5B0389A9E05CA93A2B12C3C89BD76F55084A03148E29275159DD41"  
01566 "A75FD6775994B8E8D9FB10E6E80A4E30E2CCE8F8DF3FD7A8DD567BEFF109ECFE3B8C843BAF"  
01567 "FEFF5B70D5747471FB84DE2A26747DC73C9B2BE9905C53743159E489E1D0CC14E72046E4C8"  
01568 "D84F893CFC7EF15CD9F52C71D91A503802FB5A6813A7BD2DB705BADB58C06E0883F0848E44"  
01569 "C30A948119058226031B5731F36B3A9177065BE9A0294F726E80B5AD2F2723B8698D831DB"  
01570 "A376061FC2029979F0EA7AC4BCC8D87B6A432051DDA7AB2387260082E6D3BFFD3B2ACE807FB"  
01571 "6792DB21AC235C1A58D9E47D42EE055E716D389066AB0814B36291AC71B0AACB666F7CAEA3"  
01572 "7B89CB42AC4EC3CBC68F97AD6FCF5F3B5094E60D68C2F1180E5CF2141EFC84DAE75094901"  
01573 "38E0EE466AE0C7469F4EAB2E5ACC29EBF2BB225E595181F96700B552BC0E1171428FB2AF0"  
01574 "448463CC8B677DD10FB5F789857A29FFE9535D7CA35F70640B9D4B9862234554770331B5"  
01575 "2E42002D123560BA7DCF9114BA9B8B23D08C3240E3EBCB7359D13767810AD3F3F66A8D698D"  
01576 "17F20DA5242259090C0A9AADB9385A2897A5A1F32A7B1549A45D4EA8787523AA210BCB55AF"  
01577 "6B15A1F150E79FECB104CE01DAC70125FAE9D1B8D0", // PRN 43  
01578 "334994DCE050CEED5471E648F7010ADD69AA0CD3DBB3679C9D3863F86319BEAEF1370902CF"  
01579 "4978B8237638507B392AEFEF731E6FB42DFC2A23142EDCECFBA399F9C583866B1BFD7C280C"  
01580 "AA43C2CEC29C6657654C251F7BBC0F83E2C356934C825BCCCB36073A5D540AB82DB7158C7"  
01581 "4934AFF166B108363405B206E53B13A51BAD149AFAE2A40A37328D4CB3961C9EE2EAD29C76F"  
01582 "70FB4E6409F468364DDE7A2439D24FBB05719A6A489513B14229E1949142301AB538AC775A"  
01583 "F716707139359471B14A65E2D20319721FCAEB6364D04C480C07D732AE0182090C517B96DB"  
01584 "8FBBF6B8B6FBA35CD3CFC87B1C8C0F2E4ADB486474F842434CE9383A57B69D4CDAB68E530"  
01585 "6DFD737AB07F5A60073846634AA6E2A6D6F7CE56A631FE1E3DF256A1E39F0C44C3EDC3A945F"  
01586 "EB4FD44772AE094979599B9B90EFD1BB9347DF2E654BB9687A2EFB857D9AC1306E20769FE"  
01587 "C4329BD54804354E0235AE443AF3BE01E26F0D5998D7FC36DDFA5EDBD3958623B6F065627D"  
01588 "6A4EC9901891E3C63522744CBC48BAC51BCC89602D66D1B3E0966398927E869E73AA8B986D"  
01589 "9FBB428BB3A71AE7B273A16228349B81927B53DD9B7E96E3D078F25EE564495BFE7B7E6CC0"  
01590 "42E016499952DCD37F171C709D4734CBD8D46E7989C2C497B5EB9613DC311643ED4B1B08C0"  
01591 "C42A5854CBA30AA225C683913B5BD46E1DA483BD7EFC2C469DF017BFCB5BB07C64A68E3CE"  
01592 "828A537115A07876D30999468358869AC7681380A7E462ECD497A6A6FB6AA134C15C6BDF89"  
01593 "A2B8D58B875C00892C50A791D8A50514029F790F7748EB0C1CDDBB07AADEB920238F9C19BF"  
01594 "A31097F9AE3B26509DE5C4040CDB82C859EE7C4D094CD7CF8F04395684BF7BAFADF75027B4"  
01595 "A47A21CBCC6E3ADF6A2AC66EA32F670F49BF245F6CDD0FECA201E3E7706F8E37C25FE87B3"  
01596 "040C0D8431A00FD4B49E7A050837560D458BB5C2E93986EBED14944F9E92463C312942E6D8"  
01597 "340BAF149042FF611DA8ADFF9DC86CD2818555E033F3E2285A80C482D52B2EB21E2D5162"  
01598 "5FDEC689595EFC3160351B5F30E1B802D5F4F33CFA2C25D8E4101298E6B0B917B5ABEE5DAE"  
01599 "3CE09DA595122AA3BB3D8BA8BF0D68EC99C2DE8B4BD7C2A0E0E32ADD32D72696C43C7CDDDF"  
01600 "92DACF13D18DE4A1403D000245E7081B92EBA207DF9411870B91CE9DCC5E26887A437726F"  
01601 "6CE2B2CD51907FB108EBFD2F089F893A6D5B5141393C08F497C180AC4A95357E417D40C5E7"  
01602 "FB324F0D72F417F688D45B980E215DA436CEAD3794A135A719942FD0C998F3F6F165C16612"  
01603 "E2B3ECDC4CFAA9A18631AD9D878D63A8E09B0F82C12EAC730E6C5A083399B7A098819F2F25F"  
01604 "B2CBD1B6E65E3C30302CEC61680B699F504E5AEB959370E32B58B9FBC832F85AB0846AE4BF"  
01605 "678C272CADCB5277EF13A38B1AB787F8E5344D7DDDB074A92DD68A2A2F9695FD52DD685E43"

```
01606 "16FA7E913DCBF34C3F0D9E9D22CA862D76DF3B5E9E3B54992A7F731C893980F151BF29EB1A"
01607 "AF677B439DB6D20AC69A5A41166799FC54FA68AFFB8D54EFC45B49E25A93B28E87E41C07B3"
01608 "C61FF24F1D0BB74CB20D2A8D817473188FF8E5EE9C4006083BAD781C3F46BCEC777B2C72A"
01609 "9A3F551353DF186FF9D7B342D86629C4085FF2108B98D8EBC886CFE3C7A4B673D447DAC5B1"
01610 "DBF6BE6064936AD014C0AEF0078C5D1A626830C480E8B72439C9BD0ECA0B41561E5B0D661"
01611 "E877042FEB6CA0F29622EB9F21B83A4560FB26BB98468F33BBFD5CB29AC4F45EE9D68A1E5"
01612 "DFA757DC218B536A9231D7FE228925F96C32E0ADC0", // PRN 44
01613 "1B2A30E6054391704DAD57B34530294AA46DEEA044EA5E707811036742E50DECFE891FF883"
01614 "72A4AF65CAA193E1D8DDA016ECC2A2E3F6D8F2DCF9AD1449C614A70D253161E2B05AC67489"
01615 "E13345C779F3D36BFD7F7AEB2AA30604692969E0C4E375D18293D5D3CC39E31189698BCEED"
01616 "3290A2CFFEA79EDE157F2A52B4EA52FCD82DC4BBD11EABED2F311DB72EAA53A1B6811D03D3"
01617 "C2A30C76A8638E6A775EB3A51FFD4CDFBB4B961075E6102D2E5555ACB06548D119ECB6895"
01618 "AA00394D4A98C8302E02F8D4FC5CA34B405F25D2B090932A10A4A68CA18D799C1DF1C86B1F"
01619 "88AB219D69C25CAB6AF6CA0B3A4AE390CAE3AA65D3AC95339EE734126226D1F292BA3ED968"
01620 "FD2CD835F621C3CB09E59B9DA5CD23163F573821FB64E296853A8263AC9DC896D6E355A4C4"
01621 "377BDB4E7004CF399C0891C1D23E39F03CDD94DF7FCDA06261E923C9E0FBD8DE32B49935A"
01622 "03E66C13FE7B00E83674E64D5CBC14C0B9F62A842A17551E5E7001BBFBADAEF36228513A43"
01623 "D4864BBD2C8B272562A6B548C0B3E4610609191B807CE861538B89FA079217C91745EA61AD"
01624 "A6AA96464A135DF216FE5DA5143F55D76D44131FE6E329826F2BA062EAC329161C8D158EC7"
01625 "A3BD5FE57908C7523430ACCAEE3091FAD4B14DFFE0DFED92F014D19641EBAD22052FF8D262"
01626 "D39D6FAB21ED36EABAB6A2AAC22FFD0BF623CE6867F3C50FE502FF535A68BED2FC98C50BF67"
01627 "33FCC7966D816082C231C18165422F7123325F365D257FE69867CF0CCBBB875D5153B8C1C0"
01628 "300EBB1AFBA3371B83488143EA2562FA477D210BF68251312A7EC26BC837F9D861101B5802"
01629 "49FF24C364DA978C0F57C3A3C29F03F3696C04A3CD330247DEC04A385739D9073984786759"
01630 "2B09DF211D8B349FBAA760D64D5E00C9D596CBB8B73E7B17317AD315004B35E7F986720858"
01631 "11DCFE7C86DC0169997F2F9EC9B2B77B97FFA6CE2AA70E2A1030BD3ACC5FC3241F291D0304"
01632 "B71D7278BAD824445226CAA77E210A48EC8F79E0DBC8A1505D1AB9F79642716146F7A8F9"
01633 "EA25238FCB2F9C46BDC1489D35B958C44C7928BB4503DB8DFE4C1AF5406C04575C4DAB1199"
01634 "5BAC0C7FD72B10ECAE376937B56A035DC0A4D7B2E1A180EDA4A84F97D53B57EFC5562643AA"
01635 "62C9761C21ED36EABAB6A2AAC22FFD0BF623CE6867F3C50FE502FF535A68BED2FC98C50BF67"
01636 "3B2C5BE8D7D690764C256CED17E4023A04E70387EC53947395B00E8ED6541CBF0B076AD2B"
01637 "2A26CBF220BC665598D9BC1D1EDD309E687F7D1E7D71F4B2C1B5BA020EFB1DE4248C234A68"
01638 "670916EC8A72FD1EDAF0615CF3FA768FEAA586B08990E05E24E7D8E0DB32F4370CEE72D74B"
01639 "AAB68F6E57F2B9B0F51691990DB46DFADDA7503BE7F64A13F5EB5D49788CE21BA768D02803"
01640 "7BBB0F3573CCB244CA51362374776B9EFC776AD71AD1F5635B6CB73822A112149208F10E6"
01641 "57E1297026113481BE7C862FA099F331FA5C6B1099AE77D4F6DF2C74736CCD9AE834228798"
01642 "6842063A65845ABF45BA037441D967FA102134B25BC7F1F9E8BD4E54306BF8C36DC01A500F"
01643 "4A1184D7142841329DE6B27BA4D772B1B2B68348228037A9A048C0989AB12B42FBEC45C692"
01644 "03ABF843423D51CA2D67FA548BB39D0996D4C215157C707DC251032ABE51686BAE8F0E0C"
01645 "B4EDB664A3CABBBDEC85EA7AC7724A461F9F650F9752C1E5B8DF204C3520CEE4A51BF8D3BC"
01646 "9A0AA3AF9D31B95FACD006ECD27889F1D0D1B696DF57403EF0E6FD83164E5900691FB29B19"
01647 "BA4F64084ACFE98567987632483F2EB2CE76E1E0B8", // PRN 45
01648 "5513F31C262853F074269AF598A2E43052CA2C0141729078A17BEE4CCD35B761B1766B3C1"
01649 "18C06724B0459AF3E973110D7096044F5CAAD9A70F3BC7AABE628848D2A4891AD18CF4A3D9"
01650 "BE900A2F42F5D54AFC7FDC7DBFD3155F9D06078C89F8DB684B62E4CE0DC0AE0E4EE6F0E26A"
01651 "86CEC1DCDD96EE175FEC8B27BDAB3DD53D465E796FE4C49F46B0B6BEFB951BB1A679190F3F"
01652 "AEC58C00E2FCA744A97907765A3834585E4E00B952B5B7B5F10E34F43C8EA3F46C58F8E31B"
01653 "3239C0658F81FC7DD5049FD8D67587C85C78C12CF122CCA327A7860A0E34A34ADAC677F82"
01654 "014EC7217D2710112B6C0067AD402A941877970C95097A47AACB74A8DAE4C9B5F360B755EF"
01655 "C4546D7D63C580F108B197375AC08EBB1F51E8AAC68C39BEE69EB88BFC2558F0A52AE16F5"
01656 "14EC168992775060A5E474E8B2A1180C8F9FD6F3E106963DFA09D48007CC276C1464B00C86"
01657 "CEB1B7372684ED6BFED1DBFA6248F4725E7F8123904BD0552EF70A2E21B8A078CB90EBE75B"
01658 "127A1C887002D70E32CBFB6875138D8A5B312A97AEAC2DAF9D6506FB08FAA9C334FD686F6"
01659 "CE07F48AC70766CFDB001BCDD43421CF70B8B33D22795303769B2C37D25752C9C4494A8E8"
01660 "A0D9A1205A03889946833C065AFECBBEB026E95987C022CFE65DED4F4CB78ABD4DB2CEFC3D"
01661 "69A4303EEA5CB2E8F39A2FCC995DC4BABA644F65411F6809891DF1C0958FB281A7852C8330"
01662 "109BE9B87362F461D3266BA8C2D3E82C9B1400220B19277649CC804A9FDE4C2730AE11CE63"
01663 "3975EB3126E8AF60803DDB5C0258F7974BA7C93499B2C2ED1E71CB64E61854D3FC18633332"
01664 "8D14A12AC06586246EC7621FBF1BCA2578F2E16A83C9A6800A378C8E280C7F3378CC710116"
01665 "81C8848B3A56865F138FE35C123E28F1FD461F60836645A9F9F8CE211A8FD6B07D477674A3"
01666 "4F78E3B916A7AC73A0AD9C8323D0229706A8C2214B5ECE1301E470C37818D68092D5DA3F0F"
01667 "794B8D476878A5B9FB33999F5422EAD28A16D8F8B39CDE00D14B659AD8DC33A46F64DDB8E"
01668 "4F142F8AE8A9F55B94C40D62723CC51EF718C0DFA6D9799AA1863498173A0659F816FDEACF"
01669 "2DDCAC54C1D535E1C0FD9922B333C780C1149C4B294AFC5CD23CA8EFF423A3FEFD0EE1A5B0"
01670 "CB28964B8E80C42AB660665C4B55E06E44557689AC22196CE641644959BC7F80D1AA828DE7"
01671 "5CC21E0A8FD211BCF6185A7DA81222CAE0B632A7F98484ADD6E83A220CD49D5942B7D16309"
01672 "AB1DFB705852A22A4D51EDD96DB3E318CAD7E96D5531193DB24B50D3EBF55DF8214093ED04"
01673 "91B037E1B38FBCECD71199A063ED101C227A10313051F0B4BF4142057A73C491AEDBBE916CE"
01674 "4BC05D1A44FE7D5E7DD76CBD24C13F3758BF7E3BB8139690E33631F180ABA7425DE3E512B5"
01675 "36633F9A8D1B28C0B886C655AD3A03FA8698A5F5AB5484E3CE53CDE1F34A66FCBD734D15BE"
01676 "B727B0CFD768BAE40299C73A8F455ED471C82F6DCE240B3BA4EF6364A56828FE2D6BD8DFF"
01677 "03F35F5C6CBDF0C907F7473F24938016615835D801598409F1004C478BB1B2283AEAB437BE"
01678 "EA2F5FC51D8E43F78C0092ED4D0B08FFBDF1CD60454D3BFE6EA1F41EDC5EEC06357DB41F73"
01679 "CAB23556634416369365D13FC8FF9E20EDD8FDEC673F0BD6C8A1EF60E394B596B70F751EF0"
01680 "D9D7E8D573E3526D2D90325B52A08D25F18DB0E6757EBB6002251809ED47C85D4FA37C5890"
01681 "D60B40BAE3AD22258BBEC4CEAA1B6B585EEF32615F7254268DB82532BACBD9C553624AFEAA6"
01682 "BE2AE9E1401F9BE05D68A881AC8D9E0CD437F6C00", // PRN 46
01683 "7831C1BA1031B6B0E0ECE28BC7E6A8ED16E3068E8F0006DA96B8740114397B19AA19370B1E"
01684 "1F061AE8389C30D0B7099668BE752BCD8B1CCBC497E06BB0F75AF47A36EF8925A59E07011"
01685 "5D3BD63C6D77FE54D037B4A818703F6D78BB37725D7F0AECF6CAF120A80C938FF44BFF1654"
01686 "4C7847B1A989598F1602A220392CAAC2EBC390FD53D15CFD3372426CA41DCF816E29C731FD"
01687 "CA2E84CF90180039FB1AED0DB9630BC28D54201D5606906091A3F867D257C3A0F8193DCCAB"
01688 "2625D40DE58DE651C20F2DBC83FD551EF46C4E724D2C516560749AA89D35C728338E98AB3"
01689 "4BB88F8ACF50A0EDCB70888DCBCCF0B792A5D1F2F6977FA081B4EB08BBD076A43BF1EFCF1"
01690 "C387DD5B92DEB5000B4139938CBOA3B6DFF47B729BB5826C7E188A177CF6DB1B88F051B537"
01691 "DA014B3966452108DF4EFDB0DA597B3A78E9884DE4A7B3D06C259C10B78DEA1DB711CBA8FAA"
01692 "2CDF37EF0378091E408960B011F4E4FE24771D0FF3B8DD01DB0527087AC5C0F64BE693D38E"
```

01693 "0A128FBCB8053E811602A05F3DADF2166CE49EBA13F382E90EF4A143B2CF4FC12DD4873A40"  
01694 "FF7354C6E648A56623457665402632C41942D23001C08470AAF1F3FF0A474AC6586C76915F"  
01695 "C91AF074103A1CB9304485F92F9639399028EBBDF9394B4BACD6BD3B2177368B94F639AA8"  
01696 "D202D6BCA88D74AF0610C7B565067ABDE4F2BFCF8CFFAAF93274CBC913EE7DC5C77BBE978A"  
01697 "B9F82B546426DB933D67E0C6748F943F39B8FD6791847C50A5A834A78C2FBE5D21448A035D"  
01698 "83D45F9ECF44C21C4DE726AE8111CF277049331D78B06E9D7D3D0D8915C7C32CD6954D3704"  
01699 "5E7AFF8AB496A4827574C457523AD6B85F78641BCC4C9535D29B3F8AB4D944125E6E54B3B3"  
01700 "F23EBC04BAE3AB102977EF829337F51CD833B4F5D4559D79E695470F498EB4388E573D004D"  
01701 "725FE41AB1B39F9526867BF7B99301E0E50AA569B3B23EF78D02FA53C4BB2A184B02977F2D"  
01702 "2AB3B71DB09E3E8D9D5B13DB8AEB0A95F9296B7C49B1B3D6903BEEB1B9792B5ACC1A5D1167"  
01703 "2CA5D8E3825D044DA986D01F6E1E292D74443583E49CEFBFA177BE275977BB4A4F77ECD6FE"  
01704 "DC886B4B83AEF5F34AF1A78DDF6B11C201760175D57823DEE0FA5C6A9C1A91638561B857E9"  
01705 "93779B6B0576994CE86E7DCCA4A89BA2ED84ADD1D8B4E11446C41DD4B1EA71352E26B2AD4B"  
01706 "A1A1C5C64822CE21BA4B8D7D6589FCFA906768A2BA0C5AD1758AE7B8A2BD757C8E459814CBD"  
01707 "D6375DC58497279C0C327F67D14FFA5C90167941CF9AC642B1693D7DE52578AAA56E0EEDF"  
01708 "A4CC12C7469DBFAEAF5F9D3D72FF103918B6F6DF7E0DDACB208A8764B459EB01092A101D42"  
01709 "E762FD5C5AB381AEA19BA6EB8CC9CE138892FEDDCBDBF818E97EA8AE9846C0CDABE9B4FD5D"  
01710 "80431FEO3A4CBDA566314BD1AF59AF120EEE4105E1B701A4058F1112F2A5C8D9F68EECC870"  
01711 "DBE78D51F89AFACAEA3A2A068C59654EF5E5BE2F6727136908D993434E5EB1B84331119DB"  
01712 "1C02547EAF7C9DD479C5B326CEA6224B16A2E108E76CDF0C54D16E472D0CBC80DAE7980713"  
01713 "602FDA31A6804E4FECCECD96AA3901D7B35D445BDBA020DDF1BE5E9914ACE600ED24E937A"  
01714 "8F82AE583A2EA454EE7548882C1A192F11345A3F5A3F2CF64B85DF6306F7FD396C4E773FA6"  
01715 "6CC098EDB3CF855599C2EA13A6F726161BB0C0D6673F02E3DB24FF75A5698E91C1522424D4"  
01716 "43165B5C90FFFCFCCE6C42656138BD85409C45D8AC453D6742B42A43A7626E70C212B7318"  
01717 "834B7D29CAD0C29F22E6AC3A59359B1A75F727E1DC", // PRN 47  
01718 "30B93A783620E30E94D47590AA38B10FF12C3E421887E9081F9A9C5BA1D753F341C33B7322"  
01719 "187317C6610FEFBD0382DBDC2D863C48F15AC3FA86B3C5A5B9EDB937AE961E8EBC3D880CC79"  
01720 "141B79869F95D6E90C5C5E7D9448CDACCBDE184F6FB545324995184770CD2838B61376B148"  
01721 "ADF3A42123AEFCF8890CAC7D3AA87A624DF3B883E86AA5FE6DB385778F50BD4E528C6604CF"  
01722 "AE0839BD949118EEA3972327EA355EFB98E9DB563FF72C5B860103AD262BF34BE17295E37A"  
01723 "81D1C98EE6FD6690914467048A50AF09435A68FF9A70B59DF8FA7A543BBA24EF8E13AA509C"  
01724 "7A311E12773405A4024A0C2D7C2802025FC4E8F8C12E4B60543001971FD20F1641B968CAC2"  
01725 "83BCF8140629DA13D1289B7130F3714B8318518770EE1399CECF1D3074A65F891359692B"  
01726 "269529B5FC63D824AC7B6390580C516BDA78BB1F158B2C69AD2FC1BC610AC24DA696F44B58"  
01727 "DF813D9DA52529D325BF2B3575768DB43AC3BE5D0402C717FB587B6976E6701B0A70D04F32"  
01728 "50B6C54E99983E2B1057DFD9EB7B621504A1004E7E67B33B138628EC7C43A199D00D9FAB8B"  
01729 "1990A5FF29E4399051D1C62AD0D86B651E9DD94D27DD6A7B09AFBF1EC737B161A4A398F7F4"  
01730 "24889C0BE6FBF8EBAE9F1A835FCF4AFD45FB36919D362CF73F0786D93552A7B4E421EFF9D6"  
01731 "F9807885098DC99A1009D6551F02B60E12BC28499A27FD7DEBE1047CD44EFC0D90813233D"  
01732 "0886DDE79F8AE34136B04787F6A5918F5AEE705CFDD5071ACF8FF54A232FE39E1D79D82EF"  
01733 "FCB6CE1B68A17B4585817CC9AC33FD6CA21A00929F1EE5B9E3518B27060867624D8262EA18"  
01734 "F1B01A525EB6CB337B257C8855FCAB72D7A83DE52E4F1D5E29EA697930C25109182B95A202"  
01735 "E519B3757092239BFC633C4A7099290FF7B99FEBF81DF5AC41366D3DAC3384C65AD0EC84F0"  
01736 "DA4A0ABF7D8E1E2F8B898FC97325878D911512F1FA5C24F009856D44123EE49C77370F1D58"  
01737 "E84765428BA0E45CD852D19EE00025A8BCDC011BD259AD7360D12B6A630BFEC8A97AC407"  
01738 "81694BD05B3D50D9E843FFAF7A023D451E282FAFFBC38BFDAFEDB34528DC792ACD499179AC"  
01739 "D26AA3765523E91A137F52BC959D5969B301B1893564AB348606864916C8D79DE6FFBDB0C2"  
01740 "ECEB3B0A3B8B46FC37CD7D42A37B2B841CFF9820DBA44FBO90B430D9714FE2ED731F097FD3"  
01741 "0C2ECDAB7B6A276990647741496729219F244FBC6F1CE02F24F98D0212AD005E54F242A7785"  
01742 "8622D6BA961190F97438FC4B33217A84B5EA9D3471C27D08C2D52DF19F2DE486C9A5EDBA2A"  
01743 "56E1349C3BFD312B8927A9E86A97830DD0B11F505AC481FC6834F8FF7D0E67126033502EB5"  
01744 "C593D15E1A56DCB03ACA1A5DBDBD6E6FA9DD3D7D152E8FC1741BD876760F99CE780EA5BDC0"  
01745 "18022397587357614ED37266C69AFE4644D7D5259770B85B70E7E395D2ABCA3FAEDD5AB3C2"  
01746 "4B61117BBF707A58530EC435D36370D6AD8F9DA503F8994F6CD627336A731BB149D3D3E960"  
01747 "75385D8FCEB3D9B48757A79C67C0E0C701A2CEFA092D5462E28D12548FA041C5407AC321FF"  
01748 "049B37225742345E065720A1098B5785AF12927167ADD2F72F766292F421C290C2648E270"  
01749 "C34443ABE5B95B7E635B4CBEFF74478F65C77899418C8ABEA6A4C3FE217D784823CB448A7E"  
01750 "C0739C7166F4AF733BE758B6D5011EAA9B05ED995DC70E3D5A2D2298801C25049661FCF4CB"  
01751 "2E9F7501C86D763CECADB0C86B348C7BF31DF7BB0CEFB4A2EE6861D3D9D441AFF4CDD8B9794"  
01752 "65C1ED95692C5A46F5ABEC180ACDA140AD11BCD5B8", // PRN 48  
01753 "84D5B404488AFCC35CFC6EF1CF7A848EE5EE527B7D2D2081EBD8FA8432414FCCB271A43FD6"  
01754 "18EF95CAF1CEF445B50F02AFDC6B41D5C1F1FF4B6B8F2AB94F90F259383F44246B2D400C30"  
01755 "12FD76E2980827C9B5476C4651293CBE9FAA77EFC53E369B94F51BE480EEAA389D88AE421E"  
01756 "75EC67C76E30A6FA7EF02500372E4C9AB875EBA1C357C003938B979859B0A914D58F21B3BF"  
01757 "1BE0C33A1AF079C5A0A4EA532F93A10F6D96D0A63F3C99E6F06CB0C72CC5D71A517FF440BC"  
01758 "260A26E3A46688E79E733E1E2F0010846937689E2998BC769186E0B977AE0A168925D255A"  
01759 "B7D5F34715AD526E824838B62546D9569009A191BA5192535E2314223F6A0435C90D1405C85"  
01760 "30A57B402118FDD2A5DA4B40635D2F95AC7CEAA370686F0AB23AB31DED369BEFB066909B7"  
01761 "6F2C04039E49FF0CFD323628D03C26F0BD4054C127F21355C33B49F86895D5354C3BBD2C0D"  
01762 "38ECD169289F7D8403AA1753B393AC06C887F98315324DBD78F9BFD1AD8880D41D4EA98956"  
01763 "EC89BBDAD08F7CFEBF2698EB5AB01D4E49AA8273CAB6AFDFA1CC1C95EDF85284E6570332A"  
01764 "0B8242FBA495D87A01A945E61CD36E865CD4F430F04A3B3EE74CE0486B5676F7A938B1E1D"  
01765 "B2DAE2FE1C04B3D56892C7394D0B07C5A2F7698D196711DC83FFF6ABDD4E8131E1106EBA23"  
01766 "245FB0E2A696EA40FB6B0B1CB4DD22C9720AC76F09863711647A7E369FDE5D6911A5FC9F87"  
01767 "470B8C06C4EF7063B08A8320B4371866789147E9D7C5D477A662A501627F91E95C6E2CB681"  
01768 "4651E9DFACD67CA0B1FE62F44081E3BDD8B964C1E5ADDC093AD185F0203B4C0F7E3E96811F1"  
01769 "4C79BC54D2919FCE5B653F6845CF1AC34DA9E4CF52EBA626B739246E25204F9EF2CCF4F5AD"  
01770 "BFC53CD8F629FDB82186CB52EBE7136F105D3CFF9C0002610BF2C8F943DF5BD03877ED7AFA"  
01771 "25CA81C4063E1B97452A937FCF77F1AA068FE2A2676ECDF662ACDF18391AF7D627C2AA07CF"  
01772 "5597B5F33AA0ED4AC98AA73C8321EBF6918EE568ADEB1A823527AD75CD5BF7ED3EB5633A4"  
01773 "EDC8C59DB0A91DA79C248D6894CAF9CB411F302635BEEF1A1C7916C7545B8BE7C577692245"  
01774 "E285D32B4FD18E1478F88F8373E22A5CEC6D22EACAD2A41612F05AB2C54A1C03C0512359B0"  
01775 "F9C5F91615EF2EA80DE97230480F489FEDA38D52FC84E2F1258CB20FF0E850639B3C12958B"  
01776 "A6C064D0D0F4AAEFF313E43C65EFFFB47CF09F2A122D15F76E8B704DA8F2B8C71449AA497"  
01777 "74EBB7B2CD97CF6004D2FD6B37B9C689261189B85CFD50C20C961F22A644F5149737742655"  
01778 "6B956DB8C899045B175A8B9CB22EFDE535CF487E9D958281E0467539853054FC3475D2E142"  
01779 "D58F9201C1DC6B2C1A12B2878F9366E2216405069B7D03852AB37EE83112EF2EE172F4E531"

```
01780 "7A41653A656A06F6633AF59062A47263A9977A50B9115456A58F9C3424FE7E0CC57DA70FEE"
01781 "0411C1479B4CC2E6E720B01432CFB3C503983B37FFBE38C92E50D6C5795443F73244F6D28A"
01782 "E6270E27D46A25621F86A2BC4259DE6C04CCAA657FE656C640833B290F00A1660329A3F09F"
01783 "FE60152A23925F4D8B0933AC016B5802CA863F66F8CEB5C8383B1180A515AE6C51B5F56597"
01784 "C3A004F8D25F8A6235C97B4D638468B63E3B859A487AFC4320598EB6D143E5C914409D25F6"
01785 "DFF6E957D575C71D3ECBC6743CC03E5F34791744637A994F3ADF86966B4FE911C06F7AB1"
01786 "A6C20A2971D82B64E49721A530D9D5A2319D5BE0F47B8119C5A835CABEC735A935E23CB439"
01787 "970437C6F4CCDE2525EF0D7B1555CBE70280D5E760", // PRN 49
01788 "A5029C9EB4623226D321FF78D3C4EAB1F672A8B2B24A09CAEF21F561A851323C05A3C5E136"
01789 "A2DA7104ED19BFFFCAC3FC49B6D598F3060E93552EA6C700B837F7CE04721919B9C96A57B4"
01790 "2AA1D832307C7A847091848066CB84947BE5F6B54654479E39F654819D3EF7AF4939FB9F4C"
01791 "9B20CA7F83DA0FDA2F171FDB72455B7ED0D43206992520CDD86B29C48BCC687573AAC4217D"
01792 "0B7DCF852811ADC3ECADD2B34B4572675CCAEDFAA01F83561E6E240878F229698185A80E"
01793 "FEECE8E9455A72A377C24CE3FECAEC2A34B9CA98D288596D1C769CDEAB06871316CA7D1DDC"
01794 "5862E6282DCE33F362C64A73E57AB266715068932ECC31E62AD2ECFF7C6FBFE213384DA08"
01795 "6ADF49E30F432EE2C715D9AC4DEF53A7B09B0D722CAC560FE8CF0059B80428458282F7E817"
01796 "17647D7E321A3E4BEF16FAEF76009BD98B8D9822B771EB62F1D0748E462FB7F3BEE9B12AA"
01797 "86D92629085EDF8E43E1252EE59970CFF66A6F865C7651EF83F8FE10EB0E2615BF8F5C7F12"
01798 "FC601B0CF795C0D8B7057F54408BEFF86747C0F6F23EF212A9086EADD464A25341AB71FBA4"
01799 "ADED8F599C38FC15E790A5B86E64977C5AC718DD0B47C1A476AC9D7369396144F6288E84F7"
01800 "FFB7E02EC00EEAE8ED415C84648364ECBEC42164514D3E26BFD3187E0641C216FFC57E00DD"
01801 "752CDA581686916221DCD1AF07582391C5FBF047FD1B7B956B458DE925C02A756FE197233"
01802 "E0304D0E034FF9A176B5B3F5FB683AB41D2691E13F97B3F4EB33238851331197C49C602332"
01803 "32DA0E2610430461876FF6F77FF3CCAF1BB2424B8B347588667B48480476D40BD9E487468C"
01804 "D5AFEB597C750A5E665B4E7C4C169ED08ADF731FEA928052C4F8B5B3064EC07B0CB988E32"
01805 "4893B308429D1D964403F0350B7E1B06DFB73362C38318B762A972972BFB76CC5C08B5D47D"
01806 "BA0F3A2473D7749DE9F49F50C4C1620A9EE9FE5629612BD72906497411DB87D4D8EC4E1F79"
01807 "BEF27232008A2299F5317FC1A6F455F1B827F1712BC01814F0B9D0CC162B25B804278B9C7B"
01808 "C5FC5616B317F2050234A7AF92FE35A59E22C959C7163DFA5F142022BE5CC4D5EF16D21821"
01809 "6F57C2E29DA926436C00DCB82E68E16CA5A07158DB8864D38A765D14E82175114A28CD97D"
01810 "11D564C8C7B87411589A4FBD49F990D08939B7A73B5E6466B6F607F8AD22120A559A02BFC"
01811 "EF6456E7AECE8C9B7C9D2B322D2197124C05363B2CBFA58B74CD88877F22A5E5C202FC2C33"
01812 "531125F1518F40F38FA788E5E6B3307A75EC73E545391CEA200243DD6D25A5B8654A00B82"
01813 "BA57437BF0ACC80ED37ED2FED221E54EC12B93AFA6E3939223596075F4C47340355D722A8"
01814 "234A1F65EDDA42FFF5D19F7FACBF09AA77E7962F4CFBF61A0F26FB18E31A504B3717140488"
01815 "74BBE286AFF71B43E4739A17E8AC25FA77121ABBE6E99754AF42F1D0021EA1E3FF088D0734"
01816 "BB191F91A520C96E22B4A28F9A2BD7DF81E8079EE5D0DDCDB517046F12098FAF6920E0EBA1"
01817 "0DE8CFB391C63C60D62C1F4BB26BF8B6E421A830575731F67D306CEB5D6FF04637144790EC"
01818 "4AA2F435906320114CB81EB40C22B271FBB065474687AA5880F1DBAAA1744AB3E9B831A932"
01819 "A9208BEA9F5D526C52F5FDA56320E123CFB553E2B71A595DDED2ECBBDE890B0421D765D2E"
01820 "9FD0D3995DF2A9523A65FE2040710DF16F2A83F510DCA08493DC138541E5681B51EE87D84C"
01821 "9AC11612EB5C06F5A63E22BD6275E35216766D79B215DBD087E9CADA0CEB09BFE435DF9B78"
01822 "09A76DE323B373682B8C58CB4F08D9C708EB050DEC", // PRN 50
01823 };
01824
01825 // Galileo E5a-Q primary codes
01826 constexpr size_t GALILEO_E5A_Q_PRIMARY_CODE_STR_LENGTH = 2558;
01827 constexpr char GALILEO_E5A_Q_PRIMARY_CODE[GALILEO_E5A_NUMBER_OF_CODES][2559] = {
01828 "515537AD5E5F4216C16046FB0AC50DCDBE5CEE7E3CBB51B6ABB4E87A407B90E0EFD49DE1DE"
01829 "5ED29184E7FF0DC31F75FBB94F46FF6586B36C7771E5A68D060A965ACCF8D640C6B6E4530F"
01830 "DF19DD2491BCAB69ACBCFD3EC7281CCC31253A471B652E21C4CB0B43613EC542266460F0A6"
01831 "1998436BEFD95572DEBEE920A915FD854D17FFD0DF8C74E23B21B28493A0927709709B07C6"
01832 "5878C43B69DC501E9D0AA21061ECF173876CAE708C764435832D9D6FCFE62DDF2543016D63"
01833 "2A56D9BFF1007886E62E8A832C632063CB0717D723C5E8C5F0C0EB3960577D364C93060B64"
01834 "EE04A539B7601CC3113E0AEC53CF21AFAD0154DC5CCECF038474E0F4004A65B1EE2801F819"
01835 "68B88C3D35E87CBB126C02D770CC3D32A552883D351DEF47847391484F80646728221F9939"
01836 "21BFC1241E3E3D9527DE607152724C6D2DD305D3FEA0AAAEFD6509A2FE3248494A54FDA8E3"
01837 "CE7E6BBCE234E4686BA5A19724BA2CB78CFE71A6AF45532EFB286C5BB47BC3C1EEF4E4A8C7"
01838 "57786AE974F30A86CD60EBCBDF5502AA8F643819CBA4301E731ADBA1345B61C0B444FE7B8"
01839 "17EA86F8DD749C451AE7D24A68D914F26C918238953E8AE61CC8553213DD6856C7863F9F6B"
01840 "AB1B4C84B225911E7B92BFFC12AC211B2B2CD905877FE976E07057963D47C437FE47D89648"
01841 "053F81AC39E8FD2F3A726866F6693E503CB6F0C3F0AA9B3EE2EA3BCDB16D726E1C6D8B073A"
01842 "A15F64EB68D53B1F8CDAC19C7A33361226E81F1C793BF188755A3FE1BAC38B91ABBD4F077"
01843 "F7A28983EAFADC346CB941D49492625893453B364D07FE06FE42B160C16FE0462AB6366FFD"
01844 "EE54DC9CE4DCA21E4E4AE5E92C872D1E4EC6FF6D3063C98A5AA5EE72481A0BDF15152E2A5"
01845 "425AB722101474D0E1EC8401273EA1BE1DAF7403190A94305BD1C7DFBE1F35F65D5CB97E82"
01846 "B7A297047507FFA0012FB73360FB8719C174E78A989A96E60A9184B3F3A8188DE100AB3619"
01847 "21D38E8142859C8F0F7D441DB1B2E9687BBD1086643987C83DEE0BE8CED4C83BCC82B62B45"
01848 "311CE4F13ABC55BF5EB1ECDF15F5A07F8B2C42F07FACE0E299E87727E2D534FEBF7B9C3894"
01849 "CC32E4E127A294B9FA2A671273B174DBB81D247CD2846116500A072DC3962C65FFECDC0C0B4"
01850 "6DC2AF52882058259C26FDE50EB319AECEFA1FABA34C069680B9EBA9D96EEBD7EA30E748"
01851 "213E1283396A2AFC63527624641D4E1F1022A973B1898BD4CEF4D712B49371A51D60E08F42"
01852 "ED1EA90AC49EEFBCC53E7F9E899DD1AA0456F11462DF1A4C81620A73C831CEB897430A2225"
01853 "2B901EC3D6F3DF58FE26422F796EA31AA4E0E9CE5B4A9C312A22305E298FEB3B3628283F73"
01854 "5EDF726937327D90C542434BA3B06084584A9DB244839D2ACBCD7EF147A541E35687B5B8F5"
01855 "F07764973112D20D1ED75DC31F6A938542B42EFAAE0F11B0583AA4925C3132356200E8D6B"
01856 "DB3127B975F4115A7A8A1C471836E3C5450B501A24D4A1308BB319AA827222B550F253F64B"
01857 "6F7D2322C6A2D3012FEC265A66A60102A3340CBDA9B00DFDB36693D41DAD8DDB8875F8C3BE"
01858 "76AD5355DD81D67AAEBFFFE945E522BE0312E60F63DD92F25C0D7CF82F223AEC0BD745675"
01859 "2CDB1511FEB5368F8857EAFAA90E8C7499B75D46EC4CA20BA8A24C90C016B5BD2CD7864828"
01860 "C6140E98EBD9509AD1194F56D49675D077DE92CD481B469E3A37F7DF0D5392DA4CE4CB2825"
01861 "30F1C73482C09268877B0B0CE49FAD21E4C26194CE950E0078F3854EF88755E08E93801"
01862 "65C584ACDBF1ECEFA31B224FC321326B93797BFE8", // PRN 01
01863 "D67539AFB80711A0BA3CD67D963BAD346BA813D35A2EEA104D36AAAB863C656A07AD61BA60"
01864 "598C0774D432ED01EFC9E28346C09EAAC2D392E5655F0FEB3486815AE30A38014DD8520F73"
01865 "CCB871D9D326328A50998A2A3BED34E4B34D0DCD65B94807064E2EFC0420898DB96E3B99E"
01866 "A9A0AD91C63857DEAEDDA5E644E62212B23D72FEDBBAA78C6581C677B10689C4AF387626DA"
```



01867 "C55F4EBE1893D52D28D20EA365702448A64A0C553ED337C3BB911DDAE2A91727299D8064BE"  
01868 "C880183064574B5E3631E70A0590210143F4079C572BD5E2F7634C2D53B1FB1DABE79C4847"  
01869 "99E7075EFF98F033F5B2EC66C373825335D883911CEC9CFBEE2E38129B7E03D9646A7E513D"  
01870 "5069043BC62AF4C524F12D8F98D8C9DAC5D8642DFFF48CF6737AFBBCBE965925F55F03BBD5"  
01871 "123C9DB47AAC780301DE91FBED3C01D03E6464C2C2915BFA187A4BD93E20C24574FB91F035"  
01872 "C8BE0921DCC8D6B7E9976763A1D2158511861EFF5D1C0B71F608E7ADE91D9DCFF5640B55BC"  
01873 "9BD1BB322C879C7EB5C06EB2601D06241D09CFB1BC1695DFA55FA044E0E2E4BC86EFB6A55"  
01874 "740C4640512DDD6CA069940BC0FE1738FD376C68BA8AF7CCBA7D89F7966B29355538836372"  
01875 "EF418D149EBF3AC104919D91BCA2F13E79A7CF7684A4DC0AC556A2843E041A71F97C94B859"  
01876 "FE009659F593EFFEBAA6F6C1C57A5BF22752613AFB26379C42AC25804AEDAE22D63B230FCB"  
01877 "858F496B8EA6F37104D0890525DBDE06AD988BA0287B0938572F14A98EC9E60E973FD693DB"  
01878 "1F2AFF671AB03FB12B729D0867938DF6B60EC69790C992C6C3A531FE56D0ECC1465F65E3E"  
01879 "57FC9E45F0F65A1061CE6D3190B6C1B8708A8A5A47222369AE889D26499CDE8F0548B8D707"  
01880 "1F2D4DF6C0F2418BE449552327981CB0B54F792F29A71E30DE257CE1B3A7553A22275E4C8B"  
01881 "9FAC3B8D4E9912BB22B0A899E7337513C7ACBEDA15FAD3D6919FDD941CA659D78B74FD39E"  
01882 "2E2F622691B89CA82FFE602DB2578A20D4665184456F32DF4DD6CBC412EC7C6914CA427CEE"  
01883 "02F6D9810AAE1406DD68ED3869BD8E3947A2B3A803E875FF82005D853E3F43A6BF93603074"  
01884 "4C34C8B71B7722BE0AD3F475E531C9249A46271D5F3A5C77C4C28DE29AE953EBFB5725781B"  
01885 "7B636F8365FC755C22871E7D53A1F1561C92909305C9FD36AAF79E8844B63370B800B25CC1"  
01886 "355211D9919B830A988926829F808DD2C66400279E6AC14F8EECBEE8B6E9ABFA3BDBB38A49"  
01887 "535F64F719EF48C5FE6B2738DC6F71AACD70274FD40A29BFCAA594AC3E7D0C3C522E406BA6"  
01888 "392444C9F362339E8FF34BE330911DC7EB11A47FF3A62A46CDE961A40CD5B24020909E5B03"  
01889 "4F45FE96CB156FFE8E2FDBC12A7C12D60D24BFB5E96544E4F03AF26F086A5A667496B7DF30"  
01890 "2E4DCFC568C7ABD665EA7EAD8A7F5A000DAD9F43E68C4D8A14742E050769B3CB270E3856D7"  
01891 "E8F4E827046D3E55A52F0E02C883881914DE87AE3C24D93E61A94919B40398D3EABB1B5142"  
01892 "431AB919208A9785962D05061EDC951C83C73FBF6AE8DD6FF839E631C9FCF6635FA053DBCF"  
01893 "932E359F83FBB3EE310281569741E3A8975FAAD1E573E0EC3207F6DBCDB8CC90DCA1CE157C"  
01894 "8DFCD131AE4841F87A5157792738DAEE7C29240DAE26C6C3E9D8A899D2271B0C374BD2EE84"  
01895 "6C6625E31B2F8379A96F601323039D281608A01A1844E5D1D20CF1D92C52107CDA71BB3B7E"  
01896 "C4DEEA958CFD7A79F71868116CD1DD2B2E66BB94D137B3733F324BE489DA51BE72B0A55728"  
01897 "80AE1E61650B839D03B0192D290D5B36A9CF93B304", // PRN 02  
01898 "58B2E58A13D4D5F84F37A389E6B01DFA66DBC6A25338B2884EF08056BF0A9124DA29254AA7"  
01899 "9CDF1B0944DDED4BC7FE683EF7A3C7A1C359E61E959471E30F9D534F43EEF274AC6535C616"  
01900 "ADC7455FBFF43ABF268F7C995CE020CEC73BDBD04007562F2710498AD1A324F25A6233B2D"  
01901 "F2D9A429F1C39943E45F934986D1979D293F90CFBEBE01665C98C0D72D09A382348136980B"  
01902 "31F5D4696B85C3F42D5C445996804159CD4C4CE7547C4A3FB718AB62D9E9826D05C44C6BD2"  
01903 "1D708CABB01F46514F29FBE7352866EDBFABD0ADB6093E1A519D0E11E27737239A0786675"  
01904 "2B6D864686AF1E308481C53C635046C0756E8008D3CD26DB970C5D6EF8CB9DE6BC4339EE9F"  
01905 "88EA45D11B74A1525E2F1B9F91567A78E403F7ECD47A99E95C098870B928E1B0DAA984CCEC"  
01906 "A13C6857F350E808BBF66AB4EDFB0F369F6F009268D19DEEB377F7D0C4ED6414EF6D23A0A5"  
01907 "F37A25AAD3CD62C2115CB70409049CF0CE6D4DBCC5C96767614C6FAF73E9A76C7550BB970A"  
01908 "B3D3ACB0C07FAA9F36F1935DFCD5228427BBEB5FC7A2F951D9210A5BC32ACCA7E78AFD5FD4"  
01909 "60CE2A79348714E5542BF2D453680B070E85F3244E8A6B4EB6ED49F803E60B8A383C8BE628"  
01910 "3B1C4DF79E0C4A23D5DAFCB26DFC5DF1D1502FB6CDA23A05A1513F751EBD0143C2B9F5494B"  
01911 "AA74F9P95F189C5767C6F562359D9F20CC9B9C9ADFE285FA4E1BBD4A81A52ABAB36A53933"  
01912 "70091A49A59B968CFAE89BEAF33BCA5C2A9BCC1758CEFFC2D0A175C8A5CAD5C7E3FA706C2F"  
01913 "EA55F4971A7B8A0C5D1F22A26D7DA9A70603AC34566E4929AE78C8F1926239950EC0A6E1B3"  
01914 "7D919E2431E53FB08B2F7DCDCA2EF4177BD7D2F81BFC784FBEA0471831CDCBB9E11D60C53"  
01915 "D0E062E8BB8D98B8E5A40960EBE10FA642751B96E1407B38A024DC64D5C2005E71198EF39"  
01916 "4673A4A0097187D2475CF27B3EAA7955AD9F412A89B36AFB27FDF7FE699B2CC8C03F7FA405"  
01917 "49178DF8A3C1A39441EAE869BBC89D583ECB18E310885F33B95B719045CA6955720CA75D5C"  
01918 "F0B29E5F1B9E9AC5EC92980B2D37EF6509CCA6E67544A8116494AC7EB9032E1E8CDB053B5"  
01919 "AC0F60DE59BBFB78E3491D50C7ADED95042A8885FE284E90C04FFC0370B92B68379B2E7D52"  
01920 "392A42FEB26B2419CC64412BFFC036C01109B9EAECSFAF485F0D61C37A703EAD02127611F9"  
01921 "D25E4EC515CABE21247156F0779CAB57C35581646BB71E87941AB5D06FB3C06C423E9E83E0"  
01922 "C07E611C89CB000344AF49498EF5C30305DDC8958F45A45E1218837269DEEDBB0AE51D2AD8"  
01923 "B41BE425EB7AB798856911B6F0DC721756A8154B9D1CCD61092D16A3E9E1CD5E1D7C2C45D2"  
01924 "36C9B21A0ED64AD4C0660FE42FC0B543A34FB545B0D88D42F696D53BF54A1A259486846E81"  
01925 "B44C16EF146310AEFB3933BA9F9D4497E74AFE1449D69105B9F295DB24B4563C3FC0166CB7"  
01926 "F104B3360C0998FEFEBABC6276F570F7679865036DB59B51AD2F91C0CCD3BCF5C07492765"  
01927 "D480494AB5887E83FD04B75A35F88656D3329AA5FD50DBF493B0CD2C19DEBE4358661C72A"  
01928 "71FB17C812F75BEB6130AAAA1F757288C1D8626461F4162A2EC6296B3CCABE21B4F0695F2"  
01929 "D2CB02B86035251FC9C5F2FB5BBF28237417DC56471629B3B4C25AFEDC49CD108ADAD9DB1"  
01930 "330AB680E4998C5B3D99BCE8F08156B630A63EF36A51C813CE22A6B178683EDA389ACF5865"  
01931 "3582BC2A710ADE4BE03C04CBA603C0E4572A74BD9599A8442E894EB504F7703F2353A8A"  
01932 "3525EC5535E750303AC07E08D952755F97AF236014", // PRN 03  
01933 "3059141DB31B84555DDC1C5F40372C50BEF6E82B433D87603BB7803FDE8FFD74DE06809357"  
01934 "F11C1D6ABF0240DB8EBE1D5ABE7F30C91A451EE3C1AA36735FAB9185C785F05D28BD470D43"  
01935 "3ED4ADD5F6406487CA1710ACAB57EADDD577A49C841F5E78B70FDE90BC1087E79E27E53A4B"  
01936 "131C06BF33843BFEEDC0C2A207510590E52158C0855C40146AFCB75DEEB872ADC5E77CD1D"  
01937 "D6A66EA96CC223B295C13E2766ECE9F2CB807C68C84FD04E0A5179D58473860962083DA8"  
01938 "D68B70DB95C0EB91679CF5E1BA468786725EDFB179A2C5E7C28B396A53D89A98C7B79867E2"  
01939 "40176A9DF9FB0BECC22A526B617CB8FC2B05A8DE411C8C5F9BE682459B487B26643FA894F"  
01940 "9B3C97282EAE397A03DD15F07BDF7432DD95D29E6D50CE95860BEC2649E482F40F7B8A13"  
01941 "135A7C71D83ABA44A363C864BCD78050AC69A8DC0A9E601859087DBD49AA01A75053792F74"  
01942 "D9A8EE726608D1F009D06363F6E5A463BC362178918737C5F3E71EBF130691A2048F07F3F8"  
01943 "CDAB70C9D0F7B9A1180046076E1B894AFC620244B30A571DF359C7D60969A9436F05417DB5"  
01944 "759B2D3ABDF6A238B5B5DCA8A0E4C27F7078DBB6600FC0105463E745A321884A4F6C5963B1"  
01945 "88919E2C24C460F1B9CB9C063B2CD1FD5F49C06AC2E61EEA55056A6ED08B6C7C750E30EE665"  
01946 "08FF243FE1B5C494E8EAE5467AA0ADEA834523C536855D3BF5ACFFFC0802365B3889C3191"  
01947 "0A5E63B9457E46B4B7E29189F1BC21E7950BF9FB44BF5D76078D8602FFD986E1A1F1A74C67"  
01948 "7BD8811FC6992F39384CB3F9B2C91F381DC01D4E79D2E66E6D76796728E0C48F71817ED182"  
01949 "759B810F7B5D361D7138CAFBECA3234B9F2F4A76FB00DEC05D67FA8E7217E488688CF87D5"  
01950 "9AB4FE7B13C793C1A1711C401B49C381F76B5CE120BDFDF30A858DE62A3F4331E4A3CC1670"  
01951 "30829906935FAC167A7327B5FC0527E014801A27FE0F2D40EC13C66235EFD070979576DA7"  
01952 "268D1871E8487C4DFFC62D27B83488A73FDDDE1470832EAEFF621CE8E126267A47ECO07739"  
01953 "22BE480C7F26C8E4785D886BF8E6518DF13DA86B2925A432D479E8B8FA9F92442891F71AC"

01954 "7C05C0BB0E4396158A492BD49E42AB4C4737A952D6B2B2780EAA73DEE13037D659A43DE1A8"  
01955 "F0E525D7F96FDD1629FE51560F77E4A1CFE903DF0891B39F50B71E0F84965413DB34B1EE95"  
01956 "CC1B6728AC3D1924B34D1BAD4A4E4F7D98D1B938C567B6F3C95770068775F78EFA147ACC61"  
01957 "9AFD0B6D8C66034C21B2991419217256D88F265922FD49B4BB38D2955305A8D01D01758B01"  
01958 "134196C525315726D9BB6D77917A57F2C46A05DE91A3FF80DD2B5AAAD899D679EB55341FAC"  
01959 "6D5C2E9C6B703806E342B2C11CF84B394BFDCA378B26C7E0CD23C8A87B7CBEF6828D9E023"  
01960 "B2B7060A0FEE0D29CE59B379A02AAB7DFCC41A35CB9A94408A321BF78A54D2D4DB0E17010"  
01961 "1798F6622D8087A2022A993E85DEB5CBF0F4DC2A047877EEC9DD145471D57015DD59A37BA6"  
01962 "C0C9BE3A4AFB92289562601BCE8102C83292E2ED842470D715D34F11981BCCB81ACB443FA7"  
01963 "32F792C5F11B7DA5D1BCDD84EB80820EB2BA3813F5E6EC5300E622DD81B1FDE8C786387864"  
01964 "D78246A432CE245E0883745D9A1DE9D68253CCBFE7C00EA908E452DD3B9669F1E6812E40D9"  
01965 "E2C423180BCA98F4591307CF8B8E98B7F828989CA7F3F0BEEA4D408898364C6DC160C94B89"  
01966 "D879B1D07286EAA3FFBC6CD7FFE5700304AF3FC049098590DD7F6D4770ED66FF60DB2909C2"  
01967 "DC6DC67ACFDB7CDEEF1B714BFE4CA04F2D5172137C", // PRN 04  
01968 "4427106DF31EF72E43B6C75CB84BE5E375B6B6D4D2122D0FDD689E8F14EBB81372B93F5455"  
01969 "ACE6C168345234B378500BE6612FC10536607E85B884AEC780550F2A26FBBF0445E0AE9C0F4"  
01970 "CB95F01BBC76652A9E6FC457D6BA425EA8B9457A6F3C0AD2A0FE7BDD1C1CB19A9F1A8815D8"  
01971 "4EB5843ACAC4DD005378111E9AFBA9AC3CFD808C0C3CF2B39B1962EA44B0848452778168F7"  
01972 "60034A68E5980EB63B94E50D170C680FD6345F12CF30E9089A7C0A422661DB7B1BCF248023"  
01973 "8AD043FA1DD0DADEE998F7CD69FF04BC336BBD5985315D4E1CC78558D4E235CD7E05015A573"  
01974 "4896597F1665E45E0C819CB0B895A38883ABE4F439D10195D88366CD14F45947716AAF77"  
01975 "C0C27FAF6374CDEB53B118277906F7E61C583E8C7CE3FE733541882C511936076DBC6F65094E"  
01976 "37FE2FF54E0977595077EA26E03711FE9054E2D31AE7F99939B415F46C05BB55BA7BC4F31F"  
01977 "337D624A94C9629526ED51A9B93B1B3C5551F9B92A30759D7A40025E98E50128232A706CF6"  
01978 "C3F6734D00571F91307EFD6C9718D78B3B792909C1C56BB8526CF5B229D4DB47E61D962538"  
01979 "BBB17E6CE8CFB7854EFEB5591CDFD5DD8BB22FCA50E4D97BFC4C36E2573C0B495FEC9830FA"  
01980 "C229976FB651B82D8A34173F5635875B1460A023CDA65333138C89DC81275F7F34464725B"  
01981 "8C5F93AB130009F5B797D9ED536ADF5A42ABF6345B1457D5C96CFCBFFBA56CF124E78FEBF8"  
01982 "A8BCDDFFA97831A589D177AD260499130EECF8247FA5C44B3C0C19AFE6BBB0A7B87080E14F3"  
01983 "C3457C8424557BD9078DEC09A3A6704E1D77049A1A4431E423B28B9476EAB6292CA1B3B46"  
01984 "3E18EF929EFFCD574417B15A4DC1269E593F884DF5D492A464AC3F2436B8C691B88658180"  
01985 "E53228AD83C4FB843F8571D42A9BDEF37EC4926DCEE5002A12144600C9CE7AFE483F3A64E4"  
01986 "80CD09F8CAFEABC9710E01A8120E84BE5861D75705A8D18F02756D8FA78392F1C89D05F37C"  
01987 "B9F768289B8CA99E22E8F68200770D6DD2D76951A45F73C0730162C363886282524436D029"  
01988 "786E7CFA2EFBCD3D05169D362DC8BD472AB376C12097443080F3C83AF1EF7339D8FCB85FA0"  
01989 "98BBCB5EC3645A03902B758DB263A21C8CC4874F96D672EB192BC1A183A1B271BC1D6429CE"  
01990 "AA8208F5CB240F451A71B6F884537ACD49471010ACDF28887F6A0A2A04E738CF9FB5B00E0C4"  
01991 "74C460EED1F791F4EAB3F9AEC225B40CCE223D23D07CF0B60846C11A07A86448B9CF11975"  
01992 "4935E80350725CC60AACB55C71F4D5F5F2BDC3E412039D336ABC6F3D46EB0D1C2080CF260E"  
01993 "E77F5C73A35C23396E2324998B7DD375537DD35B5F27C3B545492F682C4B89F13F65FC4F72"  
01994 "539E4A7763BF4AE2443FE0A2F683A86274672FB9581668CD2B075C274B242536087C4DA1E6"  
01995 "9D0712161B868ABE86C98468FE43A42DD2AE30089ED92B7A750C73B19A6D70FBC0AC08299E"  
01996 "E2A9BC94652B35D5B2D679220D34D3372AD756F842861470CC3A3F2BEE75112B138613A40F"  
01997 "A585E449FADD6353A09363EB6025D4B81F9F224817FC2DEAE01D797AA8A0F8C945FC69D58"  
01998 "91355C28ACAB8997A0518C8F4465953819113914A7CB472EABDD1A8D943A82FA41315D006E"  
01999 "54AEA4CAB21601A2ECF6CA0521579A6A92DFB6B77BEC6D5A05B0A72F16B0328C860FAB66A1"  
02000 "D6113B7098FE50F3B09C2306D0B9D00987B351CB2BE099A7AA7EB7AA691A4E2D243DD7D0D8"  
02001 "640CF80A4D2FACBEC3C1CB222883CED667477D189100A5677CC3FF0DF31D1119D5F31ED2120"  
02002 "E5DAAFB9B9E8F36E77B3B078C47B3D527DFD6521D7C", // PRN 05  
02003 "593CF84751C21D591BB62FA0B0AB65E993408A016415D6296E0F080FA9149A4C31D8B2EA68"  
02004 "DC72092F6B61B82D8A34173F5635875B1460A023CDA65333138C89DC81275F7F34464725B"  
02005 "94570C1B8E9B0A7705002ED56D335632ABE3ED8BC6CAF109F0588040DFD167DC364EA7F692"  
02006 "D5F59C5616A67DA0C5EAA4D9FF017820225D5B164B6106CE9307B56EB08C563F123A05A4C"  
02007 "93488DF63A6E4274BF5188475E8F7ABB8C3E7A8323F689DD93B043BD98BC948A567E8C6C95"  
02008 "FD3788ABE728F7B7E299C460F9B35A59BA4429BF417B9F8C54F2DC475A7109B9C25C0843AA"  
02009 "86ADA9F7388A9915E65EA08C96C4C84022368E79A3A466B247EE6A37918CA0A8DC6AFE291CB"  
02010 "9D360D727B6E7415D360AC414BD34DA0A1D995EE1303896465D52707A4F31A30C7B7DF936"  
02011 "391435A5FD3F0E6336CDA36E473E3D990A6F04FB5E6340581AE86D8EF81B9BA6897905B8CF"  
02012 "A1363D3F711D33EE9E3A8402A14AFEBF339BA7C34FA4EA2004FFD09129667646A79B6322CF"  
02013 "00E1B46C1418557E0E62E106901176CA771979F4B6B299BEEFEF7B847F4E3ED99F56D44F5C"  
02014 "73F00488E1DD862BFFE5290DBA737FAFA0C9D95C8307A10DD309C177BFC46A9F6D3BC86F59"  
02015 "8DC1AC69B070B9AF1CA2824F456E73747A0CED38631F9AC222914825CC3898F04F6AC09B01"  
02016 "EF4CB02F63C9436625E2D0AC3C433A31D86948CE34663E5462652980B3B57C0920EB63879E"  
02017 "28CA9565BEADC47F29C43F1718CE34CA5717D7AC2CEB6758A16F6D769CC277BDB483808CCA"  
02018 "3A6DA99D1018888BB6A49D325891B5D6B4B77B18A5A4133AFD5E8AFA542710483B52E3D51"  
02019 "8B84B77D2458C4FBE2DEC1F6DD8C0FDD8FABE2130297D060F07B305B2CEDB39204E2B8F8"  
02020 "507ABD49C0FEFAD0332F8ED98736FEF5AE4FAF8515CD638E28F1555F54D7A64FB38D0B4773"  
02021 "4D0AEB8B55A72259427613733F237FC57DFC1AD49930D9844C9F44C3D6265BE3102D93E2B"  
02022 "62D3B0D7760F613BF5F5176E0C5EAE47970D30D211403477894B8D66D03CB3AE8992E16D5"  
02023 "2F6B2073DED09D504B36289644241EFD21018C570A2CEE6D59667F274CBA5733E41BE3370D"  
02024 "FA47464AF850B14DCC32BCF1A5A8385314E8B38B2A642AA90B8A5647DF9B32B83D3B4B2AAA"  
02025 "54472BFE84EF7DF68ABD0236DE171E5AEDE1770F63823EC143B30FFC69ECC464CD755F5AFD"  
02026 "2B51AB5F05B74E96F6108810CDA94F99DF828EDE147BE061FEA46CF6A84CAC720717733A17"  
02027 "677BF620C4963DD1A08512CDB6A96DD3ADA995D84A934C9825FA3B588D916CE1898F6C2F75"  
02028 "B75481383B27204C343CCCE41E10DFAFB42BC6BCCFBF7BA4C460147814014F9470ECC8B99"  
02029 "D677EEC872172B7E5839D263F55B9FC91AE6143FCD1BC3AEDB12312ED10647E011A020B27"  
02030 "C69E35CBA94623960543DB08D67B8CFD06760450A067E3F868386975793AF43AC4D0F3773A"  
02031 "FF6C2CEFC8310C0701CB934098D842604E38370182BF05A401C38B6CFBD19CD7D4BA186A59"  
02032 "5ABAA48B8740DC321FC031D76F82B4F5FD6949AA101D70A702E18D8C2BC1283C2E2CB13869"  
02033 "9D507FAEC5B6A092E5AECB0AD9BF2E59F175DACE05DC18E485B04DDB963426A117CA0B761E"  
02034 "F485B234971B6F681CE3CC5CE58A9BDAD9267E5C075D1BCEE2C88637E2A10AD441E9B1CF4A"  
02035 "324B77C6623F4DF9FC6C4AAEC068F66017BC82D562722FC93B02489DFBB5560BF60F57B736"  
02036 "212DD8A0533071D10ABFCB4EEAD5AD9E36E856C8ADE0B9C4F6F5116EED43A39AF8FFCFA01B"  
02037 "D558D7575AF0E36F2AFF74E29435778F4BA57A5D90", // PRN 06  
02038 "214AD702A657F5A17A601D77E0F4C6A67B7B8C074514548FF89A781CF2E0B5225D95A944D1"  
02039 "048338FF3AF206D331DBC3F5E9E89C7E592A2295CB783F416C4160FCFD26983ADDDAC52BB"  
02040 "FDD6A7125BA4426715F82AD9815F30B2DDA972A2E814D6BD231D453858749D65EE1B4C2025"

02041 "F61EFD8E1E250A3719899C611682E7B2AE0CC85294542DDC04140ADDCCD5DCA315C22E5AC"  
02042 "B196531A3B0071DC787C2753C3069F5D302A9FCF99B71317FF2749749C6E00B73F382983E1"  
02043 "C91BC8115FAEAF083434DEC7E8A22ECF2D1CDFC7B7ACB80FE0DD1EAF7510F694961E09882"  
02044 "E50994169BF60C47B95BF37B5D6F13CFE48A9405637F87C157863C58E479488278B918DDF0"  
02045 "41214BCEE2ADEFF4242F55DE971E1A7F2EC566B362F89DABF163A690C8A8B808ABC1217C7D"  
02046 "49933C30272A08689F7519B35FF8C1CE0F338D0B75332D163A5DCDE714B821AD3B8393A174"  
02047 "0B6BE4AAC6008EBBBA30B8E5FA2253DDCF16B3F87B4FA47C1D36AC86E2FC39DE8BAFA79E7D"  
02048 "58A3A6CD83FE09448EAB6020BF121234B23A72BB7DC02A2E235BB5DE3B2AA97D3931CB8001"  
02049 "65B93851EC81A1B4B7781F23E5C946F035026888C3D8C31170787F78333FFF8ECD8835AB5E"  
02050 "9D165A8B1D97B19936940EC474D115A34120EAC800E8D680B36D9E29180BEAF5B3CFCC391"  
02051 "3EE7FFDC37AC881C441F79A39F4A4161F3B403564F654B0E7D8D5415DEB9A6E11F84531442"  
02052 "1CC59C6A90826974B4CC477C280ADF4FBE8368BD848AF7F91D12E9BF4AD3E9587DCD7A280D"  
02053 "EE9A7F2B6F2D2EBFF7EC9454173B748D2D90A6AA650D3BB97701EAD27D6AA1D05B817603EF"  
02054 "1026E643C8816F183ACE863968664E01D6AC58C2A10AFB51B1A19A5B7BDBAD77D651B78C1"  
02055 "41AE70B074B9EF74BED349E58CF8A80A910F6DF8887202317BA36950F3B7EBAD45E6E94092"  
02056 "1043476B0FBC136EB397DB4445706F032FC47C454BA08DDBB8E71D1E7D9045655E1039C3C"  
02057 "3D199EBA21596335BB04078DB6B39A4452662D282C7AC9B76D92E112B0027B711151C56A64"  
02058 "59FFC70BFFC86941736EEF705E8DF5197B4D3AADCB31E421A90FD5179D9C3EA3EBC8F7D9C"  
02059 "4C29D736076723E2D4146EC325CFD61F6B0B70AE0DE179FA8F33A2536DBCAFAA2BC0A9740"  
02060 "BBF41134655EA16438BAA5F8FC96256A6F9999CB1C57B69087C0612FEF4E6CF0BE5D1FB772"  
02061 "58AF418EE089088AF3DC38B7A15FF6B8D2BB739FE7D0099AA4694781E581A8F7A49BFCD102"  
02062 "B7880E97B89880FC41D0890C33481B01D9BA991BD41B8D6501D34E0A4014747CF89D089F3F6"  
02063 "A189ED80C3006515CEB7F9DEA85D0766D2989B88280FD46F7BDC7C32C2294B2A1FD435102C"  
02064 "2D92E85A03775CAF11D05F2367B57F651C7586FCD32FE920B1B94EB431AD2497E75F4E3C40"  
02065 "8A0FEF89907C6C830FA14402B1103DA4607F3DFAD625AEB3235D2392954B8D3BAE8A6D02CA"  
02066 "E6D12EFC73BE46EC5E13D2729787CBB5C64910347AEE384972A5716B5DEDB99FCDD5660E"  
02067 "9453900DC1AF1876D151C00E8E4BAD38E84830EDE92CC3099861C92BDBB59D8DE31AD9F4B8"  
02068 "89D480FC8DC70382E8494E1A25D2140740A5F327D51D0D3CA4259F1CF5DA3474FE28270C9B"  
02069 "4E7F8A714F7C69D1FC65B48D7C9C69A39CC58E2C77D7A750CC0B0E5D54B2F3450B66994DB52"  
02070 "4F15AF77CEBAC3CC611B2C95F6C93EDFEB652A70748DF1788E6FA2D45AF171C1AB59B0B2A"  
02071 "160C075AA4A92272BC0F22631A24EC94C5D38664B08B46DC96B448C351C77F81051881055A"  
02072 "CE0F364C80372EEE5BE7C660C7406B03C8F5345A04", // PRN 07  
02073 "435EA61E7DE31409D04501BDF0CFA57A84C1387E0684538E8BB651C7CF4EFD6113EAF9383B"  
02074 "FEFDEE4C702FD5A070FAA3DC7D644CB3D96C8FF65F18C88C43A2E66EB78FDD1EA91E9CB519"  
02075 "CDA9EE1806F0F10653F3AAACE782810776AF25C9CE1C580C381D00B95CB02AF3F9892B4D918"  
02076 "FEEBB01E7E730079266C4FF0AFOEC305D3A168197158B5A0D4B1C1FBE4FCFBA947A5CDCE53"  
02077 "24DOC4182069CF328465EC2900B468E145C0F41D724D65547808BED468256EEBD76431E71"  
02078 "5A4ADAC80D5314ECD0D3177AD9175EEA9D9149128DE8715534A648069F5EB9ADDE486029C6"  
02079 "9C81B1ACE8C89FC68887962AE2C063C5EC7B594D88E56F2F3FA4A20963F3BC730E74923BC"  
02080 "2DCAED6EF38412E9094326930D2D2E3A0FFAB2217E9059A88882E2AFD3E62FA4DFC5B084D"  
02081 "0CE53655539BD0931BC063CBD4E660A0B1BD9C7D6CF2836C947CEF2C07B859686424B57FFD"  
02082 "D0DB6C34DEF824D835918505F36755C298A09F935A2F95328248B5E923E55AA4C91BBB0733"  
02083 "E4940A34AF5226C21EEE9B0E0C3D15397A4D8C069B94C2C38F61FEA2077AEB1F7C5E0875B"  
02084 "9A7F2041BAEACE95807EE05A366016B6E0FA2E7E722BA34D3654DFECEE78DCAE199BB44E16"  
02085 "AE61729CC7A924CA6C7A6C061C6DA0936B4C70A02223EDA01D7B8C0D2E012F998C3B94D138"  
02086 "7B6CC65DDA36360ED765F0A23EA562626E12145EB2FAC0F49BB0DD7D1ED3E9B2860DE1C54F"  
02087 "EE8D60EF56D4382497C36181E864F92A95E76CC516E2E1396E2B0D880ED6CD14D2410C9BCA"  
02088 "159F57DC9B763ED7CAB73F9EA5D42F4BEF91977C114900CB0F249B10DE8F4FBE220181D4FD"  
02089 "B5D836AD2FB70F06869DD7F197F121FF748ADD6DF6C48ED9910F51BD7528DB60FC5A42CC0B"  
02090 "72DCA6FA66B617F12FB2A9530247B1709DFA7729A6C9FF5B13D073FD7165033C45B5B7C7C6"  
02091 "6045D173597132C49CA4442986760CF8F0FDEC0C0BD5E992641075C76168B8316057A993F"  
02092 "513536FC917BFB783C28B2D4878D120889945FEBAB84BFE4840C61012043E96910A374CE33"  
02093 "B45F461B5492E2F96F0EF91E42D7DC755F37117852C4FA99B106F5CB5C70BE5055ECEA5DF9"  
02094 "0B517B08252AD6F20013FAD50F47A1682CEB37AA0DB64E4B699E126CEA7034969FC5468D1"  
02095 "9455A1F2BA4011AE5CFEC2544329E268326345C9970756F8712E7793301E9D3C92787FD10B"  
02096 "7637F5511E27A16B8DDC21C116F13B0F27FA2E5B6DE60837F011937A986C00A4D9551D37E9"  
02097 "94DCA84093DCD5C4DE6F40D0E25349284131B9E0B3B6B9BEF6B9DB6DDBED89CA329DF3B0"  
02098 "32E2C05BA9999CB02D7896258E3C4504643DBF49A155F96A69148A51F243CFCC3B2BEA7E8B"  
02099 "D534308E752C675CF478A231CA99C2EE402EB5DD5BE2AF5F4FAAC783141BE8D2A3BA44156A"  
02100 "3D1E8B9EF763108B9EF4670B25569DF3568FE0E183938FB150FB79F7F28D51C8CE40089C06D"  
02101 "6A09F25845B50E64AD231B367C280883CB7FFCDD5E4C044BAFE6C0915A0C52BB1487608826"  
02102 "7EB84A01B9931C03DC799995B6969E65CCEC63697F1F0E146B75EB6987550C23DF2793FBA8"  
02103 "F04D019C31AF4AA747F3BCF00D6FC86CAA707F1F27DCBE598B18ADD6BA048B2B174FFCFD20"  
02104 "F93DD2F8E60885A15CBC2F06CB10537C205E143ADDAD33CE8CE5E6EB22C6A917172BF9620"  
02105 "80290AD26B88A91C7B63B64047BBCB60309C01F15271CAA73AD93DCF99400BEE17F6F771A"  
02106 "FD156CF28B788586A18C21739AE5BFD8F8F6B8FC94A9CF1EAB0399F9FB53FE70FE5E4278AB9"  
02107 "09B2A00F5A181F67CA7F7E04620A2394C5583CEF4", // PRN 08  
02108 "C1A7D5578E6929147D34AB5B29B97864F46686736A0966EE7298963AE27040F4598E0AEA2"  
02109 "35B562DE84BB1943891101155AFF6D185D9F66E5F9DD037B048914D7C1B55505455D08DB2E"  
02110 "0C580716AA4C4E779EB7348D486F445CB8BF499DA3054F087ED95C2561770335DBA67F1EDEA"  
02111 "C600071AB5918ABEAD7E55C5CE609896ACE2426C47941C1FBD8C4804D8FF4C9290E8ED35C"  
02112 "967D65837D6E2B153A3DD843712AA34BE1FD7CE0D829B04DDAF036D9D16063E28F627491C1"  
02113 "C37018FBD1EA3D33899BEDE421C1910C2957C13B8CC4B8A06C8D3B2264E8E1B8FB581622A"  
02114 "9986B4B6F673994F6F5FA072ED8A8564C2AD450CF801C9992779F292994A98800AB12A9EF0"  
02115 "7F777F441F19FED6314404A9949218B2ED4E239EE677D8D1425D0547CBA99C1333FCB1620A"  
02116 "8D11DA5CEFC51DEF5B3C4862E17A6BE3C194F88B7BBCEDE98325D63887775ACD448C0A970D"  
02117 "63603B58E3F4B8A361D36EFCB6AAFF2858E14B1830FOFFFC4C5CB08BFC21FB091045B9A7F8"  
02118 "DC0FAA02A62D1FDF7178B5DC90D2111A64CFAB0D83680196B21D75EB91A5B3E2568C7E8A6E"  
02119 "DC65048C88EB34F6494CB697E44F895012AFCD7AA86DF341CB9C63A7EEE8E5F95E3D60213B"  
02120 "C23C5A83CFAD4225BC731F07B34FCA550DE26E0D6DA29F03A1D9C4233D9A0633D31AC5F55"  
02121 "A11C1208C8B85306CAD41EDDE3FF5EC3A61194E724EE2D6A4C94FCDA217BD7C7FB328A9CEA"  
02122 "8064D94FC66049B56AA1DB9F977216AB4242D0EFA59AD18540F1430A6F4EBEC5CA7EA81EE4"  
02123 "BFC6E77F9EB6352B71275D450843304F073AA82D77EE5602B43DB62B9A733320A67B2727B2"  
02124 "1B705B86D5DC1459B5EB688D37236FD37A7163E78E8425C589DC600609DF3E9F36F9094D06"  
02125 "C60731527B8ADF58A95E40C1390FF657D418D675F528B3B665FB5F3C56504DF40C1D3EAB6F"  
02126 "3CB85F5C7954587F87744775382517B4C652B4259571A8AD219A6517E2D6D3212AC1684404"  
02127 "DE7CF71478BEF7819DA24B09A11DDAD5717CC63995BF83DDA735C3A50F6B164F26D6A79E3B"

```
02128 "20A1488DE2AB44347694DB8EDD3D957B6B5510C4A266E514164D23F1594D84487757EE6F43"
02129 "53BD06CA7F90912AAF5188F217FCC11A62198E2DB04A016D36A583891D346C0D84241027DD"
02130 "E80090356ADEE1AAFE7CCDF983FD7F8827BE286280BE118FD9FCAB26A10D19BCC63A9AD9D"
02131 "202036193CDA36321A5FA7252D0E65FC0D8E588D70C3B629D830CD1DE2798309A6B6113FC2"
02132 "A16B341BCA4271B5994ED767F2064FCA7DA335BD1D6B21F54CFA0DDCD47E2A38CA924DF85D"
02133 "24B869F6DEA8FDDDBD20728A21034355728DD84287FF0EB5A9DC113A71F7C69899EE76BE7D"
02134 "186B39C7C868E78C03E923FF199246908FEB48AE45261B3F7A8D6E4FDOC280545A9E3A5789"
02135 "16364196E483F57EC37439B0FC9FC37C5CF56E21A5CAE0B70D25B928EEEE456F19821DBB97"
02136 "4781C646E860EB3DDA8EF8577CB0E3B9E96245E14CF4E51A5D11F7AD9A77B932467808FF55"
02137 "E588455F810CD50367DB1621469AEE8BA6FCD52C75BB027521C530A918F9D646C69EF8DEA1"
02138 "F7B14354815F11E0187F01F677E2A0FBF78ECA2B1E51E898E26CE55693ABC8812F4CB4C10B"
02139 "9BBDFF385C6AF0B340DF8DD2E957D877DC1AEC30D4A6195F40F09B4CF4010E810C764EBB578"
02140 "91652C156DD20584E427902DBE82AAA3CFA728059D31C36253649FDC00CCD29517AF2FFF4F"
02141 "5050B0FCBB73B2888C989761A7F0551375E4D3C54E0C9F8A4679E640A73476BCC6587195A8"
02142 "2CF56E55080A85D57A97E3E5D0B9A194EEC3E41C0", // PRN 09
02143 "F0E94A299789538824A90801AE02D93A78EA914B9A8F10FEA2D2A7D0812009C732665D48F6"
02144 "DD58CEE4CF72513CBADCFE171F837E79B8776B392B2224C02766D0468E513D0BD1BF6EADD"
02145 "6C0437464503A3EC8539CC707110D18FCD8E000EC1A3DD03F7008D37C1FA02E419FD4CF7"
02146 "A6C81157E07E353F3C4882C2AEF4A0CCE1F7A5CB5EC753528152C521A2ACF7548F19862EC3"
02147 "1D8783BEF406EB7F756E41CAC3FBAEB9A901F9268E78E06ED5D6CB19BF4606C9DB660C03A9"
02148 "D5B06BC37129B3458C2F2A6517694FD8816E8C71E00D77189768C423702612D9863FC2DC1C4"
02149 "A2F4CC9387B13AAA8131D9F21A5D97C13C7DABEBCE61D72556466093EE0F87E9AA9CB4224"
02150 "5E1BF55157013C4350F38596CD26D2819F98698CE98F5D4023BDB3C974B0E61FC6E6604CB1"
02151 "707BB22FF5689F3DB3AD4831BFFD3F93C702E0DDBEBCAF626FC2655447134570B7150E030A"
02152 "3524A6979579296CEE062D18CFAFFC4474490DA81268B27B60E9E9BAC2A9B7D2E2377A2006"
02153 "AA5C439C263E8F9EE360B42E217646F7C44EBA6D9E3B98AD20B1BF7CB8CDFC62CA7AB139B"
02154 "C2DD1C85A837F51AC9BF08B3F36CE176A03F358E2E37EB5DECD0DD2D544E58D203910613E"
02155 "B14744E914F92C5E9274FAB025CB6D8AF16EE03FC86AE3947F7A30FD8C1EA23EC7EDAED405"
02156 "4174E4E6DE048C563B1832F8839927ED3EA54B19BA41CC144C8D8A6D8FB66032BA6F4E16DC"
02157 "81C98D37129B3458C2F2A6517694FD8816E8C71E00D77189768C423702612D9863FC2DC1C4"
02158 "593809C74D0AA881DD7477BC584376DC6C177347EE17F3E17B524D0DFB315196035FE6AEE8"
02159 "3675494E23BF2433310D4C72E84DC82BDCA9B791D5BFD620212172AB2562B5C36DCA28C5CB"
02160 "8078CF41F2BDFAE651B04243326DCE0C51ED88699CFAE4DA3B478DBDB92DCF8D2EDC4472"
02161 "691173B6C5B9E5E116C30DFFDCE7B0381E8E1013CB392056E3BC7442B2A2EBB631ACB65DE6"
02162 "39B3CFD8EF3918531480A35314F3F23F9AA6CA41A9B8F6D08F6C589DD2584B6B8285230808"
02163 "EDBCEB99C0E918D4E73AE7B397328502F2D47078CA2FC0BBACBFC9D9E491EFA2A2DDBD682"
02164 "E0D829F74D5ADEFA100E6195CF3D7AAD2041078AACAB8A8EA62FA42C6AAB6203D92794F77D"
02165 "4E30B84BD16518A9A0BD0AE4B372A537BEDE43B31A6FC1A58F9B43AABD485C55E630FF8FA4"
02166 "359DCD52552B8A09DDF5CEFD972551714E5F0A2A16442345B33DD9D232AC8431EBFBCC5D5"
02167 "165D12961D1E954D01C0379B5489AC0DC0357667325AA864D44269799DB710BC6A25D90AE8"
02168 "E0C9368BE911C91D68A49815850B4B29708BAD74B4BF3590E6488655622DAF3A7B52356C80"
02169 "4487133223337679B33E583B622EF873962E8FE259DA8F5DF61748B6799A14ECC763A1B545"
02170 "42D67E57BB73E411AC023907E54BA0EA8B7F20E55739EBED7F83E408238B73249CE69FD2DD"
02171 "593565141FDAF12BFAD3A43670935354ADA4067DF0AE4150AF306AAD3071EC4DC0AEDE6F56"
02172 "8DCDD8B8EC98E4D17AA99F8E881600D35B25CDBDBE5DE184180A9C35EE8B42F6890E068700"
02173 "F06E4829A742FFEBBB214D17BA925B5C0B41244DB5BC7AF49D9A24B2C135131E553CEB2C2F"
02174 "4E6E2EAB840AB37788323024123853A19CC04F6E344A175A4BF1A16776B7F84A17501DE4A5"
02175 "94BC15E788FD0AA0ED4BA3602838A6D1D485B1CB80BD24D97905A9A7E2DE71A9A4CA3268"
02176 "9C46AA9EE52A9EA985C85C9289ED1577B577C444FC73FD1BC3DE03F9FEFC10E6A17553A7E3"
02177 "FEA6D57AB71E2379C9C5F8904EC43E57D5FB119CDC", // PRN 10
02178 "A8C2390DC3F0B9D01B9794A4207DE3F62F00A4D1188C727B2910153D9BC6337AA46D2D79D"
02179 "09DB7656C0B28654F1FBAC5C3DAA8A75B16BEB5FAD8A256E38EFD232E007A741028319EBA4"
02180 "A6DBD0AE1C1F8989246476AD4090F9F4F560D0CD72815FF6AAC29C8674B1D366120EB32133"
02181 "09DD9F3901B169A1CE03D2A5E471B9B8DCAA957259311B97817088B486CB680047359440D"
02182 "D39386DC2BC24AD2AC30436A08C335199F08861F93177640074A241D9C5884EED07EF0C0FC"
02183 "E8CCE08BA30221150C9EDC6562344CD3E14D3735B41DE2CBFAF4DE0F6D417299FEB3814662"
02184 "4487133223337679B33E583B622EF873962E8FE259DA8F5DF61748B6799A14ECC763A1B545"
02185 "797DE51E4AD9B3B9E1F9BD18A0FCA255CA8AE7E9E192ECAE5C60E9F5D578E59F7F7074A12F"
02186 "A75F9388CAC0A148CE9223681A93D62A679254AC2F41CDD871BD73BBC6C4E0CC9DA67633B6"
02187 "1BFEE815F4E15ABDF639899DE105A5F4D74CCDA3EED3219BB447EA5C6411CEDB351DD76CF"
02188 "11456EA762C2F300671A12DD6ED8EE9AEFB4D0BD86D0CF35AEABC9B9E7822478A6DFEBB887"
02189 "DA0E2D3ED4C010E5031C4332FED83480C21ECF904B65925C4CF64595F20D072C7C6AF12DD3"
02190 "C76C682833CC6296BDC73DC8E979004DE4ECD749BA45D9748DA728AF757FF5422B862231"
02191 "46AF86AD4AC4D4F407DC2025B8166E2CD64C8E1543E046ED91AB1AA5524E88AB6851752AE5"
02192 "A2F0FD933DB37EF48775017E000386805EFEB8CDCEBABA613A87E0C554F99DF0C8FDE49"
02193 "FE73DFFD48379299FFCF738EE21B9AA1AE4DCE7BAB98A356241545B6F7D186AB544F77CB61"
02194 "B6492BE1F02C4B77DCD5083DE4319D1A1B792961D1D1552F8AFA649EB5D2AD64227524722B"
02195 "0CC97260A8B36493B00D258A570CB238D3C2511C0582DCB478F878528B8A8A5B33F9FA6C"
02196 "41302DD1BFAC664C692001F7F7D207A8850D6FAF22ED887589405739EA5B5B9812BAE61788"
02197 "A225FB2E6F9EAF77D3B3758F9ECB89637D1FB20C428AE3420C3679B834EFB0F89A389FFD7"
02198 "2D7420F2435D118045FF7B7361FFDA0BBB6934F81AD8513A626CC25D9A5E2A5FD1C51229E2"
02199 "D5FF803B8983B672E92A1FD72C16520531C694F44C9C99BC3385A610239F6963E0BB66ECDF"
02200 "4CA29E1C175E414A5D3F9E9F495F2D03A773965643186C46EE9BDB41876A41ECDA224719F5"
02201 "C76AFA6E955061805F023498AAE965184ADD6F6BA1DFE0BBEFC3D1EDCD40D8F55BB518C9B4"
02202 "5CEDC95682B24F2818795CBE5D8E504F0885863C750524D9D238E4110AEAAE095E1C545C29"
02203 "89BD7E0BC08E8AB750A7D319AE69CB080DA562F7DAA7068F38B14AD7282E041835B58E2A48"
02204 "A340F993D10FF4F805280C183A3FB4555B8D54EC145CB7B9BEE71D7BA652FC533ACE1BC8D"
02205 "621F7257422B363A5AD2798FB0E6E536854F2BD25914AA0B6DBEF1E2EF71FC7890E20FBCBD"
02206 "AF751CCB0DD16098D96FF22961B6141257AEAEACACA339750950F6895107355F04F9BB3EEAA"
02207 "B3A9024E4C3B79B19ED7947EDE79958B669077E6BE182020536F7F234AB2332C2F2790DD96F"
02208 "5816AE3FB5F2CCD72C6D177845DD0CACB44128F6EE6B1C2F673A8A50842E13C8FF0087CF71"
02209 "8BB8A0715637A40021998652FEAAEF5719ADB71EEAAFC8093CE525602599278B35600C743AEB"
02210 "41EA780427D0894158DD4ED564C4742033324C58B01C15822B97C9634FC8F320C0B103F0"
02211 "A904D98C9AC529FA799B30FB82C0888E47A3BACDAAE5A3DE6DF39A693A9AB3D95E7C973C71"
02212 "25C0F7D8C59B37EC924D7B9737724DD92818360F2C", // PRN 11
02213 "2EB63BA5756F5865AE33947E311A5EED4A0CD056442E19888A2B8302ED8017085DD5D137A"
02214 "3769058F0AE0476AB202CAD822D6A781291C537120F316E90D2C3CAD13381EDBAD35505A5B"
```



02215 "7DDC41E46771F460E501B9BE17DF8D4DFFF631C365AAEBDCB7D5CFF7010BE2EBEAA0B99820"  
02216 "E0BF3D290CF27DDACF397A5E4922EEB7D8BDFDEC96B87F8202BD409F31A200515EB6A05E25"  
02217 "A5FA504093F1D82A83A777F5389B5C90E66C369BB654598CBF3F258E914DAE819837A86EA"  
02218 "775A05EC829B4889B816154A2B7882EC7FF71BA7CDAFF873B5CA4F35011DCDED8C0CAAD101"  
02219 "F7BDCE1CBE67E081D98EFAB71490AD12AB7BD9B4CB9331B6B04D6DA2C12311F96E01AEBBB3"  
02220 "50BBC7311E8AFC9A6F42CD541ED40E2D7BAB44BF7F1E4465CA942A0E8767A38D89D0231BB8"  
02221 "97A995627EFF71B3E4483ADDCAED084B402A3CB23D5F91872BAA49AB361281454ADA57AC8"  
02222 "38AE2B34B799A1B46595AAA02A9738019EC3644C63BFDC72F72CFC35D48D83B831E6EF3974"  
02223 "74C43CF665D64DC0376A437CC768862A92C941B9BCB8FABC6804C03BEF7BD2C5EEE4609870"  
02224 "803A5E9BB2FBC42028C5619C3706F773426CEC39D096FAD267BF5E6C7C7510B0D1948FAC57"  
02225 "3C5716BB4D3AC6EEC492B347497B47C0D18C77E7C264E9A0A44C3DEB83B1101E40A0A8653D"  
02226 "9495D47C13A5BFF3C6956E668DFA999926231811F3AE84195B65213FFB824478911230719"  
02227 "E8EAF206DD99C7CD7D3D7661E9F39250793C1391033EA99A0E31D8CF83997ACA08C43B4724"  
02228 "0BF1E5E9D1D741CD9E6D3F5DC2043C9D3D68A2A689615D39014B75F5B3B4866488C4B37B68"  
02229 "B5DC5F9B9D6AB9AF190789CD610E939960ACF607A488964C33DF0091FA4EE86654142083F6"  
02230 "8436D9700B3B50949C0D97098D969BA25622CEC15FBE7D7C7B78C1C49F671672971E38996"  
02231 "C2D77A024919FC1A87BB9BFECF951EB75326CE48D3837068F31B9592DD8505AFF1E0E3191"  
02232 "1D1F4E311C81B81D8A03D68A5EC075D8ED7D69A1D595E7AAA70928DE93B979438315A4B98"  
02233 "EABDC8B5E932257AAF79A9CCC74E6FF88C02C33217FCB3FA8DD72AFA1E549B9E5CF6099E2D"  
02234 "ABA07C91BD3BA6A9A4936556FFDAD89D5F080DD7F818EBC8ADAFB3D51D8BA7B48209AAFEAC"  
02235 "184E1A86749E6B8BDF92E77C6D4B8B983F4B93A52C473F1BF27B4D7514DBA522ACA79463A0B9"  
02236 "C2E0D9A2D47DB238AB9E2A1636B9AA825272F2B9C8D36D64614A4CCE2EF31B67682F328B46"  
02237 "D31FAA31DF874F157799B5D53E143C9E9127AF358BA1B43A9418FFDE00CB16D32AA137FC7A"  
02238 "5A1CFC2070FDA112DE600F3C05BA0F50F8AE1703AA2CB64135C29FA621385C803673A4D771"  
02239 "95B4B74E554B043FC07B106533D0B41F7D882FF3702692709A64052C4AAE51CDF8B733A92D"  
02240 "86D507B6FF4CFF5F3BAE4D0D9F3F90AFFD774591E6E14539CAF99C833D2A1D9A12F7E9CB1C"  
02241 "F74FCB353D1E22B8216FB91564DDA8F1D2C9865080422DF0E0D8AA47577F31A7682B5E36"  
02242 "BC8344D5F5725C819D3DD3A948F4E6F6C893491127267145DF1B286C18C4F3EFEF0BB6154"  
02243 "1883E413EC3EC3BBF99D83E3919501A3CD59341BDED35BC39B655D61686555CBAD7A1B033C"  
02244 "2E3E1A86749E6B8A12E43793DD1B72EBE858B40ADF4400A59363BC56063115492E114688"  
02245 "F47447132D606F2B99DB9CA4083A91E35AED045C513EAFB7A3F0040848C1DDECOFB3C2CFF7"  
02246 "17BC4CA36C425EC3CA9BC51B3B3C1885FD8B4182AA22AD38CCF8FE3990DA75CDF5712B358B"  
02247 "E866A6435329677CED89571C3877F9ACDD0D3E66884", // PRN 12  
02248 "F0F04621B3CCE418BF2A77BD47D671191216B230C276A0D081C4F79AEF839857586F9DBA52"  
02249 "B4E35AEF22D86FBA66DAFE801C686F03342849B5C3A492280950C7F3B725DB4BD6984A6389"  
02250 "CEC1AB86F29D653781EED847D7E69C99320F8584700E662070E9C43957F4DDAB28DD242741"  
02251 "5FEC6F1503551D972DCF11BC4EC3B3FE745DA7F717F35832B8CE373A00AE6572D8DFBD7720"  
02252 "E00517C50E2D6B34B6BC3942F9A8C035AC80161CDB45AFDD7E9C473F8EEB4F1FEE7A677E820"  
02253 "874F03B546464939F3EFDAA157C9F478988F3ABF39BCE02B722C559CEAF997A13AF66BC0B"  
02254 "BB4DA7B68EB6CE8498129A95D021B7C3A072683593C2990627FEDC52F66F3A16A8064018E"  
02255 "6618BE5D7C63BC622DDD4A15A33EE19E4161A50EA38CF6626D5A088FBC5721F567145CCE30"  
02256 "C66DE0C6AD79D31B703B129D27627BB5AAFBD9BC7D43944B9B0D255D8F338C7CB7EC56D121"  
02257 "C3F4E61FC0D46CB0E036262FC182E8F266741842DE923D276CED79D4A25FF7AB3D2571B1B5"  
02258 "89E01583371B8C8D8196A1C991BEB90F7FE71B99EF9E100375D3E767637B2BB1FC907846C53"  
02259 "550DC46C778AD7FFB9DBB65AC30C9644FE9D59A74B0E81A5BA068CBAEC9D6B362BA05202"  
02260 "260B1E86EBB67BAE2D74F60B7E1C0B510BCE4EDD8D71E342D5465A8032E87617790ADE2770"  
02261 "D0DEB1ED26D62A863D030C781D2FB02490718A5B55D1BF6B3BD1E3B3C163361B2F13E616FF"  
02262 "C87C1BDAA77D3C88A331511ACE5A45CB94DBA557EC47A670BFFE839069A546FEB7183C27697"  
02263 "A56B65AD76273578FA009350ED52819C23AEAD72A7150DE845B899669EA37B4A12D63BA7D"  
02264 "6DFDCA495A0957BDDCEE6BE11B8B296358E92E60A2317CF4B7542CDDFBADFD18C88F5AA498"  
02265 "4614A131E992DEAA18FA172C998EE6FC8E936E69953EDDAA3FD233182C7B986DF9038B63B"  
02266 "1C649EB0F9A7E65B90572132408929968CDC39D18020C6D81A3665639CD2A3612FB2135A6E"  
02267 "057F0F68B10D3CC1C35DDA2600AD1C0A6FCC0C5DD0D82F6218F2C1F8F5B714C47636CF68D5"  
02268 "361D97DD1367681116F2A2A045894A85F5FAF1BAA27F93A3B8809404181040C24156433B3B"  
02269 "87467B27D7C3ECE44EB763D14356D6EBC7542E21863700505B8A10D4B4B6E9D2A4C2EEC965"  
02270 "3BB30D9FC68DCB53AE7C8B6CF3120C921B824765E7641A3E9CDDA804CF341012964EACBF55"  
02271 "D6430673B8C12CDC53100DD8BD618A56B5AB68D83EA484F48D4641B94AA2816BF06EC1D1"  
02272 "285C11AD060735509367FDD0E300413911442846DB240612C1389CD4554DC8D7D7C7E4D4B7"  
02273 "9145D4684F1FB75759E2D506F9DFE7632CC553B945D91FE0F9A829AD6A9A18817B09B986B1"  
02274 "8FECDD61541A0FD6B6070B15F2484AE8B3EC3DA36D7F800747BE336B749104C2695327A49D"  
02275 "34741D76752DC943897B29C9651EA3C034BC4B8E1F380A9FB0F9D2C23B627D9DD8317894AA"  
02276 "754B8998573F025A024F43E03EE86A1EF583CE02DC7EC2BF2A8EF9A3C7743DE2E6DE38D72B"  
02277 "5E33BC5CDFC12184C9664EE7A3F4FBE9498F59107BF8B6DF69DBD6732AA8013FF8B4525EA"  
02278 "9E40FF6796FDEA2AEC03311D6E8908D426293E369D52F818233FAE1C2BA2657B38FB82E786"  
02279 "4CE1177D5E21EBC8AB773EE928EBC4A7EE9F4A9CD2D9F15146C08E4BE0A77CE272A98AEA05"  
02280 "A6DC18D5B85679D1B402F9851BC84DC95FCE71320D2A5540D40BEF67EE2709147EAF3C3D5BA"  
02281 "BB0B21090BA4C94CF7242982BE35925F691847A7C296B973EF2172F83AC9ED5E0A0C91AA69"  
02282 "7305F79A3FC6A8EA4E7D1DA88A81D4DDBC1F604BB0", // PRN 13  
02283 "896DD4C837F1BF155D4605D46411A63FECDCD64D7AAE873E1DE22DD62CC2F8982EAAA66BC"  
02284 "AA8DE8DA687910A03200FE437BB87CD5E6D17C6F5B8CCFDA073F29183A1E12917C5D78CC1C"  
02285 "CEC5C2B04040D39E1E74867B21192EFF2629AEE22427EB5D2399515D892EEAD80934668BC"  
02286 "A906109F6ED85AF29E702FE6C112992CBF851232A4C140BE6CA9C8DB383D1729B41BE93970"  
02287 "4297BCB41E4B8C7DA5B24B93FE7E7D2214287145A764D3263C9EE136457A2A270EBD03408E"  
02288 "CB657B78CB8C0A5C1767521D75A8E12D9FCB3CE8F6321B55C50A6102381E95C12644549A0"  
02289 "E5AF29183316016B89F6797BD89BA8F1E932F4C91C0460F35EAE32F46D42D66F639A87C25A"  
02290 "C6E9B9C4E475B7A4A94F10EE0C1043159CA71BA3FB4F794E766C5A7C6F7B85088CE1A2584D"  
02291 "4A765D7888F1496ABE23ACA42FC001E2D30B3959AD7BD941916E27DA648833EC82004AA093"  
02292 "7723CAEC0A8E7FE3C6742DF8E1CA797F13FCC4A07BA75874FDB6A66014C0EC688C5F5E0D1A"  
02293 "CE231B32A1A14A2503155D412BDA98FBEB450D35797BDF96264D141EEAC77550408F6A6399"  
02294 "3DF3F89D1EDC90DD0885227C273546C2D8D63513AC3241AAF96C137B69970E273B73EEFB8"  
02295 "668B9946CBD118CD1977445F87557DA03668D4F3A6B5DE6D0088982F6C5DCF082207A99B37"  
02296 "6B155FDCBA8A5E47F2CA4A50D48DFFBE05F8C1D28A080147FE9FE04FD49C370DD9E4861D8E"  
02297 "553A22B7F324C51C20F10BCFC18EB59BE931C69232419F5E037333B52FCE4A138617B087"  
02298 "67506B793A430CB3EA0A4D9A2CCCF813625FBD2F471C39DEB2DFD0792F3374A980FA4B2081"  
02299 "85332FC68798DEA1728054F28771E0AC3FC065645B3E996EA729DC6A29CC23F872BABE6A84"  
02300 "911B1B6C256C38C0A62331EB66E2F8B9A77ED79E323B4A96728F0ECAC851C3E9C15F2863CE"  
02301 "6839D3752F4F206C72EF0BD60F087315E6AD3C7C836FA3C5CA6BF2B72D877E33C17EBEE283"

02302 "EFD62F395637E784A8A43E4EA174BB501D2CC216306673B06433A05ED052860666A3C3D7E7"  
02303 "D7C991911A56100A31048BB81FF71110697C941CD0ADB08705A5A73CE66AA87008B08B51DA"  
02304 "BBBC3FCE66B1113584BCD8CCDF4D010D7B135AC764B678CF307154B5A96D1541B75FBA65D"  
02305 "7A0D5C2BA2D1DA48BF0EFB9BEFB5141AA6D30A59695EF7FFAE250AB52DF0969FCC349E1C7B"  
02306 "8D90099EF0410A861FB05E37D7156991FC65542B3495CCE223CCBD383DD4432113B817B0AC"  
02307 "4000F9EEC55EB377CF48B0F51B8D02359449CBAAFD6AB69E3616D6A3806BF67069F2572D1A"  
02308 "A0E7DE28042DCA01533E1C381449E22A68D901424498DEA67F0EE80FAB6FEEC56BE0195096"  
02309 "9550F01D62ADD52396E927962F209B262674D2F079D0E48D1C994864BA90CC4CCF3CEA06A5"  
02310 "995072904CCD97313E82D3CF76F24E3B3946D11FFF45319DDFEE619D2B5D777795567A4476"  
02311 "7F12B41D84A4BF738A2C6AFC0DA08356DFAC343EF090176425CDDDEBD1264871CB7AD526D9B"  
02312 "E4E469E866817A1ABC9973C598B1724C8C3185DA3C999636074C27320694F0BA758B0842D2D"  
02313 "3EC93DA9A9CD3F65B321E378D95F003C744EBD7BCD4712B6FBFD084D9EEF0C84C70651E3AC"  
02314 "D999D64BCFB5E8F7D9582C290CE59176072A5157B1CEBB5684D47271ED482B9FEB7FDDE78B"  
02315 "424988E88627FB1299AB8B96CAA72239BE6930BBA4396118041D6F3CE51CEAD81DA54CFDC8"  
02316 "BBEB0F6C8E9FAEF93E2C404D543A0F0026C291CE992CD71BCB0C6E9C459143D995C92A4A1C"  
02317 "9A6AFD36E1CA87E1F8EEBD30EC9D209BB3F6410018", // PRN 14  
02318 "0245F1FF6639D3C45134190F729CD06608426830011170C357160EAF351BB00AAA61F254B9"  
02319 "39D89D335881D5D15F3DAFAE92ADAB2B59DB7D3EBFF526D9A6F44E1D8DD779753433A9A32A5"  
02320 "41A798805AD307787C1832385D8B11DE57BC3ABA2A81B70AA2221C8C9247EFB3A40943D8A0"  
02321 "677F1D4E6F9149D97D6D83173FD3D3114293D628AB97AA4CC0121E2A88BD5B5C290AE7E5EE"  
02322 "4BCBCC1CE73017E3C58FBE575701C919ADE5E94208789CC2BCC330719F022C1F79880B1BA"  
02323 "4E377D486ED467BC58B13F2FF97454BF1B4F4DEFEB3B5998FFFF604EA0BCAFE859CABA133A"  
02324 "F969827887AC252B9D21D8AEC49A0D77FC83B4D9A2930732E9A22B7B823E302E469B52537D"  
02325 "B5341AA0979642F4505151AED71B690430AFBDB09A8454183A23F8F6598719B86D1E3FCCB7"  
02326 "2DD6599C9895CF6AC28FC5B1D4A821A6F9D9BB7AA63DCDA1678AD816D96B919E47D0D0E95"  
02327 "756BFDDC71AE430CF71E521701452709B6E670306A20C2F60F84D6009DF6D749C5E02D3DD"  
02328 "5495DBCCBE9326D7302E2A56032FA4CA5C82A107E06A867D09A2E5DE7DB9C250E25B5279A4"  
02329 "867657AACAB0E3D601DD3BE26C936BF527CA9EEBF3B247F75487A01694C7595619FE6E90AD"  
02330 "78BEA799B0BB0C126ACF3DE54F805725CA01879BA5B8412D4FD3D3CBBB37AFA955CF90248C"  
02331 "C521B0FDBD214A8D4C9E4A41705714C8DF5EBFBFB2FDBAB93707B0D779357E3DBBB20D5CCA"  
02332 "E707BA7BC5D21D4C23D72AA231E83880867FFF3A6B587EDBF5485C516239028271B5462F78"  
02333 "E3B226B057F01DE2D348C7D08108B990DE8484E839EDAA174B3CA0DD6D1338513B3D1E4ACA"  
02334 "BF9C14A9E03CB373D039645B6447D7853257F74665CE10EAF27DC1640C0AF88FF4C207994A"  
02335 "A23FEA97D1CE707AB08E4A615C580E3AB971D64371715CF0BCB8421D8178D708E4D3DB24B2"  
02336 "AD3143E2176B4BE37DBD8812116E134DC9B7E235E8811F7F404485ADD7321D3DFDCED23BC8"  
02337 "209A9DAAA198307EF1BC0E204CA42D7274EBFA74C88251398D6E479D53C5C6E7528BEA5BED"  
02338 "C8677F6F5A81A07BF7CE58E4AC38387824300D14FFF131A54390711F95E03CA59DFECE3EA4"  
02339 "696643A16874F98B76F6D280F7328AFADC455C51712BA748D33741998838D33F65F0166BD4"  
02340 "D9788B129CF7F9F0802758631E5A718A5058C63B0D38D9E687778FB8CD8BB7AF099ED442C"  
02341 "52DDE7D3C0746BBCB8595C35C815E80C71E98425FDC629C7F8084BA2556CEC31E707408BA6"  
02342 "FD7A2F0AB83286A93E565FDA34B7344474EF48FE5C99017887C8086AE4C89169141421515"  
02343 "50B0222CF6F617817EF86920D1FC00B9D9E5BD95C6047731373DB142398EB7E5F1B29FDBE"  
02344 "41ED1E9BE725C2748EAA2F7A1471AAB6C15BC4596B6C6CEA5CB3C9F1E57179F96B3C3017C2"  
02345 "244B179C384ACB93557170FAE0CD387D1D3428A7030B4276F8D3FD38074E9348033E3BB3B4"  
02346 "A05B3366064C6BE0B25F36C2E45FB0AD4606CF5A6B8FEAD45CD9790BF5BA1B0ABE5C67196C"  
02347 "05D637E878F6D3108B36ED2CBC007F6BDBC7D74362085D651ECFDAE47FFB069ACCC7C55B72"  
02348 "6C0C6ADCE2760DD3509649DC58189CD7195B11245D76F64DF2D61B07E0C172313F39961BBC"  
02349 "410E4000C77BCC52593AB7915D5B5B3E746CE8F15AFD2AFEA8200687BD1E0D767ED297347"  
02350 "A1B0634A4F006033322F787FD901E7F2492E1B491D4F5B901FF12CA88B2B2B2982D7EEC9DE"  
02351 "C4538F8E946172AC82CD468812EB9E7062AED4075AF0A587A92E9BD971637B3222CDE1A896"  
02352 "5DE1F2CD4A3F2B7457D27CD328016FD2D92B0560", // PRN 15  
02353 "EB01608B4B6324D02441F4094E82D3DC8ED644FD9D7FE96DC544276D7E02568453EBE37597"  
02354 "4990442B96B85A9DCF163A5974EE96274A6AAA9EEB0C83AF9A2E27C4A4174EE1D4A4035782"  
02355 "AEF83774E93F67A5A7B91E7C8E05611BA6747FA9F83A788DA50C9D4EFA17BCABB0361E0555C"  
02356 "E6D07DADE204EF601AF3D1C7A54AE1011C2AD46C27FC060A27073781A221D182D4AF531D30"  
02357 "56401D61024A92FBB0030B8340E91AE3A32468CA48183567199F34A76209F7846ACEE0B5DE"  
02358 "327B3A381087204DBB6FF40E1463CC7BD38311DDB99FBD05BDAF3B84D9E64958F67421CFA0"  
02359 "9903400DC87E212FA320DB95DF83AD6A6E247E92151AEA0638D3755A6B409E5853E1A7FEB5"  
02360 "4FEC711D75AB9CAB21DDB34D7115174E01E9F0A3E650F852666D06A2A8E133C872FE9034BF"  
02361 "9A61E259CC31925B09E40096B6D15809FCE4C8993A056E4133BA1FC05933B643B5D476D398"  
02362 "7E7BD372412852D439D38600BE4A7F22BE1777CD86F70F05DF3B260B41DED95653243419C5"  
02363 "BD9B6288697460A0FF7C0E2928D707BBA41E46D1AE11BF3751852DA1516941BF67E5C17753"  
02364 "83A7F88CBA065BABD2235C155FC69D3C11F5658B019B089A9B3A983A2D2533BFF7694FC271"  
02365 "597C42EF35F8AEF804E18AA37E3F42E49D23FB9171A312A562BBC9B093D92D45EB6A1A8A66"  
02366 "5257D956174E344E117397AA7C477FC1BB176C8284DBF4558B5259FA69836B68322705C"  
02367 "F783121E3C4513EC11B76329F1310731CEBD2CC8B7F49AE8D87BCAC07B1CEBABB872B5F83D"  
02368 "71FD10F300F9FA30F4054C476F8EE75AC68F1F80E352D3F9753233E53F67C3917C264F80E"  
02369 "8A61AC0299FEBEF95D9A46BBEC9F5C16B5EF127EA18BFA609B99B0B362036F4AE14D11BD7E"  
02370 "C8ED2B02C8BCAE2CFE409899E06DA304E23305DB1AC0A7746A72746C24E6D073F1FFFD3BF"  
02371 "AEF300299C8009043FA392DE2054F237A098AE5FF0E537105F25B7D8535A44FAE707B0A3D"  
02372 "1D6B30359CF1BD6EA3BE57B3A6147B943D1C96C4E2C871DB1AB64D9359F1F3D15F9DE5F45F"  
02373 "E8E03269EBB795AC78F47DA7B913F09402EE65271859D623908D464B1B5E59E4580BE30E0"  
02374 "1C553DCC79B190A809C71D8204A735350C75C3361EF3C89C07B84270502BAB82FF08979383"  
02375 "81B3B041A17AB396C09C4BEF716329318B95C7C4779889956961F8D2200C163C86F45C1FB"  
02376 "A0567C8615F4E815AE8CE2E05589C006B4560D66A67495C2B99E3A951500FECF953D9FA7FB"  
02377 "5843F102FE0F82E72867A65C09318A52D4634A3FA28B032B92160CB2E7D6ED156586D071B8"  
02378 "CB147D139212BC44774CE580E80791C14FE49DE7688B3A99E966A9F869B0A7421062F18705"  
02379 "AD23AC238B52E6D80F43DCF2B43F66965F6075469E0E789B40687D104DDB5CE5A12C0B741E"  
02380 "DBF1A3E57BDACC92265D592FDD6370E354B78DBE9F13CA0A4B9920EB740FB220E01DC176C0"  
02381 "9CA51EB905E32398E739C6A39C02043042C5FFBDC9BF959D56842DB894FC108AFC5A1039A9"  
02382 "B28D19D67127856E29FBA3812F764D31AD4E2FBA5F00A828F85087B4A5ACF24CFA79A3B60C"  
02383 "954DCE40BA6BFAAE4466C88A60293AB46FA2FEFEFA2E3B0DAF0CD72B0CF914848AE6A9BC26F"  
02384 "6578D08E896479879D3469E0CF39E3EA6508A412853C83BD33D0B993678FA08DF14AB40F41"  
02385 "CEF0E74B992DAE6F884E193B10E5136DE079BF85936D3B5A586323B08FAF8A39B24ADDC2D"  
02386 "7E1EAE1F2F90A39F26B130F060ADED352AF085F9169CAE21093FB2256C4E17416F57D29CD3"  
02387 "7CDE3BBEC8CBAB280DB9643C86696C3A6D3517150", // PRN 16  
02388 "ED28B364884EDCF3082EBC5FCD11FA0B0823DBD808FA19BFCDFB42BB1BE2B73D5D1434C95D"

02389 "D48DF1ACDFD7DD17C741A387F18E7D78B2A9A894266167897AF1C8FED13D463B6BACDD81BC"  
02390 "C2F3BF17D2DFC0D43DD04FEFE39C801C8917D52BFE8D2E6F8C0C5159E0185805BF9DF9119"  
02391 "AC97F644889189028D50B274848B429AD4DE45194ABC458485617B59FC5B1F1606BD20283"  
02392 "A73F41A6B36731F6F9E978B276B33B5DEA3BF87282D21F63787BEB1F6FCD3B21004E6FA793"  
02393 "30A3345F2C01C8CD1FE661FF9AB5B484B3260F2332FDB00216DA42D2840D3180B6521B0128"  
02394 "1E6D0F09664675531EED14A5C1794F3ED1FE634EF3D6E9D1B6F4871A45A83DB56FE2BBC52F"  
02395 "CBC1E05FDCA884F98DEB436A9A238E086FB339E1A7AFE8E04A5110737F021C89EC5DCA1168"  
02396 "E5CCAF92A04C950154F6209CDD01B287FC360B93E2A5661E7809FF1BDF3F71B40BA7564118"  
02397 "94B340CBAFF9DEEDC8292E7C316ADF4F3B827F1B82A406B53AAD8D2D660FD0CD2D6A6C693A"  
02398 "73511FBBB97612A8CD3F0E177EE38D46100E64C4C540C834DD87E062DA2C6C3BE8A0BBC46D"  
02399 "2070FBBD9E1DC396423D2BD1DD22F1F4FEE0E24F48CAD9241990B5F48A50B3B178A9C1CA71"  
02400 "F71321226E98D1F79DD7F3CB75E0B0C16915E34A92750BE9EA173C276421CD0E54319ADB0C"  
02401 "4E5ACB4BD599842E5028F613067083522F0D2A891BFDB190855B9BF36A7F2EC81EA89AE11A"  
02402 "37B6B89126DA7C51A4BD0C902D0DDA55A4286E36835D2EE95EB46AE7D6B919C05D42E543D9"  
02403 "6D22CD241A8E6985A18F29184586E6A7C10D4A955D0A1983EEBEDC5A9FDE306FC366F77A25"  
02404 "CFC10E5B8732D8F40D068224177288E9A13408F06AEC841C2E8612E6C794DE216A2B758C67"  
02405 "6022D4C0B99A4A54166CA1153F43FF66AC03F183DC30E9C83E6EC3BA33F63F262F39838E5C"  
02406 "0446254B3C1DBEB7ED957B8F7CCCA6C63B837ED93B2C260F276175F2F95B235D4D1DC0C901"  
02407 "4192D72CFE014094F29A0AB8F0A9BE896067D74BE5489F7ABFBD7EF442579875CB17956699"  
02408 "91438C0B6FF13E06D5BA805DD2D0DD06731ECA45FC8CB9D7CE568921A60E68E31F31392EB8"  
02409 "2A39982BDFE8D953DE2CE0AEFFBF0243E43ED202DF6FF9E075A90684FED84A4334A8E70997"  
02410 "9FD5B60BCD60D41907D1FC5D927F9C9CAFF2DE85AE1350C8EF8D68EAE38EBD0DA0D252F0B5C"  
02411 "82E1C3EF78CADF92CC78BD990B06461BD77B78635C406B4109FE58850D2A97A6C23CAA18B9"  
02412 "AEA535344E41E783B76A7A439DE0762A23D18244E76D3219A10100264EEFCA45F19022EFE"  
02413 "72B856FC34E59ADF6DFC78D31C8690CECFA45321AEE33DFD40943B158BCBEA764C3579FD5A"  
02414 "06B6297D363994E6B959338A3B48F14A3C58CDAF1D91EA1711FB39CF78F04E72B293B80847"  
02415 "516DA41D5024CEBFCBB399393E59FC4C89BA310377AF576105CDEB4F6DA314E990B1513BE9"  
02416 "DF738AD35B1460DA23326D2FBD86722DEA5D37CE5E34C11408C01E1881BCDA250C7FCAEBA9"  
02417 "DF85AA08FC9503546A55F2B4D67A0FAB2F50CA8667C7456DF6972273B62A0DAF38BB7D38"  
02418 "B5B62F2F1EA8476F300D0116598725C0903B0323779E02977F33AA8C45C2933B412490EC1"  
02419 "FFC2AB2B4130CE8FED8D34419FD15971EEA5759EEAF9F3E8FB4FBB26615B4286AB4C2AC57A"  
02420 "6E4BC8DDB396043251DBAE26ADB03767DC9EE4ED3E0FC7D74C58175BC1F393F0CE5D16589"  
02421 "879C6D53645B861B079B298E056E4539B2E2E70920D3A273398BA9E9247771B8A23B276DA9A"  
02422 "06945DCF343E7A4CE7B06960A1F624A75B1B48AF0", // PRN 17  
02423 "CB9F5B3A549646C455D2FBDADF2A79F9921C54C6FEA316D7713D64FD790325A87A30DEDBDF"  
02424 "0ADE3B7BEA52A0E319A57084A79DD706228989642E78BF51B7F1246BEF6F1B8724C57B0A36"  
02425 "33FCF4440E9D57DAEEB343283D642710266386DED8FFCFD68B32949FAC691E66FE9B8D736E"  
02426 "BCA6F4551B39E79557154E9AF590B6F5CA1C6A0ACA82FF4B839B6F9C555CC57255F96B2C96"  
02427 "3536E48EC9E85BCBD270110EAF7C64AD9DAB53BC791CB11189117A34A74B4F23A8573A988"  
02428 "6BF974528D17B31CA9B6A6B7EF7F923B8915B640C599D2453941CD16633383F36F9849C016"  
02429 "5D94698D793A58A7869AAC8B4C007CB9E43D8DD100E9504B1942CEABBA733A242599D5B8C6"  
02430 "0AA42FC48BE7F040877DA2648D083A2272A46A90D215D96DB2FF1ACA6CE24A2936C18B7A28"  
02431 "3615E8E6B729602D018174276E2ADCCB869C43879B6BD0CF22CC0601F7491862971F65D3E0"  
02432 "BB2474549FA18964B14D9EF81429F08383F77E8880E858B3F350076B709046284A4BF500A6"  
02433 "C7D9D694F03ADC9753852285075197FE5F30F5FEA91C7740AF38CBF5B984ED852C99FD2711"  
02434 "3F690CA7BD5CA46DCA330A7E30E7AA3AEE8CB62251DDDD62DDCC29C94D79759357FD928552"  
02435 "79A851ED0B07D8F2CF6509DEB6C180AFFE120B1C06A4DFB1CB76B9DF3113B08BE85A50BF7"  
02436 "6480D2FE0C8B77909E0D146EDF0D751A1099EB2BD019525048BA145C7E0B3FEBE21A13"  
02437 "A41A9BACED48E1A4ED32A57FA061243EE840ECA69191B1BCDC0358A86DDC98920C45F1FD19"  
02438 "3D529C85B33E0135DCF2B81C7FEED32E07A2431755F3EF3EA685577A6884041C6600CCAE0"  
02439 "E4182CF4C8BE7F040877DA2648D083A2272A46A90D215D96DB2FF1ACA6CE24A2936C18B7A28"  
02440 "7F6C145B7C769D9019AE1B8666834B51AE95BEE2A8D606825C7609FD3F24057425B27D0157"  
02441 "3FF353771BD47902A9BBC41B2016086A3F9E366F96DA8AE3C62123DF05113980D8FDEAC5B2"  
02442 "C82BD44C69F2D1420BF8E7913E59FCB88845A7FA6A84AA296875CA6D2FA21A4FDF875B5CD"  
02443 "F0E1C0D84FEE983A469FEC80F02923695640FA0858E9B91812DD4D91AF0DFD97655C7FDC60"  
02444 "EFEB19DB58D7D7D0A5F93FC88A2AC1205D1BB60DB2F3BCE203C802752350B5E8240B6FEF4A"  
02445 "38DBF35AE5A68498DCCF1BA047DC56D67F15E3C3807BD5638A9494DDC268378DDED39A910A"  
02446 "6C7A59E6738DC68D1A531C7EFA68BE06AD20BAC86BC36083E45AB0A314ADAAB741C6E24A53"  
02447 "0B3E3FE31E4A1B3827057F944918664A96277528D185E821093D392012803ADB63A4DDB14"  
02448 "983B844392542F2C5A978F1911A2720E201FBB862AE98C3CCDA720A244B5B58AE601CED81"  
02449 "4B0753FCE5F51E7ED797AAA98FC833EA272D86A238524C07349F5DD0BA4A2F6281B2EC2D2"  
02450 "9EB546EF52F07B832053AA433A7E8BF81C6C9F562B603421787BCFF621A69ECA5CF07CE851"  
02451 "2BEF945DC744AF403B9DF0FDD48AFF264DAAEDBD8702863B5A345A733328466B22FFE820"  
02452 "906543639F0F6F85F3F7DBBD28034019F4D517676C5D8C8CABD4118768B10158D83BFF569D"  
02453 "3B71F30AB57B833C750674E2290C950E9D48D6527E885A0F7CF580717CF467759CA0D41C0"  
02454 "3A91B75648DFCEC55BDCEFC2EB2007AEF2F01651A328DE030546933C1BF88F34AF14DC01F"  
02455 "0F37FC8FF4E015DA5D25D4A1D52EEC44DA8E832F01BB1D9FF6D067021D4703E8F2D51E173A"  
02456 "DCE564EB53A048864D191F21C759F0B629E0053845D704990924832BD7831CCB80B36644B"  
02457 "F23EC71847286C959BF54C6CF9EF075B1EB7F43898", // PRN 18  
02458 "576592F533BF0DC0AC52CA27BF57894E407D8A264496F2196DEAB4706CFDB4E108E07FD18"  
02459 "F5B2253EC3BF1914BEA0E7B8EAC0EAD4C6A3D2E0162411811F8579262AB9F3411327C320D2"  
02460 "7C7DAEC764EC5277EDFC2635ED39B0C497365B8C3633626E66D83DBFFD8E24BDBBF09A9706"  
02461 "1D7F530F69D929DEB7AB12488951D753EFF9F8C8DA17F34B17ABB9A1955F40A4EDE9D058E"  
02462 "6F48BFB1B5533D91B55D333CE4AE379E124CFD375D82DD97B682D80704EFA3DDE799DC6F7D"  
02463 "E55B72A99DF2A89159CAC16B47F297A467358EC67A2EC72A01007C47D757DFE274893BB0E1"  
02464 "41337105C46B159B08A0414A7EF8DE0806F90B5C5A9362E837191DA4573B473C6E435462C"  
02465 "5667F75721370F036BD6CA025A49C15BF63A8B30AA5DA01CCF9746222F2113AF119DA437D0"  
02466 "CE98ABEA322483C127CFC4AB990496CB53BA7D20255E63368FFDA40BB7022A1FC115B91ADE"  
02467 "56B2711DDBDD89FB00C5FBE928F3121420288CC9001F865D44ABB4D78B3A1F2BF6B1B2C25"  
02468 "1AB1075A65A233C4472A460FBC76ECF191313D700F73BAE29795EF9A5FBD0DD5E2E04DD17"  
02469 "2C85E15CB0A63EFC2A4E317C5FF01965FEB3BCC81CBADFB201C670F4C8AAEE2E7E75F1AC4E"  
02470 "4AE0B826FA9F87B300188546E42F3F142AF19A4F91192741ACA71EBF29A6E1BB0D00A45340"  
02471 "95A8B988526F27DB9B676009631114A2EFF9B2D4C10111195F0C178D2634EDE70B4C0C876"  
02472 "13CC9344BF2F62E9BCB42C4CECE15AA61A0C7915E51A5645609620695B4D0C913337DEEFD9"  
02473 "195CC9545AB81AFEBFE5F21BBDF2F140A4D146ACC25B8332861C2EA55BA75086FD948BD27"  
02474 "923BD2590C62389CFBE89498C82279258D11689613C21CA8F1788C4926EAD381C9CDB41D0"  
02475 "9563890433AC4BF9421475EE42AE5735D1CD10653713BD62299F66F793FA095B2E04EEA1CF"

02476 "59EFA731143D5849032DA833E1660A95117A2ED97DBF4D22FC0BEC0B5CB8EC32B63BA18B63"  
02477 "F737F3ED775B6D91E052FCDBC0C33A6F4498082542B039A26BCB25CFB1B23088D9BAB9DB12"  
02478 "CD18BF2AE73C78792BA9096AF766114337FBC06CD63BA8D1C0A5E5C6BBF5AEAC38D6B4EC29"  
02479 "22B8ABC930F58CFAF2C65C43422193FA4EB436C05AE3213CD19F58CC7623886525FDB77DAF"  
02480 "08CEDB5D48233D63F1D6BDA7DAD15E817775DB4C9DFE754F4AEB2A94DE6F883961DF23801"  
02481 "A645906BEEF9128F3F6C9FDB57E6D7B5EB46326C4C39F2C0694230CC28852D54281417C4D2"  
02482 "4470B9EBB625AD1D144B4889D23EC975CAB65A855CDA704128A61CC7C50C5ACFD6355570E0"  
02483 "7CE55578E4453CEF4582C368E585CE0F08BF786CDC2C0158E1A10971AB769872C52DAEA5D7"  
02484 "C8947A2B6300815B42125A7B8773E6E8C79239D32D7AF37CB9D4E8688CEDCD71A213795B0C"  
02485 "1BD5ADB0AA337989ED3E4E65DAA164C26946978C9468EAF74617C47EE4328C51F41418F2D3"  
02486 "5BD33D47A9A1ADB7C4F5A968946A7A900C293C65648C3127EF62D58BA9995B19B10D931C0"  
02487 "7D67164B2E685FB840A019EF4752602DC9317714D89F6D931825EDBAF002C40D5721525AC"  
02488 "0D2BD2B4FC64138F0FCC5EE648E84AB6908C1AF71408D51D31F7587F1BC8E19D1C045D8921"  
02489 "B447CFA9287AD03B4EDFC3FB85C7792CB2C479BD5A14B72EB8D3077CC511B0A7364C80C32F"  
02490 "11CD1EDB4B3A0806BBE9D7765B091E04F46620C23DB801A46C544A935FEE00FA07927171D"  
02491 "37D3FDAD3DFA81C40CE60F93EB65E667FE032A79DB745EBCD7EDE57FD51B647CE256244441"  
02492 "99E2D9647FAAF9F43554433B25CE78098BB33CF24", // PRN 19  
02493 "A888113D9682BAFE58B278C036FC2EF41CF8C156C39D67C378E4E0EBBE750656D67250F844"  
02494 "7C420E6EAEF139A11EC90B1B50CB582E5089694A4BFD843602382C022085141A7B8AA1DC5F"  
02495 "68469A2C2AB04BBE91EBEDAF2A2A19CF0538AE9D47421C3F98084073F340F0292B894645225"  
02496 "1630987DDA49E6A79F1562C1EF57134F036A5DE932B2AF9ED5D24D28463DA55F4E7C29045"  
02497 "DE9CFFC4372F7A6AEBDCBF35DFF2D952D90BD4650FDB3BD68F25B65A7876B65C5AFFE8B9CB"  
02498 "06CF91B88A588136C80D9CD361AD83F25B6D1E17D161817E64CB06262AFED0417DF26B9B3C"  
02499 "QAAB0A06B1310A7375352C46A410E167366AD768559A6DC618B416395F432C0CAA8D10F88"  
02500 "E9765F36B196B37EEF1F1C13C4AC08150C9E4CE45D9B1663FA317263672875E0BB5CFFAE2A"  
02501 "132A6B431AA7099E47E7FC8349D04EEFF0B84C3E8A743572C4B1EF38F760656E49FB1A4DF4E"  
02502 "2D1F6B24090CE675D299B32FC5EBDE40BBC2B1DD4CC0D1935F1FC084749DF9713DE41365"  
02503 "D8CC9AE5041D60967B25E16FF1941A9A7336DC2D46DF5AB88C14A4BDEF077EACFB1F08B834"  
02504 "AAECFCF75B5E3C1BBADF3EB3C73E187D862411AB97D694B2E557C47E5093FC15780805518758"  
02505 "B4C5B548306F92B559DC58475464C6DCC18D2B3905FE5F60B4530F87716107A1A4EB4A0090C"  
02506 "23527895582B02141386A726FFA11E9930B6099B22816DEEA2626E60F4B17C5E175A852A2E"  
02507 "174D59FE8219C190D0D6C50C14A3C5215BB193FA56209CA4D14AB2DB5CEAC2B3B20CEE6DA1"  
02508 "F798B681D2DAD8DD13674B9156E5C3CA4A09B1932CF69D62621DEEF990AD264996069B01E"  
02509 "6F7A21F2E5996251D09FD955CCA7DAA902FA7D46E03AABD4FFC5A1A873711FBDFDEA0935C3"  
02510 "99A0178CF2516FD03746B86B5A80576A34119373EA7F059A0EDA3F0D18CF33ED63DC27C815"  
02511 "7E0C43C537E86D37E58EB5ABEB5524FE8E19B2298B9DF539C7C31FF0F7AE96A56FD3806B8"  
02512 "1088D8C13224131998DF42066802ADC2A08AF6F69D36A7B9B518430C5532D3AEFB20BE0BD1B"  
02513 "F4B2C968AA78331C614EE51B8E677B19A98E30756B39BB6A1AEF0E91915BC91D98E86FFE75"  
02514 "84BA6E8D6D51A132EBCL24C056FBB9363B8610EB3D7A77A3E581C02E7D25184ECE47A688E"  
02515 "718922EA672CA9567BBA9BAEC81CD8DE7FBB8CA04D20B51C850DA2A4AA63E87FFB27AC89D1"  
02516 "8FAC40AAFB21331AEDABFC354533E57C73CB7D6B601B7180E4489269001A6786415F4D1AC"  
02517 "D8EB8C3CD8E78F6577A9E107ECBA3EF5EFA242F80202C1EDAD26D9FDB8F43F70D250D899EA"  
02518 "AC973042149B80BF7327D72A49222B5C403645BFDF3FBAF78F2AAC71B9FB582346CA456EC"  
02519 "183867409C18E11A1BBFE6FF5B681B0F893778C87A9BE8A67E196919945F4F22D70FF5D85C"  
02520 "55D23893E70FD9D685C41B28B118EFC905C450A9E91112C5CEA2475EBD07DBEFOA125C298F"  
02521 "467377E2BF8D4BBCCA25A2DCC4AC83AF877E1FC820D09F31888802B3226FB0129AB9E47FA6"  
02522 "5BAC527CE74BADAF59F95CAFCB6EFBDC889EE8075CF7C71518535DA63E7BC9DC566841712"  
02523 "3CF11DAB1070B050DEA01F46BD86E5C5B3175EC1C0C1244E0E7B5B6FB5D26979393E430C7"  
02524 "B79893E3BF819380D206438174038764B7196DC0FD94D4A9CB8382CEED4490C43C19061D2B"  
02525 "ECABCDFF386F1512EF3D635601E8E9CDECE9A452A0E6C6AE0D6BBBF3AFE226A46AB3285531C"  
02526 "7AA9A0CFD80ED9A557F20E9802139A7C8EC2B2872CBA3AAE87F0BC6A226BE4AA5681B14E"  
02527 "080A399BF7480E8583DF2505AEEE32E45C4112FC48", // PRN 20  
02528 "DD364986DCF106EED4C8705C1A9FDDF9480D337BE5C61AA3F661C6968AA01E3E454698FFAD"  
02529 "F366C73599B987DE85005C9743A2416E637E4BBB67FFC4D35F26B7539C33256EDBF680939C"  
02530 "C8986F5BF32D2ECE4905A4F6A1727BE1131771954856A1C66A402FDD0DB8B51E9C02E6BB1"  
02531 "36465FC425C8D98FED1582EF8AD22828B6A35580FD58A0038A9D97FE2F661F31265804183"  
02532 "996979C24C321D4EC8229C5753F399029B0BD5F065D5A19CB4548C86CCC1F3A9873FFB9E4"  
02533 "F421D033E4EED4ADBAAF302E24371EA97D817AA33C72708C1EDAECEBDE7C06AE0C95D823751"  
02534 "7666D0FB43B2434BC50EC2BC8DF1E3CB46099BAEB869B251E79E9B175B75FF62F408A5FCAA"  
02535 "4F6DEC90F9797B2D6272389696B5E38D26B946E0575907C358DB17D046D58D061683E07F1B07"  
02536 "5C10934FC4260043AAC28928DC877D27DB9BAE618910564EC3EDE3C402E86BC3981203CB7D"  
02537 "CB9940C37D948556A57A11488D1332B05F4E5757D470A22D0B86C88DC456CE014B673CF5FE"  
02538 "ECAAF26F4662C9B243F46969025692E445ABF375AECBAE50EC9E48AD25EE1D1D0C16624CC5"  
02539 "BCD1BEA8C8FFE3579C50B732625A381002115A1D69DAFF5844731F41AFCDE42CCF1A13FC98"  
02540 "A64F9769F03CA6F1B1C3ADA6A98F40343BFEDCD8096652292B60DE83385FFEA9CBBBFF51"  
02541 "450C768EC8CA36F6DF2F76FDA7D5886148168CE2BF693BE0179DE250D41DAA8055348804DD"  
02542 "B01FF97329E94C7DB33005B498FA8408E13FD5BBFE079F6E40EC5C8B735EBF9914B932A82A"  
02543 "41D9D5136C662294956CB52939F4F7E335A3B9749CC1B9F374AAC89C9AD1E718BC469CA21D"  
02544 "2CA467F5DED54C402BA0CEA000DCC878822EE3E1D329CE76E421305178FDB31F5DAC177838"  
02545 "197C5B46A4F96721139695EBBCF39F04BC1BDBEAB567E8FD812C355CF72785CDB3A6909F80"  
02546 "1BD2F448B3EA0C7F96D09DBE4EFFB1CAAF2A47FE6BB830FDB937CC5ED2B485AA6B85DA799C"  
02547 "1266562DF181005A345C62BF94004C1B8D85AC06D114C8F2C5E2D569C709251EB929237C76"  
02548 "E9CCFB9CB39206BE3C391FCF5E814E42AC2BE1576B9F0664B20F03BDA1F6431D1D1CD4C078"  
02549 "3CDA2A7AD32281A06B2BEC53EF1568840277EA4341F5570388045526AB214B477EA23023C"  
02550 "13F4327DD0C84CD1744677091B21D8F95F38A8904EC5D6542AD6E3A68B355D688B1F6CA50D"  
02551 "B718127B33D654B1DE41C14575217FEE44E5B631CA49995EC2CA787EEC4533657FFDDE7370"  
02552 "329C9BCC301B8785F7A307F1D06F88F21F5BD19F94827C128F799A9C354A021146985256B3"  
02553 "D3E8BE46AF6E370EE193BDA3494E0FCC7AA287C3A5B93B638AAF9BAEA46288DDF690A46401"  
02554 "7765FC2F769A5A930FD3E5A5F399B2F972CE286E81062B6BA85FC04DBBCC05B9971799225E"  
02555 "2F0867B55CBCCDD3B85D6D56C6948B03E91080FBCE3B2C5438491050BD2E81C41D5C2F679"  
02556 "0AB1F327D2C7023FD891F8D6E13D6F69E7AB1604692F21ECC7794F028A8306657D72B58583"  
02557 "01F5296A5FFDCBEC4863D0503749E9B483E4760AFC05682304EB17D67E016553FE9108895D"  
02558 "F3EA98890172D795009AB414E4539C47200C56828919DBC1AFFE040FB3319A068264607CE5"  
02559 "1D387B9291B7483CE1C38C1F059AA86B68BC7F852D21BF759A56557AF5B4E8CAD60356AE1D"  
02560 "9D11AB49CFB9F627DFB1AF5BF8A6D85A775B6676761FEFEFE6488C383F5ABAD4E06DF0B6C8"  
02561 "2BD2582B48BC1D4B27168525C69FFE8117FCC38FE758B2DF09E38BCD1688789B2AF3A0501"  
02562 "42E50787EBF2098C229E6965FE134593AE47AF7FC4", // PRN 21



02563 "B59F424CE3388B73A01C7279251C7854C664803D834EBE840CF3851B922071026DC17B967F"  
02564 "06E8467AD73F870440CF20178BD31F8CC315AD8B3D213029D08746BA9FC70AAFFAFE2F393"  
02565 "D0339BBF28FC15B8581667E9B43E0A1534AB70BDE8954B30AFCD4BF10B91B8BE2AEDD6489D"  
02566 "590E0947B6155A1E0B4E7DDB7D5DF08A340FE33DEF34330EC77E65B57FEBD4745B0A07CB8B"  
02567 "CEFOABA92E387E2C2F80230818DB1F70F65F872A2C5580A6C39B9EF600C797FF0C61641100"  
02568 "90385243D847C7243EADCB69F69D75F4B1D921C72A4B5B67D11B78F84BD9682963E07C5763"  
02569 "50BC301641E76BFAA23EFF0FDA33B1A41729AF935C09C1E5257F84A4AA0AA3A54525BDF1E3"  
02570 "A51EDDA6A35A2C2B6DA82E5C7FC989475D99585A9DCB66325B98C20BD7550871BF66979BA3"  
02571 "D58562C1125EFF230FDC77A3BE8A7A8CAEE2F83C62688E0EE5D72D4C858624C7362A57B7C0"  
02572 "065670DD8776D1E97F67CD4E4110E53E78BC17793CD34F11089AF317407B5F43E26424A752"  
02573 "DA5AD58470FB8FDC193D2B28FD4620C879BD22A24DE4EFCE4FC46D7A5A4350136F43D21698"  
02574 "AEF1B4C318156D574E93BDEAC252675BB1F1819B651D08E637538DC14505779521CBA1C569"  
02575 "5CE7C5F9BFA3A7E68E14ECDE399144E6D514CBEA519FFD2C677310214DF222CD79C9B9F763"  
02576 "060C48D8FC4E02DBC0D6396F6A28BED4ADB4F69FA44BF6F52E7516E77CAA981712F10E94D"  
02577 "AE683DF1CD858492A6B7C1FDC92B3A8A4B121B1C8B8791E18062548BDD31DC544FD94DA7BD"  
02578 "77A0BD18B3940760A3746ECAFEEDBB6B99562E836421F24C2F2A99F686DCF9C57E91074803"  
02579 "1521E359D21A69933D3B3173D520E7B8810035140A9F30DECFC99A117B8C81990473378C58D"  
02580 "1313E29A9824034B86D46BDF15479E8273BBFC4710958348B17C8EB8EBF8581AA9D1032AD5"  
02581 "91B7AB62B3E2DE12F36BF0ED71CB0DB342D5233243F466291DAFFD75971427F37549BC74CC"  
02582 "8EA2C9D61C55FF4EEF46EC74F37E30AE9227F9A346A5A379A386E4706FC6DADCC80FC7CAAF"  
02583 "6164673DF6B75051CF8C1A2C262F13A65CF1DA6D3CD277494735E5E0B4366804A816373AA"  
02584 "0A33F3FB3566D9811D26380EF7E049F004E5540E3A984793553B866E364BF3ADE51ADEF86D"  
02585 "37A9392B6D0D518C0F07F80E9BCF795C540BCF8BE8AE72C0A0A67160F296E30A0F5727391F"  
02586 "82DF95DBF2376882EE6DD6C87C05CA191AB8F32671C4E404734D6626DA299682D9A49788A7"  
02587 "D961B737282CF16DBC233124C5EF4800E1F25316763BDD92F090088218C071E91EF2D76D8"  
02588 "0BD05D23C6C06EE975279B0505CA18623A33A4D94EA882A6C0C5510986D537B49BAE93E363"  
02589 "833A8571F91C47F022A6E05FC7673149128E3876AB6F423CA96A7F0F4D5AC1682DFC05BD4D"  
02590 "3ED478218D40A42A894DAD908A587A83299D9CA685C4E25155760F53B27EC3F3E722E46E63"  
02591 "BA2CA0D353C60DBF985F3E10893A0BC01539FBE6126215B090F1355D211E833BC019C7BB35"  
02592 "1B1A51E5286A178F70DB39D3A8DB65E510A0A93FD717FF30BB586B8FF49D62ABD56F0AA4B0"  
02593 "1D2A8E6A3BFEA54FF1D36C8ABA385D0974C7F29B25BF12B1036E4926F0E70E6CDC93233E21"  
02594 "1693F7B315C15594790BFF888ECE21352C8FDF906DEB1085408C77A98BC5B145BB1D0945E"  
02595 "33409EB01E243B386492BD62847AA67D4EAEF5E8B84468F70457470C1FA15BBB44AE4D77BC"  
02596 "401BC4128654B52B2B8CA48721DBB267EFA7F0B2ADADEF8C2BE9516A42BD2FC2C7AFB8D0D71"  
02597 "FBB8A6D7034892078F4FE50A3B63105B31451A7910", // PRN 22  
02598 "FF81F6EFB1CF3169B63C2F2544105BE42B20467E1FF2E4204601A486928A77571247AD40EF"  
02599 "058003E423521B4196118088D773BD1694CC83C02219F5E16C31BBC7350A023E4FE072D0CA"  
02600 "713444A455DA08E5653B539A555F37AE9357BF538E3EA6A2C3B2BD30EFC32665B597A611EE"  
02601 "98DBDA9A9857FFA648CB875677DAB55EE8528DB18C4188DC693A9DA0D7DD736401DA333DDA"  
02602 "13F572D4652D23F9016CE9D39605B5E2F341F487980331D88511836C7560B9F3E8DF6FF4CD"  
02603 "B7F74459BA199E081FA49BA205B368B4D519B2D0ABF92405409EDF1DF280A04F398758E551"  
02604 "1B6E6325CE00C57186A0938974207E0059B2B8BC81156E2FC0A76E84C5BBADF4FDFBA379E"  
02605 "6F960DF58DB82FA306535F0A8C4D925246DF1D9B8C7A711554E9087A5FDEDD656CAE74357F"  
02606 "C6AAE8EA0C72B74BDA7BD4898C6FC7BC50367FEDE6D5261B0B72E171043C89C8C2149D0146"  
02607 "6B226F6A1F373FF378E34A39566C60F0FC8022E15EA68DE5DAC57BC370B30F9DDCA6404C5A"  
02608 "152831230DA9EC3662DE701767953AE8DDB9918598844202DC2AB41A0ED711D640F3C60A8C"  
02609 "C96BD8D9340A32330F9D27E7A023CC99D173BFE6CA4F0604FF6E5B4C59EC2E1BD4FAEB3321"  
02610 "F755B2494343BB6E647210041A1B96A02B7755EFB6B482B3C7122EEF9242FD64D0D7BE24D3"  
02611 "20FD33ED0D4DF2756BDB13BBFB622E7E928676D228DC395DC32465E0E3480B7ED2EDDE0410"  
02612 "1E9C0F8020D48F3D43A0F756622A3DD866B966B7B0BFC1F40B1D460D6A54A45ECCF5AFC93D"  
02613 "AF8541FCC62A58DCE907D2DA6397B3F7B535DD8E30D400610971E4960859DFBD4A441F2090"  
02614 "21E9E4B05D3EAO80A738F0C4EE9B03492692C543968122FE39BE1F2E5D682C1FB8549B12D"  
02615 "6F32AD5938FAF63C5F490EAFCD9D7DB70681B5650081B098D0BEC87013C70F890666AE984"  
02616 "C5FD92F6939419AB0C1E981FF6EB03B1A48D708438A8F6AF81515D8C42E0DA5E13DC07D22A"  
02617 "0CE018105BA075A696BAB7CAF420A65E38EA1961B61ECA43AA0085BF2BA6244EDCB56EAC6D"  
02618 "ED736712045911D7724E9DDED1C05DC7205580FDF52143C07960AFAC71528C1B87BFD9D21E"  
02619 "A5DA5F9649AB037CEFA5713647DBA70A2D0456A24AFB0B55B8C859F9C9617E83E12CD09633"  
02620 "97DCA09CB16E22A47EAD91206E8F3D925444AB85D76DE73A43AA45BD3B0412FBDF254CF27"  
02621 "3E5F20F6BD0D938E30ECD5159120BF82FEFC0FEA1DB898D5D1E674452DA86C9AD4AAAF0A"  
02622 "7AC6A304CEADA8AC82870313BBD56E746E7C9C183EF5B53BF5A374B955D053D1CD7E8EE2"  
02623 "5108BBB8F2A6066375832213D1B9064C9F12CA77835ADFE28D1A5E1E65FDBA45147F05294"  
02624 "8A507180EF12052D5FBAC2B3F4E46DBC4ECC7DE08D6FB3B6162432B25EEDA07911CDF4E98A"  
02625 "219603E19FCAAB142621F546C89818E1761AE7907F624A5D01E36612F06ACAA157BBC316E"  
02626 "9AA7568E359803A408B0C36B4593DA1B4D46D1E6AC290C90D19635394E10F0BE0B5116577"  
02627 "7517AF168F38B7D8B36D24B3CEC3DA55D9CF9E8166D99B971137F6845653B8CFBFA78B78E16"  
02628 "A8F64BEF8DBFC6DD4F6F9093FE629C99585D5945ABC30DCAD609903A26C181FDF48164204"  
02629 "3066E5B0561F13934247D19975F136C0E0F4CD41ED3F8E22739FE2D88358CAED503D7C36FE"  
02630 "755ED34B9EF871F2323AB593A0AA8BFD8DD3D86304E206E5B4FAA6B905751EAA846FB576C"  
02631 "3AAC9387E38A02A6AF8EC72BC953C4831FED1337EA65241E4D94F90A0E99018DA735E1CD4C"  
02632 "835AD9F301A618395C8FD4C982CB528256F8A4894", // PRN 23  
02633 "CE812897E0C0F7BA13D73227DA8D57F576A25F23338DAC16B5F3AA45522CF8B1C3F0C69F"  
02634 "6E1EA86C6BE0DF818820D7611EC41F6078EEA4461C336D02847EC3C5644B85EBA26B68C592"  
02635 "7D67E36BD381F8100B6BA90852D529419EBC8118A5354706D2CC769C2402E517A86809687B"  
02636 "A7E0B1940A88F0B2586484EB7D1B63E119D489A41691D0C1A88543D34D8962D17FCB8E91A6"  
02637 "CBE2635F4D0814A8C9A4DCD7767CE7D62E8BF17117A6650E3E8933DABD4CCBA43F40ECD46"  
02638 "EFC541CAEF6271E87BE92E63DFB7C2E5B7D7D2ABCD56A0A19E3C9F75894860A850029ADA36"  
02639 "F990DAAAACSAF56B157475EB18943D7CCB8D99CD14813A27036ED1F4DE3234BE28FBA41304"  
02640 "6A26113124B5FEA24F4074E991E163AC879520669ABE6B1A0203FAD6E4082CD5ECE201A86E"  
02641 "C87676E350021FB2A1E04AE9B19A03152F8244542B4D21B4173D05F159EDCE12BA12A57673"  
02642 "63794DEF4D390F56208EAA7680AB527C2035B065DD58AC8182BD5660DF75511EE1E0F407F4"  
02643 "EA72AB3FCB028C82D8FA3003B034EFF4DC1C179029C452BC4B4D75AD02CE9E393C15872F8D"  
02644 "130AFFAAE9A95640EC68771EBE628C42DA59AE4AC2E9156534430D00164E3D2D28665BD2DE"  
02645 "6AC148E2E34757E00F41078B672835B435711C560166CB4DF17B56884C8A66DC5E4D42AE"  
02646 "675335EF457AC20E4BE7FC32FEAAF7EF4CAD795ABA7F9A20C6703E7DADA30B63CD6CC21411"  
02647 "4242EC1A9614712271CC05867595D9938ED557007713F36D1A44AD2285A564C9B242AF6BE9"  
02648 "979F7C474385680FD574D9D33B8E2DBC138EA71977FF83BE8977B93BE3EAB2401A136E021D"  
02649 "A182B1AE68007FB36067507FF6900AFC4743A9A8503B7A2CEE04A1BDBE9D661691E5E8C6E5"

02650 "568D16F70C15D156F45DC41199A9C67FAAA36D8CF67672717ED0124E996CE855392F5A8BD2"  
02651 "EBCD644AE0CD5B1AD2711200130AA4F8D3B4599AE660A56EB6E192709CFB8890510CFB019"  
02652 "5D7FE794B9EDBCCB25ABE6F3B6AAECF3FCAB9ADD6D270D3D5B934C46C8D6800CF9617AA"  
02653 "B8192C28AB627EB0D6234F7C68F0F4241662A9F062B361FE89E92BF9254F823E4164AE5466"  
02654 "1C2991A56C4AF8884DEDCEB20EAA02D3097F2D5BC45FAD616F2DE3F37FE2DA9C838ECB3BF"  
02655 "649AB6CCC38675126381561D546B37C743241297476497184A861ED1FCC75BD508915AF397"  
02656 "52866B39F1BF00799F3B254C6710D15F5F79E29766E3EB75B93CEEAD0F3D454A3A0A4B9B88"  
02657 "534AA91B604424CA1F9801F7F16E0A5489CEA18CCF76537CE33481328C83F3205B59AB1BAC"  
02658 "1467C56A28DA696DF2B08CD380049C5E0FABC6AC8C5C50F9481161C1D60B0976FF459B2C11"  
02659 "679DF8239AF63C1379997A8CDD47954DA1A9ED1606A3D1D2700A2B9B0C9FEED839E011D527"  
02660 "52CE543ED224E5241340CFE1219A1A0117E033F48B262A6C83FB13BAEBF036C2ADD4ACC5D1"  
02661 "CD9C5C334F4EE7D964851D9A5B7CF5A38B3C1CCD6DFDB5D1AE3FEAEDFEB72A25C7CA779DC4"  
02662 "602FA0E6D8C4B35900F634130D48BDB8291B7BA751256EF38FD6FF0D5662C1691B82571444"  
02663 "2F5B633B30514058C778C78BBD6A70A2096CC9A375AA99CE11E8FE4388ED8D403EFDA3E604"  
02664 "EA7782DD4D7A1FE7C51B73F286520D56DF828B659FC63DE3DEC09FFBEC46FC143D943A096"  
02665 "34A91194D77F0A70DAED26ABD3F6B51457EEC61F9AB544F544B21A1284F0E70D75DAED0B9D"  
02666 "0FD864CC73236CB64EFC7C696882B1B408B6B02F35EFC4B5DCAAF600330C59B1338DC593B"  
02667 "C2286A1C4716D72E7BD622A3561F60362F3612580", // PRN 24  
02668 "DCD55CF8406A7CC9BA0A5224797D74777689C4FBC0D202DEA8B9773A0F20DB0438BEFD40A9"  
02669 "96EB2E3F49CBE2E475DEEF73F112FF742269A701AB43D8E47EF94DA4056086D86B1FAD636D"  
02670 "89929C60238133A5B61DC4B7F8DFC765CF0C8C7206F3E0DD3DA13407CA75EA20F172AC38C4"  
02671 "2ACADC946B3B18BBB30DF871905777CFE0D3BE5363574D2545D0214D6A43E2D34B50D7FF7D"  
02672 "9F2DAE7053736871FAAECFC2C355C1DA645A995981E1D617BDDD14EB9AB230D10085DA1AE5"  
02673 "5D1C9E55C225997D128CC2393D6E5BE9F1BDE1543DCAE918B38AFD5FAE6031E105BA12EA41"  
02674 "62F7E24572688C54291A61635C9DA26BB367348473D4DD4AEF9A5C4E8A6512AB149D62F625"  
02675 "D8D45576FAA4CE28E5C3D5D2293A0D5C48821E1F5CDAD7257C97B3C1A6DD83A4ABB55F780D"  
02676 "09964D36B6709D000948B89F319E1C2CBBB3EC33A6F0398880B8079E6A39DE697D76"  
02677 "803B04C230EE980765604929D60733BBB6382B4CA4B13AD3FC48AC9F944B97133C5AE6C3B9"  
02678 "D295A760D786DDBF346C87332DC3D2591926116A318F63835783A33A17C452AA6C620284"  
02679 "BA99DEAF2D397208AD2B370CF3016F0739103A518B25AE8D59BB017C8199ECCAF3E908F9"  
02680 "144C778E0F5F47FA06FE8A2E2E3B2CE3493EAA8745478D653F8D3711F832D9212D060A7F0B0"  
02681 "B62AB9A3C9BF012B61A9B5E6DBAF4B48D9031250419627E96076925AF95E2B238E8CD227A0"  
02682 "F3D8C9C3BC68D5990952D6CB62D3B8D193FB06120BCA9B195E92B995B86B92C455A6145D3"  
02683 "F77898E457E1B938183E1D76FF61C163057A4D70697507BE020CC28BE6527CE306A5D2506B"  
02684 "86152B5E3B954D5A660C79CDAABB2EEF0D135C0D4AAC7FC3EB5A3575D95070B168FCEFA4F9"  
02685 "B36FC3D55B84DDBEF10DF6BA659F70CB2AB82FA001D51ED0F040F9ADAD4129C324BE8B44316"  
02686 "C0E3E15B27C2A8D54A948585834C2DBD40337B61F58F3AB99B3303D56EC1D30ED82F29CF6A"  
02687 "68208B07DA120AC4073F102F4D796E38628D344384551810998F6211A082BA225A1B7D63C9"  
02688 "A3E8703D945BE1D979677FEFE715396988A873B88EBCB10209E4DAFF939776AD14C5DC90B"  
02689 "BE9C1A57581A145BE0D4BB47CF50721EFECB50C96E824DF93D675B1A1B4D920CB1EE82B03"  
02690 "69094DD84FAB4A3FA94F152F87525ED6D17A800EDB1BAA6C13A2EDB5C9B7C5A82AE5824AE9"  
02691 "BA63156EDF78517AE2B357C1A1E8EF6CB12DA415A914FB69FA633527B8A165929D2079B580"  
02692 "DEC210231BCA77465BC6A9516F2917226957DDDD9787313FA599BC2BDD14DEC9DCAF0F2520"  
02693 "318A85C55DF40881FF65E4AC21F2C9EA3CC06651309E1F89982D45FD6EB5BD588B647A80D2"  
02694 "7433EF8508A07AB2FFDBFC7D2B54B215E7D27A333D4942CD23D53142F04976093CC0344C19"  
02695 "F9CB3A5A4D522E26B7E2C1C85C1ABAE1B4A360945A46D097A1F3676A80E3BB9CBED0BD363C"  
02696 "B70C2B250EA053B183CFD1EDF2F6A211CE618F70EA1AB535E96CDD0B2FAB7CB7F0396B6DEB"  
02697 "59745C1BDDCB452FFD70C95E45B88325A82CF07EFEAD7176FAF04C8C3E0CC4458540BE99A"  
02698 "BB9F52AFBE48E42C63DD076A37C07C8E4EB6830FAC011936EC047F6D5B5857AAF90F80B5D"  
02699 "33B502A116828BD27E30B3A69F351E22D943A482BC6BC41BE7BF941F3AA7BF85A2BA5C38FC"  
02700 "930680FDE5A4E174AEC4156A75E7650CE488620D6CC62BF9D085FE1782EADB11541A5C19C"  
02701 "9061734513316B9CA608A6E48D333A110B57BA5128E9331E6266174B85F7B7D1DB264361B1"  
02702 "C6C044A8274A7C8DB19C253BF8DAB819E85130E920", // PRN 25  
02703 "79E4509D963C15D4930F17DB3EF8E9ADCDE82C9F2308A0C9F77359575876D90A9CE5ABBBFF"  
02704 "E09B8E145F35C7E91B141F66F74B3BA923D9E9CF863A8A455086DAADC247872C953476A8577"  
02705 "20797836F37C964AB1788479175F7B4E9651C7DD5EDA94F8BCFA57B3542788FE9547E0A44"  
02706 "4D249A7A633799710930CED2E93B1526A02BFCCCE416F0CFAA631B878BF23E41F95770315F"  
02707 "1E169EF22B9DAF0E2F269E06B03B125CC6F20D2AA9A13B98CE23FE3A7FBC70CD05496ABD89"  
02708 "ABA5BF58FBD88EB9653963F0452CA43CB964F7B145020D637905E8D26E8CCCE4EF892BC4"  
02709 "8381303161644FCA1DB9C4775E8F0A6367F1A842C3101C9F94730F6724257F2EE1657CC93"  
02710 "950C9CF20832DCC11E0001AB347781F21DDCD4786234C908772B553D7BDF4795FE69D7D55"  
02711 "E950D4C6E2A76C44995CCA5C9CD539ED235CC55DF6A004C1BCA852C2813DD17F333CF67711"  
02712 "EB8A5DA439D73271272D0D62B34DD457C7412A26FE2F4F23E5F5AE8C2F91DDF86B7048A408"  
02713 "72AB7C728B20DF7AE2A0E8E52B00899C565E68390DFC83344DBEB7257CF8031664FD93DB65"  
02714 "738013BA234035F11B611DEA4F74B6100724B2682EDA9639C11F4F4580683DDCA2D9159F63"  
02715 "7976293FD76AE2AC7EB5203766009577DF87D9DB654ACD64229495A5B4A94AB3AD8237B9C"  
02716 "FA2B5AB6B7239A5146C777CA3CCA9E1573D96A7C1B333F11D284AEB55DBABE6B4F12A421F"  
02717 "F7BD442AB6B8A7FC33A2087886328B33037EA78D9A7015ADA2C8F463AC1FEA8FF3EA886823"  
02718 "063992176C31152573C10516BE386CDF81A6568D8E3BA2FF195B23903C95B16277BD60D238"  
02719 "F309A7DD6BDB4ACF4541A56DF5D3D4BA4078E58641C0453B1BC8DAAF6F1DB28836D6FBA347"  
02720 "4D69870E07997DF4541BDA5C62291A3360F5C5840752A25CFE366186F7E4E21DA0704266B5"  
02721 "C14ADC1541E463917D5DA677945574F800BF47FA028E82200F9E53C48C887355D9E3CE5FB7"  
02722 "78341A356423D00B18D3048C92A80279C2785AECAE9EA57F828766D6F953AACF93FBE8A4C3"  
02723 "2C7FC9843F466141DD018CB005ED8A16E27B43C5C0EE16045D39CCDC7E4A27EF636408E968"  
02724 "B4C99C895CE77288C819701EE7184DFCE889328E4CD2D8BA7BE5A1DAE50E36D562BF2441B5"  
02725 "6242EF9C4EE63022E88F0F77E6094AD7DA93F672C06B6BF4ED5523D1DBE7D8F13892BF90DF"  
02726 "FD5D2A3BD08B956EEF03E560BB28650D873116C0CDD00148C90CFCECB6E9CA4BDD7603B5CE"  
02727 "9FC458431C00B1D704702B552D847D0C89265D842FF6599A150854848B730AB33CCF2BA653"  
02728 "3638860BA564540C6678061CD97BE3D00B103314894AF13999FCBDECAFB5A895671DD1ECDD"  
02729 "95C0C9A250B014CBB657ABE3028C8D6CE436FADF8FB7BDD12AA2DA32FA6435889F1B8C16EA"  
02730 "8488FD6AFDC7AFBF5358A31B23E0C05E315B60F180986BA17E26621D92CB639A0B815E65D1"  
02731 "40BA8A68F99BD42AF2428345AD3037274E515684EEEF018BA41A7B0138D7FF2C0B83882319"  
02732 "2FDEED577028052EE73D65E92812E4C33FA3CDC11234F47F702AD81F3829ACF1910B9D0016"  
02733 "7A3541372DB79996FC8F4B15F2A56B272B7DCBC4B074A36543B12B3A5396B0635068B62F81"  
02734 "A56B33840516A376D4ECF7BEA4F8ADBEAEACBDBE23F5F311A39D90D0B11BCAD30609F6CB4A"  
02735 "657CC620CF8F727E30F77618E870F72EDA78EB89399236F7F5432560049262E29A2979F83"  
02736 "2F2B57CEB25BBC2405A3098CBDC3CA0CDF631B37440A47D316F4860597AF78E5ACEA13E8B8"

02737 "B5420210D7BF54B1F07A433E7A2B193D7EEFDB2288", // PRN 26  
02738 "B63460020483CDBB47DB98D8D720FAD57072D41EDB5FF2A2CF65DFB70ABF3524003842F7BC"  
02739 "F9D36FE187F86B864C5FE0181E5E3D33AE11A50E67F56DE93387F5CA7D1641CED92F7195F"  
02740 "2ABCC7B32C6C7BD8AE561E36A6359BA6171726A2F9B00C7655C7E346310C6EBE10D2D470BD"  
02741 "5BB7B4C88378DBE2352AA45EDD728F42A3DF3B5E80BB6B6BF55CD663203437F575F769CACE"  
02742 "66809FD2593204387F8C51E0A6B2C4C6113DD45D31AD7567C9B924D5B10F124F6C1CAF345"  
02743 "8B3B84E9E42DBA9AC879E2D711ED8681207374CCD299FDB7D7004678D56FFFE0CD34B4858C"  
02744 "C0CF4F611D8EB5B47768D69AA69BB1875E9C397EE9453D37D4983359393A02941E772F9127"  
02745 "08085F2FC7211DB358EA5D4E00263E14A1CA3430A6E6FBCBA94CBC8281C356B4114B0E0B67"  
02746 "B3265D3E1B9A64B1D4BA5501CC0F6E19F56A7B3B1BDEFD8018340ADF7640C58FB26ACE46B7"  
02747 "88FCA44BF228A737B02F160B0462846DBED0792B27FC6B55757BE01FEAD3655FA9E992485E"  
02748 "2B29321816B04C2BF079BA5CD01CD7DC0CC8CFA471FAEC74C95041B7C62DC9BDFD11FBD4D3"  
02749 "B7173A3A92044C793EDB6AF4DCF760D38D3D412B3B292C74E08EF4B4515B99A0D4A167D910"  
02750 "3C3E776E213AC2C1614B432A8A7D243B0EB5E7AE82118DE996BB55329F49ED4A7EE738350B"  
02751 "565F50CCC312F274EA203757B372D542D451DB960DD8D7F0C5E532AFBA105C7963A394267"  
02752 "32E2E30C68317C4CBB3C10802C5104923E9EAE8C9C6F421B8EA3D5039D5175F1EA85497140"  
02753 "7C7B260BF57FA912DF3829BA91BEFC9A7A88E5A289366A88E7E6C5C4D4781D583CC15C827A"  
02754 "4FC753F66C81B5095D51F3069AF2E3BCBF01362F118690C35D3E6B5C0F41F4A4964226464"  
02755 "6DBD948FE2D1AE837C0C0062F77A40DD02371EA4B69BFD02803AC3AB972CF450F59FFDE724"  
02756 "F79DD7D534D261A44DC4BED2A47E78289BEE615218D1F861896AE5B07729568508B0E35F8"  
02757 "2BAD9387B5E22438456C2DAC3F82176E4859A8288FB8E2B86FFF4BEEF93D9F677CCACAE17"  
02758 "60D045F28C819A44F5F678E61585B74D49B1F12A9B28D4056CEFB98845385FCA12A6AB0C86"  
02759 "DAB01CFE8BF5D77618F797288706F18DD71B33F909CDB15A63047D0EC462BCC02AA7E6D4D7"  
02760 "D5C3D0C5FA2C03502EC51D781E8CDBB63A2578CDD9116701F7840D6A268DB082223F3D4237"  
02761 "6C5D796557BDC568791BB3A5BB6A9F501837CAF0ABEFBCBB500868EBE0FA0184ABFB927A41"  
02762 "4A87BC3A1903DDFBAC369B77B5C9FA203C7F6CBA7BA826867CBD37AFD20E32DBCDDBB7C1"  
02763 "2E3974BA0133B9ACF768AD4652C3FFB01D7E2F4EA026D543B2292419B10DB09C47C110005D"  
02764 "667EED5D066D4634175B58F4BEA53CC39F2D1A9623B7C9AF2ED29A37891F2BE24418746A23"  
02765 "3743BD340256801EE39E7C05932AA41B56F7DF4E479A365BA224E94D72A395741D3D813520"  
02766 "8C96AF089F2E2FAB29BD97DD802D149275727E12A2358F4CAF9DEF137397AAB54DF78F5CEB1"  
02767 "7CA323B696C3267AD2A944C65B2E8F542A2F0AC3DA7A9D6F2AF073FDED7424942B700AABA"  
02768 "291FB8672DF10AFDD154218C0B0088E84486EDA5F21615C06770A2607E93A2ABA60B864505"  
02769 "434ACC6B6B815BD00C3881E59F0B481653BE10929E7EAF4A90BC181DF76434C08F3ED37D86"  
02770 "2038232AE6A5C56F36F98574029AF4B4A280693551082BEE4CB6DD4B5E500AC4954F2A0112"  
02771 "9861DD2A70FC30FCD9073FD2F45B31CEF9724E673032D0C87B2A9FE7706040293B6DFC3A31"  
02772 "97AA58A53997BE5A2A763B027D5E14152774E3FFB8", // PRN 27  
02773 "6D562B96C4DB6F8783C945E358693B3D99B7D8E77AD50B7FE9F5176AF257ACF63E037ECE9F"  
02774 "C071CBA13080A08E1116AE09151BED458CD8BC9DFFE5744E0CD2365293B365A01D30E894F8"  
02775 "616AFDB611566FAB15522C8CFDE96644DA1AE776E0BB2912CF584AD1DB53E90043B1F93D"  
02776 "67D51B1B10179F0FED6EEB52A127BE857341CB5A223864C02E4AE3F3902C90BACAFBDE2A42"  
02777 "D73DBE88577F871D9C2EB9885C7CEF0443EE4A1C449951B24764CCBE716487F2EBE474ED6D"  
02778 "AA65E393F1AE17E52B75EAF245A161AA30CC8250631DEDD2B233D202CDFD1C3DB64FEB26"  
02779 "515F2568DD7C11DDCCFCDD77B9D1182A29AD5813791DD9987BA735082F8A669F38DD84EE7"  
02780 "C09334AC732849119D769DB8CD79375D1C8EB4E051777A057A4738FC4E06ABC70F8FBA6B52"  
02781 "948F91FAC2F2E248F0C417E2E727C60872F905DB2CCDCF15B2363696424BA5B62E98D5E4B5"  
02782 "1ADAF38DC97144CA32BB6DF129EAB07313059C725051883475E30B25C8CEDC63BB517319AC"  
02783 "DC1E0ED86D5BC5790F11CD2F4B2CD882345706A4CA5863C0395C3159AC3D7FF8F7B5F7B599"  
02784 "07CCFAA53EDE296807B04FFDDB0B221B3DCD1A6C64A591729A039C230CA52C94ED2C311879"  
02785 "4876F7478E7968F962004F01E8FD0546207A64C09A2292376F77D7789CE2CF04995C7EFDEC"  
02786 "91B5F13EFF915E9158F6202D6A7B0B97BB1FB121FF99AB12F56732E9049A2C33597A308A47"  
02787 "FBEEE3F350E357BA6B7C98D58114D89C0E9599C6B17BF409B3B63FC67DE29FDEE152711B6"  
02788 "3C02F919FEE1A31636E36ABB4F9BA607020529814E06AB2D8E4FDE4D6C1A0D709ACDD7E44"  
02789 "2B1EB65643FE7D9E8D89906915AA868E9DB2E070CE6511EDA0506A19A731C0DFE149788BCC"  
02790 "155882D9D52E921430566F9DB82BD062023BA5FF61FD26F8209421E8EC2C22FAC9F58059D2"  
02791 "23B4659A23BE538BEE894015F8E137849280668B971AD628CC57ECB49701EA1F0CE2D07B2D"  
02792 "7D8406FB1D2E010A51C02F6F8F04F38703326711B85ABF70D46D947A42855B7F13B79D39D2"  
02793 "22A3576D336C9B8A69A12DD38184EBF9DCF787C0D7611AB4176294C239B60351B62309E5D"  
02794 "BC92BE0C0EE1422D9074905131757F70343CA00ACF2CF9DE239D8C794AC868565FF9900A1B"  
02795 "9D187580884D509642AB4F6331A89FCBDB367067712B7B4A7A32E266D7FE571E4474FE5DE"  
02796 "28CC0D480D4AD11DOC6B599A56925D68C1B07CC57612290221AAA0F91C8F7D21EAB35431"  
02797 "0250A224BC0455C760F62B2981ED68F7E10619AEF954B79D0D0DCF039DF951BA2245DEC02E"  
02798 "71968009ED5364774644E995DCB33A41D53B3C9C1BE952449CA8E671E15EA888789CCDFCDC"  
02799 "C91E0ED0933CF1566D358909886E27972C262F5E1A481B8F483BFB311D1595F51FE1AC8CB43E"  
02800 "BC92C0D17868986B6F2BA8127BF95E159776570E273393FEFE203FB079780E675BD62F475C"  
02801 "C0D1DF76CCA50B5145565C3615498DEF9B7BD37887BC6988A9853B4E639C1BFB61E32954D1"  
02802 "166156F944C722444C29AF577C12F567E568BB1BB44A9D0783F126D6A079345C3842AD6240"  
02803 "F1740585102F558C5DB6C22A057EC2D902CBFED926BA5F9C298E6F9118B9D2B44A47E4DB2"  
02804 "E4F1CC30E756B6A1B67FDA687E672E7136AED3E02F8CC346F520E5929526B6E9126450E708D"  
02805 "30168D1A6B41D19949AA4BA5CB7A691DB90A7B5040B01518FDD6035454280BEF79E019FB"  
02806 "96D899DF97BA47D0A7EBB39A11A84036659598294B0B9A905CE2E0CBA0E3AFAC85FEF2CD34"  
02807 "46057ED5DF2FC389D411CC3128A8DD23188F830D90", // PRN 28  
02808 "3A9010CF2726ED13F833686B6C779687CEEAFA4A99B40BE702CF35774B6BBE4EDF528C39A8"  
02809 "FCB10A4B9D25FE6666535391A45D5DA45D8073F705184427C3C5C4225F66F041151A22423"  
02810 "F1F5516C744FDA91305B2B70BC11569416ED456BCBE6C1A47A5805B464EF6A645682D88D75"  
02811 "E7FCB458BA2F9C44E14589D2805A904AF02FD3693DE69D0CAC00E058B26277018C3339EDA8"  
02812 "76FC820AD6C99149BF76A8C47ABF795EEF76D55862F4A88B26A73385589048B2E22C542461"  
02813 "AA9CBB41DA58C7A5A32047B9F85426119402E531B023E3C8C1EFC7AF70D3D553B50408AB56"  
02814 "8E4D2EF96E5FAD8BD3EDF830FF2B12829E97C5C57927E69A438737B1920553F2B921E62A3E"  
02815 "25331B3BBBF99C9FE1912E0A84B15EECAD16B03D059876B7B02908C7A7DDCCC51E01213B4"  
02816 "829D21C9FE65AE129F3B58B51798C604C9A2D2BDF7376FBB50732D9694E199DC532623309C"  
02817 "A43D837EBCE0EE9F3943BF57F9ECAB4D939B2AE599FF5CF5370C9B56B7B9800A651FA64F85"  
02818 "4962A68D4E1270EFE2A481D399DD3F35C9C43605FB03315259989C4F2EB41B83C62DC2F7EA"  
02819 "15B315126D27A0CED8B626022F9F665B49B8C32A10BCEB0AC7FA80335A255076BEEA7E5B"  
02820 "EA0472531B22E07A6579F9726600E661B389A221D2A6D2AE9E0059D24874BA5AEF82B1B537"  
02821 "34F0835FAC5732FF0AED0F092E06EA84B3B5191BE3D226A41A6BC39AE3064A789A8DD9CED6"  
02822 "8FCB902249622576D4086D63B924C80C309F75F1FE9A62071FD2629E927A3ECCF3B575961"  
02823 "221C5C1E76D2D6F1D330028F262E671905340C7621150074C49D9D3D64563E4E6C8B3221EB"

02824 "DFAD5A369E25DCB2C95A1E19D05B4ECB8E605BBF49623F6C62383A4ABF25FF823CB5F924AC"  
02825 "F6D6173BEB850C46EOCCF6AFA873096046B7CE53271A33D18BBEBD6452D1874E312B9DD203"  
02826 "5E511FA43BCBDF3EA9EA8C602EFF6051AC0062232BF8A2C1AAAE76F0C12ACF4A290AC3F6A"  
02827 "0DFA55BCECCDDFC1702D38F74BCC3FDB3160C3F52518EF2B1E14CE9CF22AE5E4CF2DFD5310"  
02828 "57EC85896708CBBB9D810B7D258AE975D678F19139A760F5609F57E2558E1AC0B6AACA4636"  
02829 "017083D25190CECF73D29E2F7B44393D274074A68570B720EE6CEA533AD788F9A7F127BDA0"  
02830 "9D43F96AB9863B5365C9361A9EADA698F5A616D5EA0EB9F83B0647698DAFFE659979F5A5F0"  
02831 "86FC8B59D53C737E2D38EBB4F74C5CC6429B526FEFC9D1426969AB14DBD3F54BEBF25A5A88"  
02832 "46874562CB282B2BECBCECEDAF015F41037486297B30C905572211F80727CEF292CB2DB1181"  
02833 "57E666B190809A10B177BC63158505BAB5C915D37626EF7CB011041B08B03FA92A40EC3E39"  
02834 "BF9B4D7279A5D1E4CF38377BE6C00EE12CFB4549C036E75240D3DF9FF92B50954F461BC25B"  
02835 "D155AF84316A1752B9F1834AC4DA1B1B5A4C84992141CCFB0CC6AC5672753F4877403FC3F4"  
02836 "A52848786C976B18CE968509A4014F3C69A77F9CB4915A1E389C1EA74A05DC08B6A6BB1BE7"  
02837 "55365F1E96E06FCFDBC7210099601912F55F639C9C96F28460CFC392C855CB2D3986989FF8"  
02838 "5A0DC7889A2F6AEBD3E1AA196F97A3C3A0FBB33ACD3BBEDFEE06606DDAC5DA26290149E60"  
02839 "61E8BEEE7F2400588F3FF853EC85B5F5742DE3547F362B0A094D6F935DB0A2B5D1480DB2CE"  
02840 "FB9D665887F6030358AB9E5B26DE8955B188C39D6B146063FA59D1BA260796A2DE8CDE856"  
02841 "0915F07087E8489340103CCF7E1B75AD5BFD209A99C429A52E8D4FE96EBE9A32A4CD24A79"  
02842 "40B446003E1F8296EAC6653659CCC3C85E01F7A154", // PRN 29  
02843 "59CD72F81F8FA04AACD73E9C815A312E8C02F1C1769FDA17C16350D9EBD4F8D10F248035EA"  
02844 "FFB4FD793327951E3BE9E1807288A9D63A6504F3A31E7DF322AA1B2A9ED3C23717C0C5B1"  
02845 "34D2758666CD04C2FA8C1EF5C1E58870C5EE1378619851845BA80DADBC37A764BA17E6DF28"  
02846 "5BB3DDFE31A56D4BAAB0FA622B8D3D24B7F91034D2F773FE578C874FAF5204F3DDB90B8533"  
02847 "906AE05834BE67263447FB4485BA069EAAACB68806028CFDA8FC6DAC6383837F9A059EBD15"  
02848 "29A92F98F44D55AC06083D6B751FF8F9FC5A6444C1A7D346025AD4B67EA12439242FC02D22"  
02849 "8CE537AC9044F6D193E0592684FAA4D259DFB00F5F98DCA9FAE4C0D5F0427C6936E501148C"  
02850 "56EAD1817A81DA30989D0D9C85BF8D9972FFE67AE3DEB2A8959D63B3C3E7737A1851EDC8B3"  
02851 "13D75AA7838C6F7750280FFA6321CC9228ACC596431D72AC0302582CA56C6172599F30D0CA"  
02852 "04E71F3FD7A03D467E37A8320CC0FE3AC72099CEF7C4265F7F18E2980168DACF9C058E39EC"  
02853 "9FFEL1EEBDDCA5A7D63BB18BF054465179401346D4CF8927580B9E0370B2D27E4C6BD4687C"  
02854 "F0D434B804BEC8E424A42B2CECC668DF9809F580BBB3F3D59BFD96DC8C2E57EF5D2989D6DA"  
02855 "54C04B15F85EFF639253DA3301FB6E6F92264FE48968C2C2CA6D40C54C327E0673BF769CF1"  
02856 "16D9D70CE374BC447AA1B121827E5BA510E387974B19B050D9909185A514BCD8CD374EFE20"  
02857 "89290D3DF46D088D74F556A8E5F36B08AB17E54143B6FC374BEDD6FF88AE74E71C72297A31"  
02858 "E13D1F668DDB455DF00DC91DBBDB2E96ACDB811ADF7E4C1C460D8FF1FB03BCF95EC489446A"  
02859 "8DFC5D220E58CC298FBB7A9B8EB33B61F6461D77CEBF9C36BBA32B34998B7B9B7950E7E98"  
02860 "857371003F40279FC26237F16B808912651E55A04F9418989047331C454D2AB04FAA192938"  
02861 "209261029FB3C7070FA056D2AD52BF9C7A0535FE2868888BA55CB9389E358CEEC68D7A0E550"  
02862 "5FF0DE4263CE47406EE83E8C797BF58EF49F2F8BB4EB3656A2636A5EC604A4990001190A"  
02863 "8A7965A36BBB41019EE19C8EB45C10173B513F617B9688D7102ED82E60DDCD080233A7B3B4"  
02864 "3B641A3282C4AF6D3F01BE30DD78AA184B9D565F7B1F927EDE214E363A18547263412EBE35"  
02865 "E939942C4B5E9F8B1E439B325902EDD4D2AC06340296D0DD5F0908F5C8AD88049D88282D0F"  
02866 "2DEC28D133E1D4736869C352A7CD87D8A56687CC25D5517A62F0CA5A023E8709F181A0D96F"  
02867 "1154B3D7517EC5B076B7C59DE90A223B96AF93F94B11EA0EB08A280379BE028DC494C296BC"  
02868 "46659DF9C26A38B8BB47E063AA4EBBC8973CCA2866865CF45076E6E42B8E3BF9F4A39F0"  
02869 "57DCE71D4D5F3B621D8791CA533398ADB31486D22F608E9A8745B9548EADDA654F1D4BC04"  
02870 "2DFF9B3C21B15AEB37050864E4275D5C36F301D533552D119039242BCA25DFABB524C9718"  
02871 "21B2C0EEAA602A4C02C6487DD113DA4DC5ADD7BBBC325CF8C96C942682945A1BE921F77E6D"  
02872 "76FA1E3345589B53C549BD5056FA01BD688D0A9011497C0DFDEC20210BC7F8B713C46A100A"  
02873 "396A9ABD9A2BB94E5455793BFA28E39E130645973C30FDCDE52F6C272576291F0130474DC2"  
02874 "E3C30AD54818E9F7531A538028DD40FE01D993DFCFF82679DC3C66850528A173FBB534843A"  
02875 "B540E21C7731C65044D697B42694A1654B3572FE321F61E98EA22171580F5DB90C997D0C9C"  
02876 "7CF44D174E8D24F500832C158278CF62BBFC6DFC46AEE5DD3F8906A5770C3BB6CC3B731243"  
02877 "974A76C0BCACD9EFE37E52CD9751894A68AE3596E0", // PRN 30  
02878 "C0211AF7A94B37632C667AE5097F8B5992917210A515C3E48276A363A6F5696AFEF5B39196"  
02879 "89CC8F21E077310DFDC5A58D6BD428BFFC678299FD053E7C016EF3A154A41DE5FD3C3BC50"  
02880 "70ED6EC8F16170C880F82EB0B7EDED047436E8FCD4E2290F450963133226D17D7BB99C7109"  
02881 "7978CEFE496032EC8C6E73B99B05A33C4F5373BB73CA93F7128A9818197E6FD2BD88451A4D"  
02882 "8905214157C4259228DEB5F8B216FFB206E4EE1025D3E16987FB6064147D3E99CEED73112F"  
02883 "6D513C8B6F18E5D2D0F6794B307E8AF97354573C758AA95AB0A926386E7D288D74F9C6B0E"  
02884 "2C71FAEBB36422C2120511F5F3D2B308674AE355B1FA1E72DA55098AAE3C70CB1B85C902C1"  
02885 "03056E00EBB70D37F08FA4D90E860C74E841F4E99957B3097ABE602E65F07B8787DA172E8E"  
02886 "9ABF9ACDDCCFF6FE92F902082BDD49547EFD95B54C3B8F6C6D243D07F16819E03189F0A3F3"  
02887 "E0D05B3AF3EC92CA08846FC909E809CE9C30F1F3AC6FECED9043245A00590CBBD83EFC41C9"  
02888 "EE3CBD0E328426F821ECA32E0E84A8B64F200DFAA5F01B262F4E816A90AA33A2984AB8E269"  
02889 "B8CCF61E65094FBB7D10E11F9E1022CD97F17011DA53455B24E4ECD0793486C1E6D4E3BEA9"  
02890 "4E6DC9C710CFFF30CCD48E43076D763957260D1ED4F7C41EFA0B7F78BFAB7A9274E932013E"  
02891 "605426C36CA1FFA003A4B24DAED61D15327BCA1067ACE961E718DD5E57667F5F33411D17BE"  
02892 "093A9D7C6675097FC551014545727BA3F19C5D5D23A0922B11C4BADBF285A17EE96C916238"  
02893 "C1E1DE8A0D28FB79E72EAF5AC37710B4E6550CECA2F22697DA18F858507CCB95BB7F25A00"  
02894 "56FFA55EE9564782C6042686195BF56901D4ABB3398C557CD47D30EEB344B532E775668876"  
02895 "93CFAE50CE8C0BBAA204465FF33456ECFBB1B4DE4F8D6E2F3E8A84E90410EE3A79ABB47D40"  
02896 "F4691CEF37E0DBDE1BE386098181EE36051EFFE8B0F5898738E400D44ADA4B3348315E0BE8"  
02897 "8C4C43E8CF95CE8D56EEAEADE754A09F170D3ACCA44DB101037E0001DCFBE2C52DC32D040BC2"  
02898 "7C1A55C9EF5D65FE418F3B761284C461606ACC84D36B9A7A3C9353723BEEEC80CF75D84DA7"  
02899 "B2533F844E81EB8835289487CC7EA4EB277A4BA4BD50B59953E3DF3EBB2BB69D7E148792DF"  
02900 "55D4296BDAB9B40B00FB14A948F2797B7488EBBEB4B11880F7D3062D3C0561D7F6FE715C3"  
02901 "BA6BC703BAC33C29DD87EBB84C178BA09961F2C10D8CCA42BFC3977EA66D039FC24CBD2A7B"  
02902 "3316EF37C0D0F98D3CB8D5E4708B4287FEE6B656F1E300260D5643E9AC323A7F0527A7B981"  
02903 "74ADF3277D35D6EC63865BD4E816351B8492474116E2426A3298606039280AEB595293D18E"  
02904 "EC671E1E139EFC7936CAE3056F9698BFE95568D050F281F17F2CD65B62B98BF36DE469474F"  
02905 "440E27073613A121370C606D5729DDFA12C8C400D0364251D850D90804ABF000ACE57AB195"  
02906 "F6EE24163C8C443CA8EB641AACABE70675ECF22A5B9DAB6DDF0078EC89F2354C27D2964110"  
02907 "41432AF390DC71CC6141B52C54FC778B5BCD31B55EDDEB62EE364983E97BBB65EE98863AFF"  
02908 "D2310B482AFCC28A2FE9C7D8E3ED24D20FFAD1F5B0AADB344C77F89FB88B2FDCD64A77203"  
02909 "BC331FC23BF78E28ACE3016FF570E3AF32ACDD5BD5E7B59115950CE804C231AB1AB802768"  
02910 "42FAF80A576C9A3148F527596252E29F76FD6B4C03EFC9E24A4740EDED98B290FF1389268"



02911 "11607CC553FFAD5ACE0FC24D7FA1D7F3701A8E0D7F614941C9FB9215389CC5EF6D3CAC193E"  
02912 "B15798CB1FF2950E2E0EA96A6B9E672CC343A7FA9C", // PRN 31  
02913 "28EB964EC60708D60D921BCA637AF0F0E19E127E0E563EEB30DB02F8FBF2848CEC8E351077"  
02914 "A8294AD441049000D2A3BD101B691C2F1DF3741B8BE7DC57592C2D09C9F368010B4B3922A8"  
02915 "A622887C37E959C9AC31DBE89C9EC9FFC26EFD9C49362E3400E660928BDD7F67C6E45614A"  
02916 "DFFB96622390ACC03F1186D18BD11333AF0348125A92EBB40370A32F5E383ED858F6720D4"  
02917 "3633A9A3E3559A0ACF5FCC326DF117AA495759FC12398B2E060F8C6DA823F3E58E1158B81A"  
02918 "FEDE50B89F956D4F63619332928B4CDF5A16E0536FE7426518B1C50FBBEB4EF3830D77924B"  
02919 "E1DB002BF2B0A9A8CD6FDF3BE7D7D1D7A3EB4E8CEFF8EB9E2CEA9BCA74349D67413E7D05D"  
02920 "EC014D1E6990062F0C4CF1E4720D45E23719D748D0C9BE3C3CA444E357F181E44EE841E35C"  
02921 "B5207C3F993CF2244A90EF5E3610254EEC4C2F2EAEFE041E1791518F93BE9080634AD5594"  
02922 "B98C6D0161BAD23AEF1403FEE6FF846D344BF0B42A65762804A33F00F12D769B3C58F123E14"  
02923 "B4001CC631049F6E35013DF92AA50F89E71E78879A9B428341A738CA001E4233652802FC16"  
02924 "AE2377C9C0B5650805205D31445EA0AC0F82A109E5F7C0758EBD5C737CF48D3D5D44816504"  
02925 "D972CD2FBFB0AE556B8247F81E404D7DAB5A939AAE51375154A3125345C061CD479F8D7D9E"  
02926 "B21C6ACD971F75D8430FAE557DE7A669C59E35DAC4E7686C53C34AF64A5AB7F6B5ADF48585"  
02927 "417DBA3584D4552C241DEC336E5EBA84C4D295148B29FE824D3934F4D2A8EA2D27D2925598"  
02928 "8396A5C350421E2C2BD89B5D18155B20260DAED20E3DFF52F2E84E6D6CC2C040FA3D0B8726"  
02929 "C63778A601CF95BCFD381FA520F95146292B8D854C5F057440833E78057D13FFFEFF4418E9"  
02930 "7986A064AC8DB74A001BCBF8B8E26CE9C68BACFCF4895C956666F92B5EADD27FF6FB032207"  
02931 "FCD2100FE0DDEAAEA860F89BD6341E59671BB0368B1C68B44B131139A4BB37EAB251A033"  
02932 "9A79D358D73281325A594E2156E981C519419D925F0D6C8661F7C294D91F527454AE82EFF9"  
02933 "8344A6E39B2BB3E36431255E33DAA31B75E961FB306792510924C3E9B89150691BDDFB275B"  
02934 "9128677DFAEFC89E6562FA245DECEE2E788EFC06C4BE744CA97693CC1AD7F0EE4C633D70"  
02935 "6AD278A46A8DBF2DAA0CCB69806912F8C081BF192ED7D17EC22392B6E84D6CD88FE66DB10"  
02936 "B1EFB9CC7F3E4F3719338B8D6BBE47644009743829F288A194EB3F6F21FE7D7FF55F8BFFD1"  
02937 "312AA603C8B0C011CEA377E6E2AD1541C5BF55AB73777BDC20DE77F701E7DF26728A44D881"  
02938 "EC44E905608CEAFEDB670A261F2109DBFF79CE13E98A3519A9A86F3E01B7C3EBE2F01C657E"  
02939 "E017F0740328B97392C6824DFD4E5820458F0DC9E999A4F878D09493D249D0B43268D44135"  
02940 "C966B03AC8A69518BA5AA10423615DD3BD8B6E38D37D4E7ABAD0F8752A64736EB4D665F6C"  
02941 "3A6C6E1595AE0ADB31ADACFD6B4146F4496E739C4DC98835CE4C5E2F1FA899D4B896B02CE9"  
02942 "7822E90CABF580321487D0C9DF07772031F1A438E9D63374373958DD9708C6BC38C3028C2"  
02943 "C7DBC7B2A5A0C8EB878F7A72EA394FBC0046B62D60CFCC4256930BE65B22305C750B2A7483"  
02944 "BE0B3FC0824A011D274FCC289B8EEB9ADC5505426D8618369E0135FCA2FFF5456E48010995"  
02945 "2CFBBD422758E14643AB48F11AB4F4EB475CEA238C6493166BF384B1DED53072D897E368C8"  
02946 "BF8239DE532907166A4727CBBAB88691400E9FB9CD53442BF91B5D421323A24571DBAB65B7EA"  
02947 "7015775B6C8DE25247D0E7BF3DB7F2506B4384B328", // PRN 32  
02948 "D7554B01374AB97EE1246C2DA82B0AD5BBC76968B6E058A821418E0707D596E2A7E3AD9AFE"  
02949 "6231D8127F269C1D347A732BEC58260D2E8F1399AE5BB928716865B902611DA70761A4C3D"  
02950 "B3EB0D332E58E081C08D4AF96B1A85DD68CE15A8EAA5B95F73CBA9975924F8F9A3CB57249A"  
02951 "6A904A33F6471ED4F389168702EDA2054B4A523B6CE189B26C50900913CBC7693C4E447CAC"  
02952 "4CA7B783106698CB878F5E793284F27E182CE12BC792935FE38CEAA7F659967111AC2607C8"  
02953 "316FD3CFF111DC5C2D58502555AA589010F9DE4F735D456CB9F0896FEB163984710BCA11C5"  
02954 "D78D9ACE6CC9E2EAA70B177A9D33D8C59245FF9773AC181548BC82C751D59A350C65C895AA"  
02955 "79E9B57FBFA2D17F53EFBE0C490B43C369B71DE45B9AD67C88B2A8F1AEA2F5D871970FE9D"  
02956 "4C6D63F81647DEA310F76DE6C050729DA4575087E3AE8F6804128FDA47C1867A6FB5065C41"  
02957 "86C47D1199BEDED2D26BF9825676FFD9A98BF91E3921A07EFC949D7473CBF04E6647390E3"  
02958 "3CF765BBF73714E0AD38C4D1EFBC3BC4144CC9AEBA0B9B4843F910A7F9CFDD347F6DC1017"  
02959 "C010DF0EED1A71958CC5BB8D50BBEC164E186B6572FD09715C29611A3A5162496968C1DFBC"  
02960 "1B6A8A4E617922D152FAF2B08BA40E47AA0E9F30515C9AFC74A5C2D87828BAE5243BB7988D"  
02961 "ED81731126B4C7E96086963929F943930FC72D0F3A4784990E08145485CA0FBCD69BAB2391C"  
02962 "82E75BC2CB2DCAF0E94A3717B6A157070C77A8A3B5DD1E30AAB09F5368F46F66575599C042"  
02963 "96982E6AE8F8C3324ED24CC58449CAB0B7660E0F2BB9D96BE515245DF9654EAFB5060D21291"  
02964 "7D1E744CE52284D96200CD90F0A7B4A03869A93C9371B7F946EAF2EDF35ADCE873407C7BEFOA"  
02965 "9FDA9AE4650C9165037F353D9BE92322B4383C30404706D102F15D8D39B21D22E0B23FFB0A"  
02966 "A0F58F477DEBC4F130BF501C765295A54913A7A0190FC907BF2420B329B7F30FA65A7429E"  
02967 "7D1E744CE52284D96200CD90F0A7B4A03869A93C9371B7F946EAF2EDF35ADCE873407C7BEFOA"  
02968 "E68FDC826682BC406D4A7575EE913B60769C3BE4D81A073C05FE2BD6D7A1C3B52FD2F22875"  
02969 "BE856CE0F6D7A3AEA1897FCDF951DF4A86184E717B4077A25D37935A82E07337ABBA58B0F"  
02970 "5F744CE52284D96200CD90F0A7B4A03869A93C9371B7F946EAF2EDF35ADCE873407C7BEFOA"  
02971 "750CEE48BE3A33CC47979648FBC5EA7E79DBC725C3901D4D05BC5B804007BD46E48B424665"  
02972 "763F028E3777FC1A00F588D9A26F68DC7ADEEAF8214B747B409395016A05D8C72FC3E838DB"  
02973 "F7C5A222D03699897E5606433CA760C6D67C8BECAE0FA8C312C9875AE26D61FCBD3A320829"  
02974 "B615FED0587E4225E4B62264C9B37D2D2B45FDDCEFC060257960DF6A1AA2AC0544FC333914"  
02975 "D192A1629061730213B1455E6FD7A712A08122C8BAD77EAB9CB9687AF97A3AFC02B8F41393"  
02976 "ODFEC10DB37C385B5643E91F3345C73EF353453F34A5EADCE8EB41CB098E9243E57258AE608"  
02977 "C9FE7BBEAC524B21A0734D8EE51EB30DD6974B87FDF3B0B1E1DB01FEF13061E4B072DAF927"  
02978 "C99BDBB7691069E2493E86239A69BA33AF82998CC1D7B368A4343BC63EACEED280415FFF40"  
02979 "B0B61221D7D1C098C6391E5C888228045494F86644F34AFB1B191BFCFD1A29C678956DB16D"  
02980 "0BBEE458AC6DDBF00E738198C093170FB37F3F141619E65502AC57E9042CABC72372E5FAB9"  
02981 "DFF5E42A0EB023E44B4E536706FD63BFEBCE1C0DD59E465A61DABF566EE7216973ABFF5D3"  
02982 "4A5A62518E12E8352C61BED9348E5826290EC0F84C", // PRN 33  
02983 "4251265EB325D261D721A84F62224701C9F564298EFE0C2516330F7AA43CC3D7090581B74C"  
02984 "69218D74C8C6563FE276F24F39BB7998CBFB9C49CE649106DFA51E45498DF0F347DE8A5623"  
02985 "41DB34E94F286F583832E49AB38C09DBCC3B2C307188A980574461CB3C41EDFB933938F89C"  
02986 "88F61886B60097F727E511B49CAED645190E98C3AB8AB5CDDC021F76567AC1EAE9F4F8D782"  
02987 "B91BDBA3037BF8FF43879FB6598D84046CA06B3C8EFE275BC61E44983F979AD6BFD27DF"  
02988 "453E2551F09FC1916F4E9884012BD4B946E2A65C86C609CF8EDDEDCECBE9B1DD6715C1792E"  
02989 "261A3A013D64F00118DD1FDAC9900E8BCCCE9246446D5248D5E3423D859E1AA9BC38F869"  
02990 "7F844EA08632983F268BF7D0BE639ADFCE6625E0029FD510A41AEAE418C3AB85F5D2256D8E"  
02991 "A4E9AE47316BE6DCF792537F28514B99BD76D098C8B4CA16DE1A9ED0F2693DFBA9B460BD84"  
02992 "2452BF563B03874DCCCD3E1B876AA3C352978473A2E02C6398438A35639457DBA7316577B0"  
02993 "863762FC177010E0FA0F7537420C0443D8D7F51AE4F40A67E249D678E77F17503A0A6DBE"  
02994 "3C60FAD140DA592082C334E3B1D021E34104A9A724FB6EE3B5A22CE5D3BB9C5E896956B4C6"  
02995 "F7B1F51A74A56A7232E027F706EBB999708BA8ECBF42F44E41EF63A7627C60E5E75C678B5B"  
02996 "F7846CBCE8C9A1B6EF78D7912AFF0EA56B7E25C31784DF9D63398FEF53F6193554C4DB87A"  
02997 "96C170A89961D2654B9A663B39C76401A439CAEC482BF44F20DD12140D47D0580DDA48EA90"

02998 "5D409BAF5BCADBB8394333B97C0FD9501AA4B4BCEB3F041520422530A09A9B24EC04206CE9"  
02999 "07A33D0A18CD5C1640CFEF970623B1AB2B7EA5342B6F38C717416C17545D27C21A8F37AC2B"  
03000 "98991395D3170FB2D03C6B152915DDF9CAA12B5EB19E45FAC4CFD4ABA726A723237A64887"  
03001 "73C05D0D38D38D920C6E8F2D3476B06B40094463CDAF380026BC36B8EF3DED6A01CA89E015"  
03002 "A0B75BDFB97D859C620D329C9B26B1DBE3A3C92D8510B060ADF4151AFEEED125ABC0D6FD3FA"  
03003 "9F7F3C67B6BAF14A672977F2052A84F0437027956AAED010F63471C5848781BEF8F2BAB9F8"  
03004 "CB0DB00253E2C6FD30D2A0794FCEC489376261CDD407B4ACECA129D87FFA0C76E28B948303"  
03005 "4A40F84EA21E06C37BBEF2987AC16E77E021D29CBBDF077A912B89CEF82FF0EA59F7FDA55E"  
03006 "E99F4602B5437CE71B4183BF28BE79327483898C39D09DCAC59C3CE3233BEF4B0720B3094"  
03007 "4B9998A01CB3C135D53B4A76FCDAD1C321D39D4466F0BF95710FC975ABCA1181070A4A5741"  
03008 "624CE9AB0FDF27A5BB1805BF295861D0208BEBE7DF92EEFA7FB123B6AD42541C1A1057DA25"  
03009 "60469A680E3664B1455D6851E6C8DC8C4940281AFF025B2BAD2CBB53D1670641EBDE2AE91E"  
03010 "71F7F69AFD0C1A0225351CB1290EE40F81560A52463FB04E1364F9F3E5C454C311240EDA5C"  
03011 "6AD78A09A473DDC1463E19A8E12419DBE4966A1F12872E566F178E256F257FD1890DAE6F84D"  
03012 "E013C2490B574C1DDED2E705C0134E889EECB5575587EDFC240DE938C6D2D9CE543D5BA3A4"  
03013 "A3EADE1A6A5F0CFF996311F72B3C6761414E3432C3D1A48F5D3011D7BC7F936E8E086581A"  
03014 "C256DC6A376F9C4ABAF87FFAAD67BAC64401F4D976129EFD57102A84E63DDFEF60CF13EAF8"  
03015 "973881FEDEBFFD8B356B2A03EE9FFBDA99F77EF7251C11EB1D2514D5E657697EFC47C42"  
03016 "CF10FFA7F09765E2EC39E0A99E4FEEBD410BF4AFE51855341279F7E47DFF0066602618ED5B"  
03017 "F70B2F6D002805F41E991C68D64ADEAB7374F25B60", // PRN 34  
03018 "E0DAFB80D888AA2580DB37930D1942D6D088FDC706871A90D887B5F2CA6B21911BBB8BFEE"  
03019 "1376F48428238607AF7CF82C3B3C9223746BA9A5F01523B20889ED8F139EA3D505D55D766B"  
03020 "D056416193CACA6D5D6175C16C38BF44D7F89D4A5E3406C7944AD05D86C3101E1EC78E0C8AD"  
03021 "80FA0DB202AB1768AD446AC7FCE60B30D741114B45857FA5223E64AF32476319F16DE15C04"  
03022 "F34F4730AB31F4F6BB13252D706F0B3BD5860BEB864618353CE62469E25CF4AED50DBDE96"  
03023 "33CF18F76AB1EA5552614D4AC3848A0CFFDFD51F17A4E744533DC52979D7012AE53E9B08C"  
03024 "4980AF8876ED2FD8A6F174B3AF651517BDDA1965264F1CCF0EEC8E3A9969D1359C5C9FD20C"  
03025 "AD8698F3E28AC77001F563BF456B5A46FC747E5D7EFED93314F7AA87CFA60405F968A4F97"  
03026 "74E30E2266938B19E9265FB71BE0D40FE6BF14FE02E36FE10FF110D63E02145C78B73217E"  
03027 "ED9DD87F70077E4229572E6547D71828C773E46462E30B497F9F5F99D723D657E17A288DC"  
03028 "98724B11AE87B7E56D33386A7FDEE0698353CD46409FE8D2236958B166B1C697BB8A42946D"  
03029 "66E390A045775A6FE7DEE4BE260490B69C56476B894383091C827D02E96D09359E8D3E151F"  
03030 "50EB87FABAEE8ACC6E325B36674D9C7CE249B879B3FDC6AF95E92FAE53047BC211757FD125"  
03031 "64EDDA4DD6AE8C94DD3305C21970E90BE05ABBAF912ACC52C77714F87C2B059A9E21005D8"  
03032 "903A604D82EC22192C46CEB63C588A219CBF755B5D44CC0E918CA6705156CB497FAABFF637"  
03033 "45C826FE16F65582D4820141DEBE3B949C29EEC6D99C2A4EF34DC878DD82D0FE6D5ACA302A"  
03034 "DFB897DAE78EB0BD28A938D6FD5A3535CADEDD7C521E45AF4BA92FD40DD009E35C21254A6"  
03035 "DB959698303D56B59D8F319F18E688F9A20DAC8058E0496E0E46BD01880F3C78EEFE98A511"  
03036 "768919116A288CD3CD2809BF5B7419EB64D548A6D791F1A9A9A3E9B08E7184AEFE60A3D7D"  
03037 "7179CE47562E9133068789BBE0297FDA2B2D6F4248B288CD899BF3231EB360ED45F769C016"  
03038 "7284319D2C2B11375C2B18E5B67594CF499A6AF2687E0C1AAF09F4161F3AA8369BDBA68A0"  
03039 "22459A2DA9DEEA13E509AF3A68D011998D9C30BEBEBD04BC9983F36FA549BFF3E3D323722B"  
03040 "2136AFCEA02CC8E593B370C7B48635926B9575A77AD7C0DA18E94FF1C8FABFB57B25981288"  
03041 "E1771465360E2D619272CF3F56268B7EE130FD07DE29215FBD5D0A72B73081869F60CB149A"  
03042 "1E4871150778A2229F3C886200E0DC2C76AE049231FD3171E49FF44BDADE185883198D2162"  
03043 "9CE0AB252C7451398E0D411CB01A66BC5B141CB150BA9CFA9F9F46339625E1D5127998302A"  
03044 "FD19D5B2D85F3F0B2B354C7CFAACA3D4E7C7B7DF44E22EC2C226178DDD480DF7787EB28F02"  
03045 "85B1F973D16F58D0FAE8D9FC79E894CEC183DC24B1E3C81FAF6A546E464AF8C0307263680C"  
03046 "4D7694A0F6A0366B5D74E73227F3D503E6B989C5BBF9E6D81EDA7C46D890FB2C5CB439C5B4"  
03047 "9FE16A9BEB14B12B08C47713331E8932D4AF6B85444E3F848C5AFDC095E984449F0D126EF7"  
03048 "11FB843831755B0CFAE8D9FC79E894CEC183DC24B1E3C81FAF6A546E464AF8C0307263680C"  
03049 "EAD9ED6E80217F98CA82458BEE59D2A4FA76BD7D4425E0C77164A50A2933972108661820C7"  
03050 "5EB51C8A0ABDDDEB30F31F2E575AB9DB723DD084656691096B72193C52053EBB8B72C6B30C"  
03051 "49185166FC39F3D9F3497606D15E8D696CDC256DCBDBD01DD2C33C0249D6069788DECO980"  
03052 "2630597C13EB1AE152E8E84F6A4BF60BB09E049EEC", // PRN 35  
03053 "EF79F2E35DA357E9C076B00C6ACE7C50F865830698571E999E544F5025A99C36BCB025B796"  
03054 "784630FAD3A22B8ED499BC9A80EFA7BDBDB83B15DC6D48AC4E3978D528DCE0A3E11C3E62"  
03055 "B619EBA0C853EDB0EFA1DCE9EFB2E024FFC0B59A8AEC329FAAF418DF9BF70A250F2F6409A9"  
03056 "FC0153161F6E8713005E53154DBE15242AD801C216A1E5CE207757BFFBED75EE4C96CA0C3"  
03057 "CF44896654EB434C68E9A4CCF3C5907DA216BA664B073D73B0779204EBCAEB55446AC6E2C"  
03058 "40B8BDC666D4C9D3B7C64F867BCD8962274630191EADAE7248398B9F02744AC5D4DEBDA"  
03059 "B2D6D9F8D6C6392683FC1A795F70134E790718D56DB6D424725DF8AE7E11F02D684DA78AC"  
03060 "4B9C311806D082C1FADE4B31B846DE5F16618C47E03E8D78DDCCC6ED869F77CD39926945"  
03061 "466369CC2371D3B51F58DFC6471D742C9C82B1B7A2B1B1BFAD15FEBE592AC0F41B19C126D9"  
03062 "23BF5944408506ACED28FBED161902EC5FF19CED25D18F47A76860777A3D86B3E0BA7446C"  
03063 "3F1EBDA88F8A8E182CC0938384842D42F41E0042C53C645DC87C88429E06BA9906A7A3C1F6"  
03064 "A76671D69703496324406C7FDB20B9C3CE9BC273A743DCECC0016BDF85D82112DBD4C49E2A"  
03065 "6A5DCA22738750EB8DD6AF15B27A0601970AD12660A51C560E5CD44B55960AA266A88B82E7"  
03066 "61B53A8B42E7332EF2FCCE093BCE06FD18D916FA10F4E9891B103DDE5D808963E6F71AE747"  
03067 "9A5D0BFCDB8A22086308CFB7813EF6B7B5304225C2E9690E47CBFCD534570933541E8DC99"  
03068 "227986C0CDB40924D64710A7FE28D248E667E18197B44D83FF1A0691E9049611AA1E0A1DB"  
03069 "06595EC32A9B91C46F50096553C0B5716D2EA9C9E3D373C787783B50F28948EF96D21FB323"  
03070 "CB69FBA26E8C869C745FF08AC0C71D2B656E42928D09AB4274C307FBB04AE41FE847721D95"  
03071 "FDA55510893DF24F56522EE537F609FD54E4DAB4E98E9F3F8E8243C5D056B4CF6F23304085"  
03072 "14C6CA22DBC79BE374E2B34B3E9A9B689B9375035F5E32013587C402C14997741654DF70FF"  
03073 "5F74C0DC26D82AE7F036731AAE15D596E42EE2E738EB540A48EF41FE2172F1565AD79202"  
03074 "A016EF31255DBA5DE29C8C77B4704EF177C97DD507360FD01B44D612DCB18D7B5650032B1E"  
03075 "768851C84EF794F833F9D5D198603224016D3FEA90F3C65DA465418750A69C43050A738A7E"  
03076 "C4D415CC0B4048475936F8F675D49C67B875BD8BE2B2D65E05C49EDFE72163CF3F3BCAD87D"  
03077 "E8956101964A1C1345F3978E144303FD1AEFA4675F521BBA59F0EF43DE19418993F444C1F7"  
03078 "5BEA24B295CDB73A51510B8B4B59DD20829466409444D4FF428B3B94164666A642FC2A187F"  
03079 "DD7ADF7E5813BF64DE664B131F58B9CD825A19AC360F4F83E43FFBE0923494D978B1B2ECE"  
03080 "EF07F2387E03B1D933008C0131A9ABB08DE1EADEF78291141D285EADF1228554733CD927E"  
03081 "9740D39EC9F539FAB960EA829552CFC36CD8BFD9092C821CE4FB4F29115BF533A56D7663CE"  
03082 "9FAP2B9DA374B6E1CE5528A5B11ACB9FBFE818BF4C41787F5E4FF4D517D46D2234B77F20ED3"  
03083 "928BB38A32BBA350A66540E19B586C47EBB9EBF4E4A671CB41DA55BFBE5B39C025BF8F1A7"  
03084 "86D60C9DA6EE5299B17C0135689AC9D880B594EC51BB52398E78AC7DC2439D0C4BE7BBFB7F"

03085 "044661C9CD8D6D7EB16AFB943589C6210882557C65034BD7D044C6E957722C93537473E2EF"  
03086 "1AF5A6D05D7404E274D2DA8B83F6D2AFF9EAA55D122F69F469927EB682977715FB1617B363"  
03087 "68C22FD6A2A7B10412E9F202C6FC9BA1D7789311D0", // PRN 36  
03088 "18085D480B93B422C6BB24B09E05556F8298FCA307341877435D2D76B9DA1A1932F20D2903"  
03089 "A63A0856D9C431FA3C4C591D606E0043B0C6BF277971588390BCAFFC01E14C6256290E1A01"  
03090 "EB7BDC8611EB4E40BF72B04C21526D4139E3E75A13DC25C0691B67F24B195F57923684BAA"  
03091 "F94C7E3CFF4830FB5E828B8F7A8692C1A0CFBB9FD3F021D3FCF12C7497021178A2AB8D3256"  
03092 "E8606374D6D0E7FE11BB1F313BE2BDD4D977780C1A2053F6CB0F60F89671AE59C178FE3390"  
03093 "0FD11B4DEC25C698CE0FD4E451D6A3CEB4E41BDF08F05F556B4394858983D515266F25703C"  
03094 "17B48B1AA13056ADB2AF9A865A0FD30BC4BAC0D7B1D185AA2903D1956DDA3D5541873EFE48"  
03095 "7E08E83613428B64D2A499BCF3783466079714FDC3D0B04ECDCE5C89F77F5E6E6F1141D47"  
03096 "EE7A3C32382A89F512F9C9BD96AB24EBD4F77B911D8B8206CEE29ADC1E055B7BDFE94AA"  
03097 "FA750B857EC6FFCA693787C0E2817554717CE5DD5568DB1AE585B9BEED0C6E228FF47C43FD"  
03098 "7565D3D566ABEBD73F97DCF73D8EBBA3FA092BF6EF21730EB909327313FE1493BFB4439756"  
03099 "7DFA30BDD7D090245057098AF775CE977528E91D9C3976A4D478A5695482E4EBC1FCB27FC"  
03100 "F6E5F9B08BA8C7DF12E6F77C3B0362B1132F9209AC1D69579D5E9815535FF72820BE141402"  
03101 "0694A3E2DE53BBE7DD582C70455BCD18AA0A4E0A8BBD718217C616129F236487194B266F9"  
03102 "5E644D1F5721E90B8A16E7504947B90B655744758D97886E4860FBD728FDE80B8F32C9AD09"  
03103 "0D2DD58FEE6AB2549AF9380B8699F3EBB0F96D295C22B940044819D7FBF2364610893457DD"  
03104 "ECED6113D1B14BDDDA8077BB32E70D19733CB795BC8964FD9788ED317B5E433CDFAC3CE3E0"  
03105 "EA28A63D75EC764276BF39FCD6A1A172C777F2E9A44817D1B6B3C09AEB4436A3B115605A850"  
03106 "ACA241AE496D821EF7FB9E8D878B746C50DEB281604830360C8ABA336D1E8F444FD46A74F6"  
03107 "B1A0F1F61EEC42C1DA2B6D88CD1AE94B3208866BD1757B1ACCC09BC155DE4690A97D0244B81"  
03108 "9C0A66785EC764276BF39FCD6A1A172C777F2E9A44817D1B6B3C09AEB4436A3B115605A850"  
03109 "AA9D11BE2E660F2B214C45B00134604EE395A77BAB321298ADE920BE3050CD2D7BAC4931C3"  
03110 "E3687BC4FD7E2EFB536E26404E9950C928A3CFA8A11BE42D15EA8DB2037154F79C67868044"  
03111 "D043ADF75F21E90B8A16E7504947B90B655744758D97886E4860FBD728FDE80B8F32C9AD09"  
03112 "F4F827F34E8FCE92B576A8809738D6227C9A31A43CEB3EDE56BF7306DC6F208D6229CC3E5C"  
03113 "4B94AC5486ABA7F0079EDDCA82C3C77F41C58A88C4ECF4DBEA754DFB3424BC3106A9FF28"  
03114 "E08C6A7D086FA06C44ED23A82E23A0D3FD4CF62DE1AAF3A95747CB5CBCEEDA5B1929C13DF"  
03115 "6CFDBA6A94A62AD3035C8CBFA10AD9D37389B2EB90822E10B422BC165A3BA86F37C0A5B96"  
03116 "D0EABCA3A31FBF2806C65B37665A434A6D59B3BC9C339F4437F4598D0D6C62812A2083C32"  
03117 "4398938D78390F9BAF86E42F297405069D350F86B089ADE021C1599FAA2460EF5904A6E257"  
03118 "08032540A7747BB254679454A7B38C7491BDCC835033A76514869ADCFE268FE49CF38A9B84"  
03119 "4A97214E1E1ACE873FE051FAF282664552DC98618E35DBA4AD9B1FBE50700726091534091C"  
03120 "631B54C944C28D97DA0961F8E4106408017CF1EC86456BCC1F4EE7118D0691F0F327A0436D"  
03121 "3145E5FBF8FAC33E5179094FDB03D9DA0D9A2EEDDC221A21189B68EE1A94B13F062F21472D"  
03122 "B5466B266562E186FE01D028CD4AB5B6AC9F048D8", // PRN 37  
03123 "D50CD8EB871C5F371648DC07E20BE84263CD676282D56EA5374E21B52752DBE416DB787BCE"  
03124 "232226568621ED792A9135A59849451A4A4D8A1DCF3C804AF966881FE5156FB761B3736F42"  
03125 "82240FC0BBE72F51F0122D96B3755109E7CDFC70F9DD04C58B4A51C00B4FC0EF252B05ED76"  
03126 "440C97DABA04079C121351CD43BCDD4AA2D2495B4B9B08A9AF815DEE1405C3CE9C9C7D5BA"  
03127 "9919621E8C4E920E08061D228038F73F8E5DD66BAB806D2DD953C6FA5360487957912249DE"  
03128 "009655658BE8CACF00E6E0621739648A7A75EA06E93946987E62EDC270B266BBBCEF61D7CC"  
03129 "63013AD797589A7CA62B65D4127E701BC95D45C74DBCFF498D3F87CB1B288D6012EC932B"  
03130 "901A3DFF16FE1E1685EA1F92B1F32DFED2F9076EF9CB1ADD8B04953E5509B24C48977FEFE"  
03131 "6044B8C5B80F9DED6FA7110C3A2EA42EEE8A83F4A28FE78B27E84AAE7B3FABA56C1A4D2414"  
03132 "E0341C69C40A43D65E16583A151826F7C7CC0CF0A5999A1016BE6912F067913AEFFB525EE41"  
03133 "F1B9C82881054E9624C03DECE74976228B119974BC546F823A597C7486DD594F1FEB2B6D03"  
03134 "30918936DCBC12E3752A7072C5FD7B145571DC4A22C770E00F24B8C91E27F2C727FB049D04"  
03135 "BCA813948D2F7B531C906F2018F51C9EF8B7642C0F1CC06AD338BEDE17683E0BD2A90D4D7"  
03136 "46189F792952037E4918D2D6E61C1C8A8B6C437075ABC38913C250BBE6EDDA5CD24B630AB8"  
03137 "FE9123B08D5E8FB8B71BC95BDA4AF1A71D5E54BE98B16D7D3083AB1649CBBBF44B3D5DC33B"  
03138 "77DC1171A66F6E40E75EB86801EAA36E02D138093F28ECC6DA4CE4B128DAFB8D49A17C683F"  
03139 "FAA8E6910B8BB7B832B3A3D0765DE2ECDA9D7BFF0A822F5FD389A04B660F832687CD67F7E1"  
03140 "E1C3F5257C6FEB27B763AC7327DE8FDB989413B6D002FDFFB1DE5A80F1C5E39F06E6D716A6"  
03141 "9ABEF37B2DA9783FC494574689948DB51F78C8812E09367E6C8D70F5FC4139C64B96C71ED8"  
03142 "51D171AFA69D2A646ABA85279B7D3D31D268A2E8A272C91902F12C49D07D74E82728AD61BB"  
03143 "ABDA333DE7138A976A2267DCAB1A02E19B0685F64B608E3DA41108D54A07D4BDD24D79D293"  
03144 "830C04209C37283BAEC00B1923CF404E1F21584F515F9D59B53FA9BA774FA8D3B34B7C"  
03145 "66907BA127A095CD8CBB23375CB188CF9DF22CFB4CEC528C581DB63598DD9A53438409D72"  
03146 "507D2CEB5BD45DB5C760415A5E3B3A909EEDA73B7FE122DF04CB1E9F29E7A5A7385F632AEF"  
03147 "637714A2C21E46B9B3FAC3C13911DBE9668D1127DB02A94CDCE2C8C7B66F0C21601781552B3"  
03148 "84DF0FA002DE08892D0881174CA1F6DE80B98BAE545237C5CEF9E1655F9EF8F8293D644CB4"  
03149 "7963734197F197660CF2767A59FD56B7659168ECD576C485E5726C150DEC6C302456AFD"  
03150 "4F12B2F2D1F36B03E100E109827726020BDBA970A6B768C580E34116DD011299B4DEAAB8B"  
03151 "04D81DCC5FFCCBE57254CA79743D6B00D119CBB2C0B0BBB2FB8FB0B0EC97204E1720F99D16"  
03152 "A3257846D09BF8AE02DB21FCABF03CB0989FB4D99B6D71D5E64C6FACCAABDE63B223A91D1"  
03153 "278D676839E1A5CE6C5EE05AB4F10EEC511C366DBFD7AA1E139AF671CBEDB3160A775E5214"  
03154 "C862E95C3A7993E508D8F8B7ABD68213A531B5C648C537FA2674346FA0B3AB43A7B58BA4F83"  
03155 "B36D5835EC1963ACC189A248E37C9367F3F6FE065EDF6A1205BC80B267C1062BE20950617"  
03156 "1703871A6A6FE41ABB6B8B9188C90EEF88E7DA3338F119770B31167E3AB541371FF7F98A2D"  
03157 "59D355743D8EAD71C96B97F3839F5E28DA6227E5B4", // PRN 38  
03158 "0447B9C85642F3F8EBBD1CC7AB0D0E4D07CAFA93A26C75970C8B40443306AF8BB671F30867"  
03159 "FE4225F576026F79010AED56CE915AD9AC001447CDCB7CC3846B0E291DE0E9E443984E0ABF"  
03160 "7C1C1D2F6D07380B77116A21C3C0B7F360439D11D2DE370CFF8FF73FFFBABFE4D52BCE58CB"  
03161 "32612870B08A23CDA21E3B6E851B063BC76FBF7BC394A7D3805D45616C6A313C887D2FDC5C1"  
03162 "F065F8F3C05ABCF70F31EA5746A513D2DB3F4F3C61006F19420AA82107AF855649EA5D90ED"  
03163 "0A8540D0A2CEC685D18EC8F4508A7D092E52DB2117C233A7BB542B61C26BB92491A88942A5"  
03164 "06ED60354EC1963ACC189A248E37C9367F3F6FE065EDF6A1205BC80B267C1062BE20950617"  
03165 "FE0EB59BF8F1E69982F3E89E7EF85F3319A22A22B8BAA491BE76AC7257A2DF631A98477544"  
03166 "E3E57C7D63616D22789968CA5E929B507AC394B6772096CC70B69ACDBADF900D055881FDD2"  
03167 "09AEF54CE6966767C07EB7A2B4A03410C3417D114FC9E293DA7262171325D8179262B2BF04"  
03168 "1F5A977E2D648900295B14950EDF245C7EEB663F8292625F804BB839BE3A5510036F7A5028"  
03169 "8C69F94434A03466701BF1313020A4DC4F87D272FB9D1B7676C195C0B7B4D3C2CC6B20D5C2"  
03170 "066C7DE5DF85C9FAD43D80FA43BA917F200A73F258BF21ED8BF3E39F69E501AB3EB115C941"  
03171 "4C82C52CE91F6CAEF4A163E288A6FBAD4FA83477909FBC972E2EE2A5379456105E8EA02F92"

03172 "B0D8F79588E2282D1970C3A033F738011BF273EA65912D3ED09398A7F851BA2D6141F97843"  
03173 "E90F5C66B38D71D70DB9D93FAEB07FE1AF5783D71CF50931C282CA8B483D1F2E5DC5AD5FF2"  
03174 "806C4355E46768AE5014A22E7CA26267A1E6CBFAB24C8C436535F23D419819D6D458551EE7"  
03175 "8AEA4779574BF86DE7960969B0635F0B2996F955D863E078DBF8689C142736BD5177DBFEA3"  
03176 "CA8C58871B1FCBFFD67BB12AF82C3B8F935CC507D510CAB63B831721E0ED9C258BA7A3F"  
03177 "E245D42DDA1AD8A962847D7C724DD24187F2508FAC194EC43D17533AE522187B44D71B6E99"  
03178 "3F2CEFC6349141094C8CF935D90F2F238728D29E1F5AA09B46FA7CB651F688A8A06AB97FD2"  
03179 "363486F61D3F42B9C015652A53B3CD66372849FD2EBA05CFB5AA28520B01FCBD9C835F98D1"  
03180 "5FE01C84D7FFBBA4B2DB93E44E4341AE95B8AAA4D59CB88F7E192E89954E22B1B8157FB0B8"  
03181 "9E3E1E3862BD4C2799F2D8C621064EC9BA1EC30E0F7F5FFD7013A7E1C102DBF01698322992"  
03182 "EB05978A5A2878756BEABEDC660DA2D5370A98CA2C0A4C65141EA273EB787B2E7AF646DEF7"  
03183 "DA77A93E45C37E69B474D04FBDC3B5982D8492FFC37FDE42385DDFE97A73B8575C1305A0B9"  
03184 "AC0B0CE061A423556B6FDADB67D0A5BBA7435929D658FB0869BFFBE5D2FEC96051FE03CB7B"  
03185 "D0944304C2D433459A97C43268E43951D4C1708C702FB9C0A0C6D82876057F4D043B21A92F"  
03186 "A6D03B1D5FC28463107AAA814C79022D96D1BD9D33DE5A1FF9E435D9607575856CE4DF5F1"  
03187 "E9E90FEF5ACC4AAFBDDE0F982DBFE78C5ABE1D9C85E0452AAD7836D39591411B382BCCE72C"  
03188 "A1702A20C9DF384F5BD539F53CEE8B9977AFB705D1D28680D1733BAF5F298282846EFA26B0"  
03189 "DD7514422C0EF908F796B38C15EDF6C8D88009B10CD0D6B4B2FD93679712BEFFC4D1F7A0B8"  
03190 "98AD0CDC61D2FCDC360C66FDFE0AFF0E8A7B26359606CD4F47D28DE115261E6F98F323FBDF"  
03191 "E30406BC2ACA0831D2BB4DDF1EF443632A229CFE6BBA5179079DC7FA50C56BE7FE5FEA0C5"  
03192 "AD09D81895508722861E9EE8568F59616890C6370", // PRN 39  
03193 "8DE87744B81A0E21A62CD72148FC3782AB560CE136DC3A07F2D2E143EA0DD2ADAB9FF8AB39"  
03194 "B2A3E8858DA593B6EF6C46BB5CFD252F6DE446D9165406B3DD18CAFCCCE3DC7A6E6200B8F75"  
03195 "91169A6AB9DB6665FB72140A416EF30B75EBABB6C284E73809DE2A3A26C7F4F188FEE08401"  
03196 "E9EFF6DA22EF3DBE0FAA6DDFFB5D8C265D1A1855CF7019D8659062AD7A705362D1510ED501"  
03197 "3B96D6E803E7C6418B05828243496753E8616F4075A08346038E3F3549C0AB1E761790E81"  
03198 "8C2B531F06805092FE53D45B9A6FA5E30E3D6A2F467AECA07D29E1E9123247C69220E2B9D4"  
03199 "501EE42E5BDAD07F0D092B33A938B8FC0EB7E435713E3E428E87DB24AC570E4EF64840A1B4"  
03200 "D43C5026C80F321E537755366B16FAFF2423908FB74E9A5B08F0C1064815472AF48240F1C3"  
03201 "74F2AAAF8ADE55117E0FB1D08AA40C4D8AEF35CE6A91E54A89BBE55BF5E78ED5F66B2FA1E89"  
03202 "36342656C63263D4E7ABD70F7899DAC9A315E9508AE65287C6660A7D7F3FF408CD06F3B19E"  
03203 "1238E6D5EF040B3E54F4469BEC17198AF3F78F660C6753157603138BC98AA9F01FE79036D4"  
03204 "564A7396725E57F875FCDA4EAF80C5E2815862AD52340571A571B331CC8122C13CC58403B"  
03205 "2B61BD404C6D94C36FEF187C712B524BAE9EF150A71CE32366AE536B5B94ECFA351EF0DE77"  
03206 "E729C42BBB32C7D35F9A5BB29D2D84BFC91F510F9A1C907540AE3A80CB7023AE0635EA5C2E"  
03207 "A0548D9D14CE4BDE142436521ED1637C7B7CF6CC1DA5F826B800AC6FFF8B3509A81ACE30FD"  
03208 "969495313F7E18CE4D8A2E1D5F9C7DA14A0A9D4C4D49BF00B622B6AA07FEBFF0A8B274C297"  
03209 "C0B4AD1CDB64BB4941543FB63D9E15059E0C0FD250753B2AA664A677780C39A013E1AA6B8F"  
03210 "786D677755ADA03E1ED55F936FFA1ABD0430DA8750575C37D1EEADECE5E17148DB9FA202F8"  
03211 "748B0611EBB5015F1D26C0D810F1AA5A40D73C32C269EBA5CDA134267231FC3783D3D4CC63"  
03212 "9E567681275F423439A29118C6A0C1B07D08416537D4707E9CA3FCD3494F64B69E2AE9EDD0"  
03213 "079CD428CEB8A00BE0FC9A791FE2DCFC10D7F813315E964828A4DCC2A2D42EB31CCF192F3"  
03214 "2AA9A17C984E0D3CF3A0BD86E0B751B30C096F5F0B08A0BC439294D2BE2CB387648F8119D8"  
03215 "820D39F17C6AEC976A0B8C89C76D12AFF73059B49EE856B2591E8D2E817DC43793F20B2AFB"  
03216 "BD49FD9A05F5B4CB69165420AB96F26A46861AD9423F7DAE8829FD392799FE967E1270563B"  
03217 "CF3D46025CDBBAFD15DC23F3EE5B621DF6A12263CC0A506A5E9AB191F896D13382B3BC2D5"  
03218 "36442A62B03FC2C2252D0BE377CFBD59097259ED243EFB36C9AC0AB01B3AADD502DAACCE1"  
03219 "74A9CC82FC9AD67B4EDE81D9355CD8295DE21468FAA25036B2DB6E24A3AF3E5FE5932486"  
03220 "7658EAD5198C47D362EF64B71179D107DA748F00400F15167E84B62588F6D81FA4B68A59A2"  
03221 "4BB3D27167D0720718BD24EE556FB72113FCBD37831251DCD538F6815382E119355CF3490D"  
03222 "F0AD552EACB0D00462C10CBA4011E95C7FEC968AF6E39E1FD15D1026879202C57E2CF7AB02"  
03223 "B08B15373C13435D8A55ED6B9BDD98FC5B4539428D90A2C73D37DB112D2025EC8A4AC8EB8"  
03224 "2C51C46FACE88AC0E1F161705801C781EA07F6914122E5DD6BFFA3812EE44E314BC09B785B"  
03225 "3445844A8522B63FB34BAD2122F9FEE1245E6DC837DB032387EFF5036A021C8112CDF03BEE"  
03226 "FD89EEFE1BB88A132CAE9E1E9EED3A855B364F2A4A2F81086172FEBC64614BC04D11E74AF0"  
03227 "0F54AF6B5D85EE4773644C3FA768532B1821EC6E5C", // PRN 40  
03228 "0D1FA0561DE4DECF411AB73FF48D0810AA2149FE5F3C22E62E06C02F60189AB5690A991CB8"  
03229 "8DFA5C4FBED745FCFD63BE3ECD9CC599E35B5FA31AA11C62F3A33796A64EDD0B64F2E51E75"  
03230 "C2481D3EC9DDF07DADB71448BB336C0DA9DBCD897F777E3C9DD97C7EBB08827C19316F6142"  
03231 "0D96F3D94EC395F8BF88F5927C71397F6C24901914826B1BE26E14FE93EED37141109CCD00"  
03232 "C92772B96D9EFD74B3EAD3903854801B22DC3F98232023E1F9CDFD4B4C952A6D268DF9FD3F"  
03233 "A83C707941F8F1ACDC195AD66E7AF48D7B06C62F3204E8D03D2C92920C589CC56A5C2B0A1F"  
03234 "10401BBF747B60B9D2179BF9CD7DDF50B10823372CC4E24005D73B4C5AB918F22918B16D98"  
03235 "F15C365FE69786C1ADE9FAD71516E389938D5D38420FC79F82C6C0AB1C6D93D1896FE8A2BE"  
03236 "BA13413C8EB1FB00115E50A8A676B9D9939E7375F4B7D62D449EB341B0C03EE3FB18564DA3"  
03237 "CF64261055594E0F8D322EAA9B56B5328574AB323C4376464F3C27786DF7026BFD6C7757BA"  
03238 "EDE7887217A2D3EA22AF7D6809BFE4985EDB32ED503C6034FECCE5F4379920C73A515C7390"  
03239 "B3405ABCC54F51F686903D5354468AC566D5A03C5AB536934A6CE698E06B9E2C815B15B4F8"  
03240 "A1857FE7C0B5486759B150F48AE9745AFBCE856830078EFD7F4CB4A463587B7DDACC79FC9F"  
03241 "BD9C2A0351D91DDF27A65CC79005DD24F1C26252BEB8C3F1D69DF79D386401EC5AD57BF212"  
03242 "9D1E828D6D57CABEA84C7F77B16B919653C12AA7F770741E4AC11F0AD08507C3E7C4716F14"  
03243 "F14C548A88DC4047EA96CF96BC0CF786EF7D02424314BC846682109A2F80044B51B12A0350"  
03244 "DB9FE06E72581EE874665B9680EC118C86DB575F9E5F687CF35DB3CA83475051AFDABDC974"  
03245 "B2991B47BEB4CB967E3CF20D2B7CA2BB50825EE33FBD7C6E87095DE8E36CA6143F673A8C18"  
03246 "882D89F22C08971CC709F1518169123EE59EF50846C95C313F14F9472D4CCB8EB713014889"  
03247 "01F27E895B5D44420F91D123E3EF5527121B9B0C9536CC3D4E5EA03F63BA8FED0FDF593AB8"  
03248 "C3E768962DA165CCACE066ED7D6E46DAE97EA08789C36BD3CA6888229D15714302A39EE4F3"  
03249 "9DB0751F0E99877237524ED8EAC5F7EE2CC40A69408DC43E2AD88A661D7E443D7E99A14484"  
03250 "E3011A41912E2AD4F6B7D62D8D9F3332F79A5FE1E16E8064C91DE56B1F89A9A3A220165872"  
03251 "128771745D086DD3F8DDDB2C35AAF3DC6DA70684DF270ED50D188FAE62CE4C98B9019D30877"  
03252 "2EF036F9FB2775137377CE1529A40E03B388CFDA3BAD55FD62D7ECAC2A72FB68068BE0838"  
03253 "3C36FB66CC25A0412CFB3C993803C038A1828E8893DB453D47C5727CF8BF80850346063A"  
03254 "F4976610F824ACCA00BB5963D5DFD018E0790F4A6B14A9B5FB9517783EC1F16773C49A744D"  
03255 "8576F7957D4CD4C4E35D1BD580FC76E2632D47661A6838344CCF6476BB793FB12F92143524"  
03256 "E5A01DD529486528C5A18B24B0B75D017C4DB3663BCD561AA8F27E510A7C031ABD3708385B"  
03257 "03294E3BD1A695C573701B9D4F60232DF767439E252B7DB10D4E3FACAC8C2E8D16F9500828"  
03258 "826EA807C2C7C48C0EE8457E757854C1540EFE9BDC7C09AF5BE09AEA54A4BB0D2A328092F"



03259 "3B8741A128D4BF588384A3848A5E4F55DACFA64F7FE081366330D16589991C52A1BB782364"  
03260 "AE969E8A21260518662B00EDB117278170CA2C0D186B173F93879F6F702E03DE768CFEA1E3"  
03261 "9FF8453C7B3AB46E83FC90FC56D0EC0847DF16A4BF1ECBF29D94CCBD49DABDCC5FB81B62A"  
03262 "2CDE7D2DA3E67042944F33A007096DE5C79133C7F4", // PRN 41  
03263 "C9CF7D2DC0A2341FBEAAFA64F6A50A93F3FAE3B55EB70ABF51A4C001E15A8C78DE54E50C"  
03264 "CAEFF8C2421079F7333541018D5EB8045C3AE0F0489D13C593EED560CDACDAEA39BB49FE2F"  
03265 "D9C956197967830A0886182D98B5051080DE887638F2E3E626B8941A58D2206B46FB410C15"  
03266 "5B94FD7D15956EAE9C9F855B96A8F1C7202EE600C12DBAFC6E3DE5F075B05AFC037847BB521"  
03267 "D1D3638135591B6E1A0137CA74E27ADA3FD673684A8A1330DFA46D449F8B63FF90E36881C0"  
03268 "9FAEDF61672AC32CBBF71F34367E7547C13BF08C4B688A397DD4D00D8B19212F85EF11C0FE"  
03269 "A19C89D73DCAB9B532CAB7BB5931B2F648072F35FAA31C6A77FAF3F72F9F57D5AE65BBC7D"  
03270 "FF1C2A5A2A2721AD5B50DBDA2CEBA27A3864A6628F24B0D24FC2450890E278C4599C1015C2"  
03271 "B4AE61F2DB2A3A85F02882DCB57A7A471B5F8E51DDC284E0C1869D41EF25E260EDC53AF55B"  
03272 "E8217F8ADE6C74B9C0EAE7C3417156D671577E05A63E03052663F420B70E99DE068971C5CB"  
03273 "8D905586446D6B2C269E4E5D2585BB7A08E28B15851C6E8438EB127D517A076C08631A9"  
03274 "890F4FB627D05B91ED2164355D7F20B32498E4773FFFD17F3B15E7F2BAF5EBDC98E617A743"  
03275 "7BCF94B0EA1B2FD6F98F8BE6D64F769FEA28A5FED9DBF4B3700BFA8C0495772A14254D7AD9"  
03276 "52586EDCD0C08C43EEEBF91D2B17C9ADD071CCC83EFC6401A22846500ECCD3D4E5184A05A"  
03277 "ECBD92808AD2DA42C57621C66142784692D4F28662A2B37DA4D3CAE332A97D465D3E492ACC"  
03278 "964AED603058CB6BA972C83C2906CFCC813C0603B775E2DE8BC946DAFA510532CEFF2B2FC"  
03279 "5E1F255401BFFA078577003D67F8A3C63E9BC2B89EC7AEDD8C2E91CA426398BEC4D98FB623383"  
03280 "2BCAB8F10E5DE6668F7DF8FE10AC0EA63B7D46837EE4D1D203D54615042203552430DB97C2"  
03281 "4FE008834BA154FB2157DD2D12715ABC6D329A819A1A664727F40CC26204949926B78B0BB"  
03282 "8947FDA9026813FCF19795F8BFF2F56DB70DB7EA9280A6C504134A1B653D336101A9C16688"  
03283 "3599477DB557406FAD495BA31B488D9EDEB783759EDA32968AE4B4B96A14C9DA960ECE511"  
03284 "D4102E1EC27E17AF2746DFA54F42D2133281ECA9F5CA87F8A2DE4C937B5F9526591A583DD"  
03285 "3211F78871BBEDDE79206F946B9CEC6EBD0DA4A25AF259E4B3EC2195578C74B7DF87963BF5"  
03286 "61D7F4C6755D4DB0F288341D5E525B15F2BDA314DA38D2B2C7CD7D64F6DE35EE782A076333"  
03287 "941FC3505AEB1F5A783C6C2553D73AFD6AF896A5C2E1C15EDDC408B6D9D7EE2E7E9F706C2A"  
03288 "6CDE3B641BFF8F8960612C46996A9E7D43511302F925351240B5B77A7183C45A1B66C8FCD34"  
03289 "EB6EBF5F5FFFEAC8441BD1103D2151689BFCF8D4E29B9C39B1BE4972186B099C2C35572DB1B"  
03290 "E2A28D751D6857B7E4EF181B786C2A6FFA30A5D6E2E1120E7B1F313302ADD1012E43EC24E5"  
03291 "E5B5D03950E009272452ADEB9DB626FF38F335A520B12CD981636BCFB882FC06B440E34851"  
03292 "2E6ACF980A1ECF360E33DCADFFF5A2B413DD214D5AD1070E88B63FC9765295BB9C2C557F3A"  
03293 "AAB6AEB293249761F95EE4338243D35AF883AC33832155AC758ED05AA1BFFF0E9FD58E0338"  
03294 "4704DA5A9237F1AAF52E12C821BB0F54C0820AF8124AE6D0F4BFECDA6500995B2B61ECB062"  
03295 "B8FE2A9E6A05DC94A781137522E85A3BEF28B76E49B27BD05C113C9B3A9AB1285BFD926D65"  
03296 "BCCD5CC3837490B21527A1E94544C4ACED0352BC84D378625E3655F74B039CD681D98579A1"  
03297 "90AB2B9N104C83F46235757D2D22FEF59A250B564", // PRN 42  
03298 "48116D6CCET37DBBD1E1F45D9676B031D9B0E65684FB88A83D791CE8F55278D33EA9D3A6ADF0"  
03299 "4F29AE93123376D1CEF52F27E875EC0024A888E688EE2A21A4AD0448103569C8A7CB2FB07"  
03300 "2E3D0D09115AD8A1C234F7765E77042156AACCD4DA8AC697E41B71F436ABA09C91E9E4FC2C5"  
03301 "B573CB1D0B291F915BF922FC75E6FC6AE332B42D7E51BA8EFB1CCA66EC849099834E48C7F7"  
03302 "9BC16C79B0C87A8EAC8F88D30EEF8FC0EEF8C89520E2602A7226BFAC0E8B30A117B188756"  
03303 "87F81A6125D09EB3BD2B2F5209C386DEC5AFF7DAOFF5454FF0451F825E5AC55680EC7DEC56"  
03304 "E72CAB953E5B822EA01D95F3F948E95DA7078C850C43AF46B4EFEEAD6AF807CCFEB1B61CFC"  
03305 "EFB770D1923B41BB354C81F340CFD851EDFD537C373CB9F29303A988635C5CD37AFA576D6E"  
03306 "EA8C2B7C56D6E29D49E5DB970B50E04436B1F8B22A6DCEE3BC27EE1885CEE23E12541554BE"  
03307 "2E2FE615A140831DE6C555804432CAC0A73DC2EE5CECED139ABCA4D17B22BCBE60F05F5D7F"  
03308 "526ECB29A51ED6ABB4DE4A2865AD1AEA64041EB0F505BE3CB8F351BA21A8D941902BA56503"  
03309 "9EFEEAFB8715AD8A1C234F7765E77042156AACCD4DA8AC697E41B71F436ABA09C91E9E4FC2C5"  
03310 "A0A50D494FF8E0B902A9EAA4B4F9D3BB2BA1FFD3DCC93C45133B9008E2B188F57A82B5DC01"  
03311 "862D35F633CD6CC3FC12BD6F3A20C1EEE1F9018E41027788950DF09C4B482837677B33B15C"  
03312 "E24FE6F797291C0DE291AC0F04AF975703BEFD3CB8E59C1496E547285E3F7A29F99C5F210"  
03313 "327CB328B8A3094DB7C27CF7FA077791160A2287F44F92BA6F5346E48D4D4AE6C2C7185926"  
03314 "526719537AAD530EEB3B0B4CD718C7BA38BDBA21671ECCE0AB4044B797B7D10797EB1BAA02"  
03315 "4B76D08B2FD5C94655545227B46DB2A038F87301BCBECE0C99A98AA5BCB648DE58B6CD59A"  
03316 "C6471462BF5985C595B00A4DDF2948A84203890E2989D8DFC9864392E7E75262DF1CB78C1E"  
03317 "2872CFCD22D561F7888F42A0323CDF3EE1588EFF2F7BEDACF5AF710CC7C6F9957D992A7FE3"  
03318 "9DB696DA414AC36BCDC99AF05023597734D94C2A8FCBF41D6004C21E4F4357C19ED094887"  
03319 "97CA012F4BBF8F1743E396236E708D96A7DAAB07FE03D5483319241CB35F66F66D99D981DB"  
03320 "4427C4C849EA24AEE4F7C4E023E0ED2FEDF59FEB6D8AC0FE87DD42580F7B0CC6339752230"  
03321 "A69BE83AE13FC12AD4719D45CAEAC6E8695F6F403A0713ED37DC120B80DE82F7AC4991B4B9"  
03322 "DADB31A304E16D03251CA140659CBF99E891DCBB764293A967DE1E96848B8F225D2F9D28EC"  
03323 "2F428AFEBFB57AB79B126E6FC3D66102E3813347D7CDA5499F6AA5BA6AB5EA6F3A82D730ED"  
03324 "F3FF74B9BC0F954BED21E77146A5591941962C49F6B2EBED060A9C82D8FC780FB3AED527D4"  
03325 "QCC426988F042841250B4A85A3F3501FA3436103313C3C4CDEA772B6C840DB866B9730798B"  
03326 "69D6AE12D24E452B731FA9B4C8DEC6CBCEA2E8A0F9CC67CCDCB64523725972D81ED2E65622"  
03327 "83DE99297458452B94467B83F3111CBFF61006814CCB6550D19A1EC026996695D46C31BFA9"  
03328 "BD371CCDF1B512C61CE7BDFBE419B477DAB1E7E1D012DA4BDB05F4C4CD39511D5071C09827"  
03329 "68D9C6F558D22940D541A3C67003D05306FFC070C2131832F737872114566ADF3074F7F62F"  
03330 "4AEF9B894D0AAD5ED60998A0363B92D841F4A441C39965FC1479E93C99B083CDE6CD6CB9E2"  
03331 "6D49E16B969A357E28A61B444AB0FA665B853FC89B64E02BB02C5273C948B000417014594"  
03332 "08C95AE849EDC7CB605BD2DF31C3CBB2AA9FC780", // PRN 43  
03333 "840BCC55474C2F66B7E16C6F527695D367F70F570B13C1ACB546D8439A1AE06F362C224FBD"  
03334 "86441D82430A345B8458EB666BC93739E3D679754117C95100E8F00879B891CE03788543C"  
03335 "415AA964C3F861916BD7C141777A93B374943CC4CEDF928AEA3EB72F412DD8256227D7D724"  
03336 "4E876955B13FB2EAD1C3980AF6F8F36B3E71FE3B189F8DC22C55E3720DA11C6AC4328DCF67"  
03337 "0CF948123FC7280D6891EEB356C195D2A79326299A684F5D68DC0ACFF00B780DCCEFC7CBEA"  
03338 "140FB5B6DF896883D3C6FFC42BAA41E6A75713F8861C18A833C688C2BF6EED7F0CE66242B8"  
03339 "196C7C254F9D48FD56AE5ADA9105A5C4AF28D0042E059A90979C6439BF6D498991CF64C680"  
03340 "80B9B5392C9FCA4397647A9AAC42C3810FA33BCF31C89A641D1078A0822D93EB773F418B9"  
03341 "B3B2D05FEEB1067766AC561525F88F8F097BED0CBFF5BC498CDFC9A5B9D20F4D4A03CA1905"  
03342 "63CE543B2EAFD8DC6E4A5885DEA120DA7D157D00EB1329E500D41C7E6BD603E95BFAE227A9"  
03343 "B613A71C7BEFC8449B688C59147E1DD5411CBD967CBB46A9DDDD76C547506A54A824532FF"  
03344 "9AA5312607F1AB99F4F9BCF85D93B58D76C5583856595AF5CA1AAC07C945400B3588507180"  
03345 "2620855E18E0A3B1EE92BB95241EFCAD82B3435BBB8916C4866471BD5715882F6BE508C764"

03346 "8A59B2AA91A69F60C78482197C04B31759054FFDAA254A143021E37874DE407690B9ECF8DB"  
03347 "99CC16B004116F51DC7853370BD3538E14A0F02E7BF74D66FF898AEE93BF0697EB6E8E62B9"  
03348 "5C6C52A03E1A89CC2F56BAC591FA81BF6A1B3B9F1259F13AB45F2E9B0D93DA7B498A9C71"  
03349 "272CA261CC4EE4A44FBA4CD03C75D216672C18A60183A52B28796B356EC498F4926D833D94"  
03350 "007E92EC190E9713354C7506BEA6B7B5DB2BE659E2A5BA92D00C01611347F67971478F2C71"  
03351 "648E33F99EF201367CA3AC8814DE0F83B0BAFF34A192349C5DE87453167820C0EB0AC3849E"  
03352 "CA683111776980FEBDC993314278B37B289B528DB319CF59B09B83342C71FD5E9F13EDF4F1"  
03353 "8783FF328BC67A03F1F62FD1A4631AFD76363DB1A79CD1EA95FD43934955AE211E00EF5AFC"  
03354 "114BE00EAE2A7AB63042992E086810E61AEE3CE5CB0FE75C04969485899D552AD4B006D543"  
03355 "BEB93C1E8578168C66667D62CBCB56F98DFBA248EECD891F9936382087F2AAD6F83FFE25E7"  
03356 "7F8BAB9C62675D4024E9BDC5581126BD1ED6035994029BB54615E593D4CA81B31A5AE8CBE"  
03357 "F8E1A81C32698FB7B4A339A7E4E575460F79C4F4C5F57F9E8DB459177067869E03E2E16810"  
03358 "04EBBF62B4F60A9EFFE47921AD22C7236F25438AF46885E120825509D1E28652361EAC64AF"  
03359 "1806317416E5550A39AA8DAE2C6B970D44F4F38D571F67E304AA57BA435C4E06CC97C84842"  
03360 "2216BA778B2736C30200527960A66D63280AF7DC691F9D0540AA47F0D26416B46A6D5380B9"  
03361 "3F1B7C3D4DA8BF8AD2FE410ADDE00B0372E9E830C09B206A08D968AF2F716DD04D220E619F"  
03362 "DE28FE702867EB9C2068F7A37413BD9E65F1B59C331D45AE914731A564D33875A0939A218"  
03363 "5348C197D14DAC59C5E2BEF09876353FB16D849EE67A774B05C51AA2480F18DE400516543"  
03364 "D2C7B9E3B753FEE5696B10BE40BFFE89487DA1E8B1C2000D53B2A98B4DE6C9DAF07F0B4DB7"  
03365 "2834361E520E45F5E958FE0760F229C16CCFE0BDBF07E186B28AA69231DDA8378DD6338A94"  
03366 "D234541B9D8863B85962C6304E1ACE2B56A3E433F7E9B2A88FBAD5FC71703231D38DFA8C378"  
03367 "E15B6503A3B9959E1759EECCBFF14AA09E6BB8F2CC", // PRN 44  
03368 "152004D3334E877FE21F773408369872718CDEE15BD489D97F606779A9A38398524E656026"  
03369 "0980FABC7179ED91D699D52592AFB3269FFB8DED36AFE7DC79DB773DBA560E3ADF8E12FB8A"  
03370 "08EB56BFC4692BEA05AA288012B580E43BE085CE583E9C9DC18A9D32567EA950D8165A2F8A"  
03371 "443B7AB5E79F687690A8D8D92FFBF877C5B84D26E6FCB3671073BD808A7E908130A845C95A2"  
03372 "F5E360645611B4B55B03169CC1F50AE2FFFAAF50FED8CB782BF3C67775B613CF82AD201219"  
03373 "9E49A329496C0FA24AAF6370A3636EAFEB992C663C308AD181DC308D81FFD7135059163007"  
03374 "32FB1F8DCE238CC4104D26C588F28FE7EE4F2B8C2024357D52C7AA29D339249086CB2202DC"  
03375 "1C847210EABBD3F0A39642711B4608638564B0DD2BDF38CD79CC1CFCE1226CA8356EE19"  
03376 "E4089A6F18E7ECC3597CAD3FFFE13A5F77A58C2B079DA25B928396FAEE81FBD9C6243CE0E"  
03377 "B0872A187E6CF99F16B1511672821D11EF88884964BAC6686C2A7732035493FB9765B7A51E"  
03378 "845C1C79FABD99438FD84C09642FBC7CF1740DEB012AD050F5E8FF03E859E85C51D2E87AC4"  
03379 "1185C67C33820EFFF2D79D384B11B44A806ABB8247AF36E397222C805EB9854D9FCCDC7E"  
03380 "58A101A5239E60CF4836C38A6ADE67686E6487547933859D46D046BFC15AE95B8D0A42A57C"  
03381 "401EBE78ABF495EB390B8922C97270C790C2FAB9849ADE48EA8F4F7C6996F6DB661500199B"  
03382 "CFE1EAF1ECCFAB1F674E0BEB1D03319E73126D7191A64A80FCBBB06B1CFB718275BDB571"  
03383 "E8F53FAF81DBBC0522263FCDFB537B3BE9078F2AD449423152BB9DCCEE97003D1988540FA"  
03384 "E39CAB62593A9C094989415EEF4229358229BDD45CC35EBDBA088C4BA097BB0E28814BF5"  
03385 "048F4F64E6A587B0BDA8A8A1B3E3E69D3D7B6782FC42474773C61DE12B4C6F183F9E62E533"  
03386 "7EFD9C7A79A5D2C0C939FA668F440688F7A3E0E28A33D36D331069294DBA8BF768010AD93A"  
03387 "72AB84F4A69CF8C04F0ACE8AD1329A4FB594E9EA30FEFF31BBE418CB5515CFF8A375CBE5851"  
03388 "5437B0F5422E3DA99EC2D53B22B1A55BC783CE86CC6B1BE7B11F837CF2BD09A514B12ED7CF"  
03389 "0AFC2F70F1D6D08288524263B1129135D664A48B0596C4E880E4E01BA3889014C07ED1B7F1"  
03390 "72AB84F4A69CF8C04F0ACE8AD1329A4FB594E9EA30FEFF31BBE418CB5515CFF8A375CBE5851"  
03391 "5B32E799E8449101093A053EC99F9CF78122FA2D36DDFBD258EC81D7B4D18BF882663CBF24"  
03392 "D2A35A93C00AE8EA81A3B0E8040C1E259515156D7DC76602924BA51507C4994221A3B1CAD"  
03393 "674BECB11574212EBCCFDCAF6AFEE288BCE11B7FFF1D7191AB329B04A237F75B2046346769"  
03394 "32FB0842E2AB88E7D5A4A6B7F77EA04ABF4B9A567E5DA5267F35CBBC928B5F607E9C08359"  
03395 "F0031A934152D77E65C937181C92EF4CF17F6EA45171DEBAB545755795097766318CB132AC"  
03396 "D96CDD65777F41BBA127CF251A4E9B3348927757A42AD8FA83F6CE7342D935CB54984E45886"  
03397 "D888627F0228E50082B85E8E9DDBECAAF049C25C4D7B2EA919716271500D81EDF574C5AE91"  
03398 "E5F0FAD7585EF69C5051473AF913F887B31BD730F67B4A44081BBD567B57C061AF46523737"  
03399 "4E0BF4753F36CA28D45220CCA3EE6F07F93A29982EB5AAD70763ACBFBE55015D1C52B6495"  
03400 "4D4855ABCC319DDCAEE75CDFE3B3C31C329A77C76131F18F3C3904B3EA4E691C03C32CA5E7"  
03401 "A1F9460760231473115B5AB7E159013C96AE4885566065883CA3B9661C6773B77819BC6875"  
03402 "5D529FCE05E8E08A5097EB37BDDC1AF9BDFCD0A04", // PRN 45  
03403 "0D4897E7FBA496EF0C6D62C034088E7E9B8DB52F232CDC9EA88429655ABE366C54CD534DC90"  
03404 "38F45AC0D0362E31086CA2FC0FCAC3FE71B2D4EEE548F69CE4433601CD3431D01BA6F0C23D"  
03405 "612807EB8E50BFCA736D917F67DD76231BEE8728006D2AE69D1A1BAEA8DE50852CEBF7491F"  
03406 "FF989C37A54EAB31652230A0D3E4397B051222C4A22BF481BC52E2012188C7576A7ACE8118"  
03407 "5C85A2E841119B47B5429DF277A976F15D5652156445905EC979729A6647E1B593484ACF27"  
03408 "24E81786A0C062DE87759E8220A561FD077D64FDD3226A1E290F660722040EC7AD4383431"  
03409 "E195412DBAD0AE620BFE808D65D3937AC9C5AEFD6FA106C88A6B192D4FEC1FB1A06907B079"  
03410 "6F724AC6A5E5EBF50C168F990BDBD6B1D23E098858C07E8D2144532AFE040E45B25AD9AF7"  
03411 "D92F9DCCBF3DF25DF574A4B65A219289407EB63FF7152BF1E651C6F124BE32545B50E8E2C"  
03412 "E37F1042A68A62AB70D9DB540D2B39293E008EBAA632B85EF1052FDA17A07B9B1D89A76FF7"  
03413 "E7DEEB7090496855B29A9174780D504F7228F8B7436F4676F51975BCA614554FE6B3F946F5"  
03414 "E0D1016371F9CEED4521477CE8E9CD0668643686D8482CEC0BB4D6A9E8678B1835F0952E27"  
03415 "00F0EC185A8683DA91638D81A85231CA4C7B55C0C0DA57DE5E6553F22290CE7EE72867285D0"  
03416 "96C0FC16A60A664C6BAB048939FF9DA476024235DDDBF7F264C90ED44E953C4AC0FB50362F"  
03417 "B2CB4C4FC2097F07B71772000D2C2A5DBDF0B6FB83B1BBD70CE9D6A8BF1F924AAC4DAEB6AA"  
03418 "DD891A8974C9BF46FC6038F6B77D6AC0D6107AE6AF5B93B186253AFFFD8CD21835097C4B76"  
03419 "4F3E0B2447776549D5045E4346FCD3B7880F1B3BA794ADD6B7F291D3534DF034F822D0A41D"  
03420 "B2AF9F1E51F212D5631D2525B1804A75B50D8372B6A1C200B81AC1122AA3C13D7463C383FD"  
03421 "123DF4029BDAF6C227B3C4B54F75C5315FE139F6A0A3E2E819D3062B3309E2630EE4A64D1E"  
03422 "3DBF23B489B96E198BA98D8D3347217A5FDE4DE453012E8564BAEFBE393E59C0A725A75CE"  
03423 "DE373501F9103011FE0EAA40C583AC73C92F47BD528600C42E0029FB3A4F03A3CCE87E5AAF"  
03424 "4FCBCB47EA8BC62E0F542BBD455F7163A2A8803583896E3C803C0D01846EBD8BB1B3FD396A7"  
03425 "C7BC3F72A9D0503C44FE30E5E77E0FC8B373EE931CBF4A9867406BC91972B946B9ADE62415"  
03426 "CE2FFC71CE64DEB7E6205D635977C778BC40C11E36BFC22A17FE641346C74DADD255D329A"  
03427 "EFC2CE5B4ACB195FE2DB453F1B41B3334D5C480BBF10FDE88C722091ABC85667F2B52A350"  
03428 "5A142E4D1456CC41E154B345E85B9CD0E241D70EB4876A70623DFF3FF43A8D66EF60488A"  
03429 "7860485A8E518AF4A7056CD02C72E1BAEA3944781642A36932ABFAEE3B88B57B213F9B358C"  
03430 "C3F05154DF8D2132A74DF27556DB6C5E45256481C09D18DA604CB91E6AB7AA29C0C1009A40"  
03431 "6F1B3AAE268564E04B07E4E9128FF02513AB8A8665BFE1E352F46DBDE7870EE027A6E84560"  
03432 "014659888EB6AACF993D7C9F9FBC2974CAE290059802F5D224520CE445C404A2ECC57CE6FD"

03433 "5961EB20DC12D436F0B93614CFC844B476D449416F54459BAD09F886CE3C1C307A91E62FC7"  
03434 "36051E6F6F6F5AD3372398458299EFD1C809997FAAAFE14B15566B381714E3522F629A5ED"  
03435 "9DDA1EF4B9D4498DD55F8B0A48ABB2F9B1FCBD991B5465108147BB6125FFEA6B6D479DB6A9"  
03436 "983A51C041375B39FE6546AB5B5E089DD7E228D5AB17127A3A1E3AE82A31E952E660F72A1F"  
03437 "399ADBE4E1E11C390BB8652F2F7B6D05E9F4B6FC90", // PRN 46  
03438 "AF6D254288BF564C4663014C7A70EB3739D28F1F17B8627FDF714E85B48EEFEB1FD4565463"  
03439 "28BF2E2EED112992AF78911A89E1248AEBF1F79906EA91D41CAFC6E541D2BB40A7050C5AE4"  
03440 "B66547C9DDA88590AB8B0A4B01B5F8216447A310D1E52C155F92659835F1086D381F1F2995"  
03441 "138856DEC6FF59380FD6C1B6EEA6E4061930CEF7217701DDFE17742721794DE824377392AF"  
03442 "3D949ED9B0455C58F7395DE0FDA5485E310E8457A4270D603BA5AD1854811530FB7178525F"  
03443 "4BB1E5DFF8F96FFE0C83C3E47E244D8F3F8AAD8374E3FDF996060E18D2A5397192823681CE"  
03444 "6DA561816C0E0C0C0600732A37731ED13948439E8961BE44D32A3097C1AA5E19876ACD21DF"  
03445 "812265181774580A8C04ED4FD1DE2FD5A1DFF9DCDF4D97C448BF9A4DACC22D4A55F588C87C"  
03446 "CF5C9E178A8FFA032A454908BBE41E13D829FA98C066A7736577CDEFC6D5441656D0AC95C8"  
03447 "72861BCE0A282ED6E9A20D4D167F37F96087F9E761F38B882CD41521DE8E2C40564C0E001"  
03448 "EDBC04982669B82F05CD0411A6BADD704B545FCB19A286BFFAF0FB6DC889BCE75C5763BE"  
03449 "1D70DCAD1D39F395A75E637E0663F3D5A855F53F700AF5F82124E72CC2A4E41A4B43FCBA41"  
03450 "E55298A1B799368B05F9B507FB11E8BD2D80FE517A5FA935006064278CDDC5EED43B7C2D30"  
03451 "9BECCEBB2C0AE3B8D09509315933E2DF91304B1DD3777EC9FA2415F7D18A96CD08095A30F7"  
03452 "EB9BC12F8739E409EE7DC062703CB3EA591531F47F5D1C0431E9F1883C3BA249492EE61971"  
03453 "9C8B3BDFC1DEC77E3C4CE4CE032FB2882247D1D54566E3819464837C248D35647D273A87F8"  
03454 "BE667760D26A7EA1FFA682A931A4D35F7407FBBBAD725BEA62449623AADE97A2789F2F8642"  
03455 "F1CBL2A06C93861DE9D27C5ACE0E4ECE40B58664176113ED6437B8AC8B4059653EA30F71A0"  
03456 "23544BFFA685343E278129DA12EEAADBDD5AB0E2CCA0B840E87AF0D0E859449893276B9B58"  
03457 "94EB0C57C3222E01E7538FF7E91807C10CDB465A12CCD358151D96856C6143C9F8D25AF8E8"  
03458 "086C83D4C3AEB37BD660D15C0C0707691E8366C4AFE6DC0AE1BFCECF9BEF46CFEA3356A3155E"  
03459 "027D1C921ECB852BA4B8FE8E0C8C00829CEE4049A21A1C3424D2BE532B73523D2128B1C3C9"  
03460 "E5B02294701E6BCBC8607F2AFCFB2DD9B1142B4D261418C1B239B0B22451D7D3AC64636B0C"  
03461 "7F96B41D7E775C1CFA2E277C957F0D59BD1FB02226A462533850E99E917E310A684E81634F"  
03462 "3876D6AB32CB58FAE9E8224BD2A17A7C72CDACDC6B3438A083A6B0DB9FD91E05BF4C83A3"  
03463 "6916EF35C5D4871E94B1671493FDBC497BFA30328DD24D340A07A960828A855FA95EE33FE4"  
03464 "FCD15B14B7929D338611239AC34CFDA6503E4DD9570EC339CFDC8D1C3D3536F5176901D012"  
03465 "BABB133EBB4560484344E044F86A77C21183C82BB2B6632CF69F5A661C98EFD5529F8FB96"  
03466 "5DCC6AF185CA8CDB4D01C5E8902966B063B8D1724AD18DAE495AD47416B61321DEEBF782EF"  
03467 "7A6ADE3CD0D5B9ECA84DB5D50C2E5C7D8B8DB9E0F398376CDE3D7E8DC46F0D77D38D6CF605"  
03468 "5D6D074CBA92A27CF6F9018CAE87C97F1B3ACCCE495ED9DCE46DF4FA5255DB2C7F2669F5201"  
03469 "058E90F7567672162E17D5B6D028C6AD23F2AEB32061E6B33EA86915FAFA2344CC58162C0"  
03470 "32CE837D5C32F5993013314BD261DE6EBB35B94B8F690C5624A8CF5509BF50D3AF5FDAD773"  
03471 "11B6D9398213AC1A267CAA0F52A262DA4F85F5ADC79947D2C5128B8E0B0C41B9FDEEF6AE"  
03472 "C228EF97762CDBD78273C16C39E446576D54A4FD0", // PRN 47  
03473 "4B7593A9ED7897766CACD515D3A55959FF99C1CE28A745DEC1D8F2565F24F4A1E14E9083AF"  
03474 "F510F106D982A2911197338499CBE38CDF3D99463B13EE26B47D26A62CE45EAAA04A3E7085"  
03475 "D05F23470FB94C42D3235D5FF9E6C37CEE8C93591CB69E0735B03EB262CE6BE6F0144DFD66"  
03476 "BC089B36D66287EA588C78E39D9B6907EFE85C1211612952CF13C369C2AB3D921E4630ACC7"  
03477 "F562A699014776B26CD1F296F736A4616FF662D5C6E18C4EDA6D1791A71BE969556FF11E119"  
03478 "2D3941F8020D2C731403ACD856A3AEE6ED7F23023BAD7BF138C702B6449E2601042D7990AD"  
03479 "B988B650AE202F3433CF26EE132A7CB13650E86A6DBA6F7FA53B2354DEBF1268734D7120F7"  
03480 "21E18FEEED293C268A2D3EA012F3D7F68DB0B18A5CDEBE13CE4A05683947DD985D4AED1E19"  
03481 "2FBD2719755846C9B758FC8FF28B9999D07E634645064C2C9DB4CFE50BB8A030B60F43AFBD"  
03482 "588EC17102F614D3029FA811457568D7726C651C062391A2EC2843A95A3A48AF58A898BB65"  
03483 "FB852F73E9A82C6AA9D406D80C072A3B426D8EBF261BF7AE0E9DC0C6DE9F4BEE880D775783"  
03484 "F910AF19DFD8EC2656213FB9B74EEBD8E1BF860E4650D446683B7794086ECE1E2AA723024C"  
03485 "219E3DCB371624C6D721BB60C797003D89096BA0F489D1CA60C57AD907BFC8E97F4E057B6D"  
03486 "709414FB0A302D30357FB4635F70BB6A32CFD842DCD8D9C9D45FE082B3746951A862EA870D9"  
03487 "03382C138425DF93A505120D93FABB8F523C1D3946B85425FB338CA7DC4B2FB6512F0C8A6"  
03488 "7FA470A16284EAA943E1A9C0607A02D27F55F1DF2EB6090F94B64076FC2D3D3B3694DA5C7E"  
03489 "B2A180DEA14AEC21156E1110DF75616685FD53C72252FB87E7D19EF5AE8D9E129D0984A065"  
03490 "20C789DE22CBF6E2271801691E0C3CC672F6865A41559910D0279AB9E0112E66B2A1C2B22B"  
03491 "6679CF70E3FF870084562ADB36F532E64D44C7264E44D488076F7714A1089526118D4655FF"  
03492 "E16802D803AC2601493CEC3AB27878CE95ED3F321913217DE12B8E5E8FB75A85707EAF6F1F"  
03493 "F08CC86B91632ABC7CDC42D1C5D0F2F49D5F412B4AD0C5C46CFA74643C9333CF3558DC3AB"  
03494 "DBCBE23A573146D648D540116136F3F29E42FF07E26C1504E47FCAD1FDAE63357E421E464"  
03495 "40424921404ED02FD4BFA2FDE68CBE6CE49E4E120C141013DF5C0BDF776EE36FD5899CFC0"  
03496 "57DAE2C9DAEF6D37BF8E85258BB36B54ED8374BDD49B6AC2C8BF3105A194F76DD512336EA"  
03497 "AFC7BD2054AF6A9D606517DE03AF445CCA5FA65307D2E116E42A3676EB6033AA17D76A87F52"  
03498 "D144CE25E48DAFD3E044289C4600BFAA9CCD963D14A19C5911C3DE649439440D11A21154E"  
03499 "F25649F7DA295E0FDAE8C48351BC005C011A10D021B3062492A7CC933AB5D854B26232B709"  
03500 "1CDOB7AA3135F28E3AE75E267C223C5E03B60FAC1BF78123C5AF76719191CB6BA277A5BE81"  
03501 "E64117AB344D92837B6D600F36702F4BB4532C5EB1C9BAD8A111C540F52A225DACFAB37641"  
03502 "898B1FF770A523F8BC8BB0DADB59235DE055810396F1993539A9CAAB622389607DCDFDD51F"  
03503 "B67CBE89F0F868DB4D27A7A38542A076D158919DCB621BD325F23AA0DF694C44206FD4219"  
03504 "2FE7F9A05743CD548DF111676AC35A3230E372A5A6D7E213C4584EEA1A4993067FD28DF669"  
03505 "0BE9D3E9A4C06BCF89BF1AA47496F8F6A18524187BE80D59A4E80193CFD757B706AD483A91"  
03506 "6AFBE2A56E0A69F3B1BBF9F4B239D05C6C556A8D2B00E9BB1FDA6620D949ECDE86EAA299B"  
03507 "E93A7884C99FA782F2BB3BEEE86046489B3B8A5930", // PRN 48  
03508 "71BB1B2E833793D854F8A9A81E6A6947057B9571F2BA99380DD25D878D6B48F09ED7DBFAC"  
03509 "E92B6B82F413E038128F6128AF68C467E9A4DF2861DAAC674B6D948A10F28F7D43657FEE26"  
03510 "577AF438A2F4422186930702EBC6C9173E661D59CE7594DF95B861F9D12EB060FCD3BA4315"  
03511 "9C9A1BDC1EF13E04893E411267331588CF4831978469FF569C1A738C54001BB5CF4FABD289"  
03512 "075A165EDE0A58F6CF6D215D306A7840CBEC0E87E3AD186F7A67A967373551E13D2956E5C"  
03513 "578A7F5BD50E2D570F9B914848D46A640913EBED2E2ABFB86916BC34EEB3E8A671AD771F6D"  
03514 "3780B6FEC143E26F53B02977255314BAE9A2CD9E5BA2B49C73226FDC724A859F8BAD3A9FAC"  
03515 "A1B0E5F2DFF7E1E45DB6D4BE2B76535A817F94594A4541C01BB62AA83A690B6D84FC13B632"  
03516 "972C61D940F2C9D32837413AF8E42045ECA3072CD044B2183400CE63C418879D13FD281A8D"  
03517 "0835256DD2BC3C9750C1D44CF1037FD7264B7716398FCF1D31CFFD0B7C52C6370E4CE6FC1"  
03518 "63BA40436490A757465B20B8890C3B5C0AFB971ABFD01796569E3BC73C13D1E4B1FBC1CDCE5"  
03519 "9D21B6B11027E5770C589603FE67779A49AD0EC66910A2BF4D8C8ECC18A32EF92F502126A"

```
03520 "5DC3DE618233B9914A9608B2F17E5161115D9A3BEF1D5701A9D465A1437DE24371C9179800"
03521 "CB5728A7F3D734A2A706BC64D356BDF591389970B6CC139AC510A98E3C75F20120450CE453"
03522 "73AAEA6A279BB17221BCD32ECF82C11B4C1CEE0A44792CF56978D3D2399F7ECDE9F8D9217"
03523 "F8BA22770E210D0F1AA852178B872E296762873765A73DEC08873C04ED69C995C5751B97DE"
03524 "C23B94CA674FE3F66211317B074D8203EC530A20B6E6DD21AB55895BEE1CDB0876183D652F"
03525 "4A0D2EFC95749F8F192F860BEA534598FD709B396C209CEAE4D9190980733E7E98C8ABE52A"
03526 "53C68D86053B56BA6FCAB5C827292D729CB8BFD1FC8CAFCDE15E4527B604018F28AA16C1E9"
03527 "13F55461AF87C9A7BE1A742002E52B3A14EC30B259DDE7BB892CEAF77D25B7670B339B3348"
03528 "78F697C00CE6740117AE7C67DF3F8A7BEAC89D4872682C47F368F835AFD7ECF0D8471AD014"
03529 "68B7BBB0A974EC469A8F79ECF8D379DC13685D2A8F6F19CE102C3DE34B11422AFB42E894C8"
03530 "D00F606296DABA7123FCE039ED27324D60E853BA94DC638454088281335D437A954333FF1A"
03531 "8D08E2A4D25CB3BA0D08BF6625E25EAD3C1EBFA2666AA49550578D3763ECDCE81303B53F18"
03532 "B00C8900AF4E0532CEC94513DD9F50BE511CFE4D3DDBB3F112AE148DB062B2EADDB901C"
03533 "DA6AB6BF59D37F356AB34AF97D3DAAAA417642E87C9B95AA546C682ED641214605F82A4F48"
03534 "6C9C72576106F76D7152615EC8E77187D4485071CFC6B0AE44880442790696E057A3AE20C8"
03535 "60691353B3F6BEC5F1C2DA07563B423BC01E0334099571158A432441256D7C409B7B6EF264"
03536 "42075ED17E2BE37F8EBC049CFBE0FA89CDA7A58DD32C417B34E899FBE86E2FAB8D30846DA1"
03537 "7144A6A6AAE1C24FCFABDA5B573FD2D6337226B5E49BB031B4D2B4556BDB871076F67AC03"
03538 "C3A73CEC01BD0B1EC42ABD177127E62A66FE8E475B982B4490F0877466EEFC7317A703C5C0"
03539 "7937340ED4B53E5DE5325197FA31B8C8E05AA2222064EE5D7C06D4A1EB53151F75C94A2E25"
03540 "9688CA0716548465C255D81FF10BACC2C13098ED8CF7F5B15193EE14FB5D258E95EDCC93"
03541 "E9796FA823892C705A5771D561787C12592D269D657FBB71F021F365B7453D50C35F748FB2"
03542 "B7F36DF28769B81EF12A26A237FB0239C173559540", // PRN 49
03543 "53DA0E7B84741AA9E225483630169ACBCC03EB8CDA28B7BDA685C756D66B14488A2D0AEF7E"
03544 "6CB2D80F2726327257B7284B93EA1B56AB80FAE668C04FA49FCD658D896A997685E1EDB4DD"
03545 "F85456B37F32FC8CDE50882EF0F09BE4ED4AB9A425806A49E8347A42A50B38FA8D1DA2FD2C"
03546 "9438618B6701DDE159060C186D50170F24F38E07B185E327EAEA4A0A7CA41A69C69E9D95E"
03547 "271287D3AB8284146A58440EA131A7F47D73CB2BCF40FE3A58E1B998C2E5EF9CEEC8EA2F84"
03548 "67BB7757C03A99FC8F014EE933A7080CD46625A2A7A251B7A37E4208956A8C9BD35E6B8674"
03549 "BC06FCA5DD04A2558C9665C7985014D3AD95ED256FAFA358962EF5BB26AE2FCE899392DF8"
03550 "58F99303E2417BCA7672E991FEB891F5DFCA2D461148367C5C0DE1460BF557194533DAF01A"
03551 "5E8E0E43D57B825AF7EEFF163DAA23B9F95C063A26B3D213459D885AA96023715CD21DFA2A"
03552 "2250F7610B78A77123443BD06EFB7DC85D1F16D0019D2937C3DEC4DE6389485ABB21642B6E"
03553 "41ADD43CE96F228C08DF6288A647EC2FE96032B6DCD651FF950B72964EE08FC2030272E3F6"
03554 "01DB7F7E770E655389CA6CFA2F9B87CE76FB0E0CDCA4EEE5E80FC756BE46CC09F84BDB34AD"
03555 "A2AF0C024ABDE0066ED939F8EBC236CB3F577C1BFD741F9D101A038EC86AB0A85462BAFB2E4"
03556 "84D6722499A6310FA449D979030B2A21206D44225800BE2228FA0A0AE6D92C8DA652E1B003B"
03557 "D2734D305572B735CC2A591E090394DB791245C22B4D29E706476593B6F90C694C5B87BBB0F"
03558 "A2C479E292A768A9687A713336A21D1199186F852C41F586E9BB64004D8BB6814BFE73983"
03559 "4C99923177AAF87B926D56A7AFC0879C027332A60951C84E9314380A5A78E1196D094F15D8"
03560 "56AA36742825D2B397156BCAD8ABE7291FB41DB4365AAE49CA82CA066D3B4366D3122ABBC0"
03561 "0F05559DAFBE9A9F98361DDAEF068D60B18265E7184C4D6BC9C3619CFF5C758090FF6398CC"
03562 "EB78176D2A8A24B9854C4ACF5CB614DC1CA0E15E7E85442241D48FD3D6E851A5D3947FA76"
03563 "9560928948FA26FA16EFBD2159994BD92B3D6B0C62818C91D4724413A7F40B2A2D67F4FC97"
03564 "B5DF6A7E3C8C03158E201D6643F402D3DD6995A42900D46C2881198CAD28A27489F5116ECE"
03565 "C3E38D999B2020E0C381DB3B8230811270D75950D9BB61548802DBBCB68ED8C7BCCB50D606"
03566 "BE400BECF873498621E66ABD2AA179B3E90E055C3719CE2FE047F815B95B065BA086B467AF"
03567 "4124E276F8CEAD000BCA5499D36217B250009A7B43E81CB3F8B1A3238EE436FE61F2F94279"
03568 "GDBCBE570BB4FC78B35C3CA31BDD432B3AD75B08107253E8F910EFE0D0B5453A8A055D88"
03569 "4892278688B3ECA612452B590AF38DBDD9A7070C5610E7A3CA6C91D24438E7F45E7A2A330F"
03570 "164AEDFFF1789D5E875EEF121298DB79C77278ABFEC3FE3DF843C46F40E847272EB2669BAB"
03571 "A808C3E3E1F13516D5066AF4DFCDE6EDB2FF0B0A4CE9FA9B4101F6F144B02384868617CD39"
03572 "175852E065473D6F566CD18D7403FFD24DD33ADDB52C7CC22167E49102C46DC369A92CE2D2"
03573 "FCB81BAD1F14B7C2F80A65D8FBD20FDA23219873ACB8CF934E68D6F8FED6B41193CFAB1F"
03574 "44CE4BFC7C67DB1E8804B47DFD7E8AE281E19846AEB6FF94AE7E7CF6FFAB46242843811E6C"
03575 "5BDB78157C67DF492FD3653D7FA5978316EB055059C6A2B6306C957418860A88F63355E76"
03576 "D96F4712D7128D9B3EB98501AF5B093F2C314F98EA2CDB89468E1B138CB2F25EB911C26B9"
03577 "7DCCA47F1A1D6C1CD415A5079A756B8A8715DD3164", // PRN 50
03578 };
03579
03580 // Galileo E5a-I secondary code
03581 constexpr char GALILEO_E5A_I_SECONDARY_CODE[] = "10000100001011101001";
03582
03583 // Galileo E5a-Q secondary codes
03584 constexpr size_t GALILEO_E5A_Q_SECONDARY_CODE_STR_LENGTH = 100;
03585 constexpr char GALILEO_E5A_Q_SECONDARY_CODE[GALILEO_E5A_NUMBER_OF_CODES][101] = {
03586
03587 "100000111110110111011010011101100011101101110000101010100000100011111011100011001001101100011100",
03588
03589 "01100110010101011000010111010001110011100000100011011110010010111010000010100100101",
03590
03591 "0101100110100000001001010101011100000101011100000100101000110110111011100110101000001110000001",
03592
03593 "1101001110100011001001100100000001110000010111011101100011000110010011011110110101010010110111",
03594
03595 "1011100100011111100101011040111011101100000100001000011000111101001011001001101001000101010010011",
03596
03597 "10111010110001101111101001001100111010011011110010001010000001110000100100111101111110011000",
03598
03599 "01010011011101111000010110111100010100000010010010011110001101011010110001011101001100110110110",
03600
03601 "11101111110010101011010010110100101111001110001010001000100010010101111100010",
03602
03603 "011110011111100010010101110100000110000100011010111101010010101100001001011110111110010011011",
03604
03605 "1100101001010001011100001111110101000111010100000100001110110001100010010010010100",
```

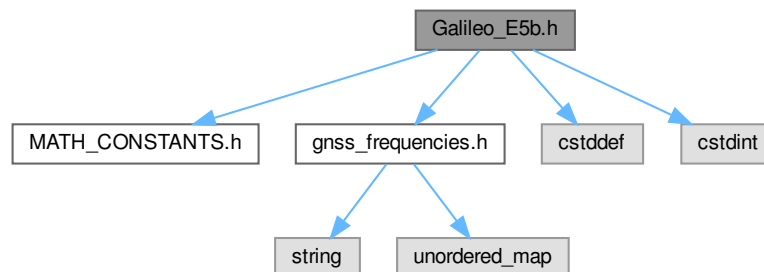


```
"00011111100001100100100000100000110010100101010001011000100100110111011000010001011110010101100111",
03597 "111111100000101010011010011110101111101101011000100010011100100001011001011100101011101001001100001",
03598 "101100000011000001100010110111000010101101110001100110010101110101011010110000101101111011011110",
03599 "111101101100001110011000100110010011111010110011000111000101101111101000010001101011101001111010101",
03600 "0001101110110010111110111000101101011011111100100100001110010101110000101110111100111100010110100001",
03601 "001011111001001000000110100001111101001000111000110011000111000001000110111011101101010111111001001",
03602 "0011010000010110001110001000011011111000100111011010111111001010101001001011011111101101110111000",
03603 "01100110101010000111001011001110010001111000001100111111101100101101111111010101100010010110101101",
03604 "1001100111010101101001110000000101100010110010010010000010100100101110111001110111100001110010101000",
03605 "1000000111010111000110111101011011100000011010011010011110101100110010111110110111000110011011001010",
03606 "10100110010101000101001001000000011101001010100111100110011110000000110110111001111010011111011000110",
03607 "11000011001110010110101000010000000110111110110101111101100010001111001111110001011011101100110111",
03608 "1100001111010100101010110010000100011101111100110110111100100001000100011111001000010100000111001101",
03609 "0011110111111111001001011110101011100111011000010111001110010010011001011010111100010100010111000001",
03610 "100110010100100100001001111000000111010101111010111000011001101111000111000100100010000001010110101",
03611 "1011100100111000010100110101010100100010110100010001100111110100000011000010010111111101101011101100",
03612 "110001110001101010110101010010111000000010010010001010100110111000000100110101100111001000010110111",
03613 "00001100110110111000110010011110011110110101001111110101011111010110110000101000000101100101111011",
03614 "0110000111000101111110100010010100101111000110101111100000010001010001000111011001100100101001111",
03615 "0110001001100000001001110111011110001111110100111100011010111011010010111010101001111010010110011101",
03616 "111001110100010101000001001011111111010100111101111010111101000000111110001110010011010011000110011",
03617 "001101011001001010101100000010000011111100110001011101011111010011100100100011000111001000010011000",
03618 "010100100010100001001101100101000001110000111101110010101111001001110010000111011101101100011111101",
03619 "01110011101100111101100011110000101011010101011101111101001111111010000001010011101101100010010000",
03620 "1001010010111111000101101100100000111011110101110100011000101111011001001100011100000001010000010",
03621 "1010100011000011110111100001101011000110011010000000100010011011000010110100010110110011111001",
03622 "001000101101011011100010101001110110100011100101111100110101111111111001000111000000001011110010110",
03623 "0010010100110001000010100000011001100111010111101011001001110001111100101010000010011110101000011101",
03624 "10011111011110011001001111000110001000011101010010111110110010000001101000000101001101011100000011",
03625 "1101011000101001100110011110101011001111000111001001100100001000001111000000101101001010010000010111",
03626 "11110110011001011010011111101010010001000001101110101010010011101010000011010000001110001100",
03627 "01000110111100111101001100000100001111110010010011001101111010101111010111101111001010101000011",
03628 "11100010111000111110100000100101000110000101101011110110010110110011101111110010100110010100011010",
03629 "1110010101001000001000110001101010000010111110011010000000011010000110011101101011110000110110010",
03630 "0010011001011100011111110010000101000010110111101001001111011011110001010101001110000011011001000",
03631 "0011011001001010001110101001111010110000111100000100100000011101101000000001100110011110101111101010",
03632 "100110000001000010100111101010001001100010010110000100100110001110100000111101110100100111110101010",
03633 };
03634
03635 /** \} */
03636 /** \} */
03637 /** \} */
03638 #endif // GNSS_SDR_GALILEO_E5A_H
```

## 11.703 Galileo\_E5b.h File Reference

Defines system parameters for Galileo E5b signal and NAV data.

```
#include "MATH_CONSTANTS.h"
#include "gnss_frequencies.h"
#include <cstddef>
#include <stdint>
Include dependency graph for Galileo_E5b.h:
```



## Variables

- constexpr double [GALILEO\\_E5B\\_FREQ\\_HZ](#) = [FREQ7](#)  
*Galileo E5b carrier frequency [Hz].*
- constexpr double [GALILEO\\_E5B\\_CODE\\_CHIP\\_RATE\\_CPS](#) = 1.023e7  
*Galileo E5b code rate [chips/s].*
- constexpr double [GALILEO\\_E5B\\_I\\_TIERED\\_CODE\\_PERIOD\\_S](#) = 0.004  
*Galileo E5b-I tiered code period [s].*
- constexpr double [GALILEO\\_E5B\\_Q\\_TIERED\\_CODE\\_PERIOD\\_S](#) = 0.100  
*Galileo E5b-Q tiered code period [s].*
- constexpr double [GALILEO\\_E5B\\_CODE\\_PERIOD\\_S](#) = 0.001  
*Galileo E5b primary code period [s].*
- constexpr int32\_t [GALILEO\\_E5B\\_CODE\\_PERIOD\\_MS](#) = 1  
*Galileo E5b primary code period [ms].*
- constexpr int32\_t [GALILEO\\_E5B\\_CODE\\_LENGTH\\_CHIPS](#) = 10230  
*Galileo E5b primary code length [chips].*
- constexpr int32\_t [GALILEO\\_E5B\\_I\\_SECONDARY\\_CODE\\_LENGTH](#) = 4  
*Galileo E5b-I secondary code length [chips].*
- constexpr int32\_t [GALILEO\\_E5B\\_Q\\_SECONDARY\\_CODE\\_LENGTH](#) = 100  
*Galileo E5b-Q secondary code length [chips].*
- constexpr int32\_t [GALILEO\\_E5B\\_SYMBOL\\_RATE\\_BPS](#) = 250  
*Galileo E5b symbol rate [bits/second].*
- constexpr int32\_t [GALILEO\\_E5B\\_NUMBER\\_OF\\_CODES](#) = 50
- constexpr int32\_t [GALILEO\\_E5B\\_HISTORY\\_DEEP](#) = 100
- constexpr uint32\_t [GALILEO\\_E5B\\_OPT\\_ACQ\\_FS\\_SPS](#) = 10000000  
*Sampling frequency that maximizes the acquisition SNR while using a non-multiple of chip rate.*
- constexpr char [GALILEO\\_E5B\\_I\\_SECONDARY\\_CODE](#) [5] = "1110"
- constexpr size\_t [GALILEO\\_E5B\\_I\\_PRIMARY\\_CODE\\_STR\\_LENGTH](#) = 2558
- constexpr char [GALILEO\\_E5B\\_I\\_PRIMARY\\_CODE](#) [GALILEO\_E5B\_NUMBER\_OF\_CODES][2559]
- constexpr size\_t [GALILEO\\_E5B\\_Q\\_PRIMARY\\_CODE\\_STR\\_LENGTH](#) = 2558
- constexpr char [GALILEO\\_E5B\\_Q\\_PRIMARY\\_CODE](#) [GALILEO\_E5B\_NUMBER\_OF\_CODES][2559]
- constexpr size\_t [GALILEO\\_E5B\\_Q\\_SECONDARY\\_CODE\\_STR\\_LENGTH](#) = 100
- constexpr char [GALILEO\\_E5B\\_Q\\_SECONDARY\\_CODE](#) [GALILEO\_E5B\_NUMBER\_OF\_CODES][101]

### 11.703.1 Detailed Description

Defines system parameters for Galileo E5b signal and NAV data.

#### Author

Piyush Gupta, 2020. [piyush04111999@gmail.com](mailto:piyush04111999@gmail.com)

#### Note

Code added as part of GSoC 2020 program.

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [Galileo\\_E5b.h](#).

---

## 11.704 Galileo\_E5b.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file Galileo_E5b.h
00003  * \brief Defines system parameters for Galileo E5b signal and NAV data
00004  * \author Piyush Gupta, 2020. piyush04111999@gmail.com
00005  * \note Code added as part of GSoC 2020 program.
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_GALILEO_E5B_H
00019 #define GNSS_SDR_GALILEO_E5B_H
00020
00021 #include "MATH_CONSTANTS.h"
00022 #include "gnss_frequencies.h"
00023 #include <stdint.h>
00024 #include <stdint.h>
00025
00026 /** \addtogroup Core
00027  * \{ */
00028 /** \addtogroup System_Parameters
00029  * \{ */
00030
00031
00032 // Carrier and code frequencies
00033 constexpr double GALILEO_E5B_FREQ_HZ = FREQ7;          //!< Galileo E5b carrier frequency [Hz]
00034 constexpr double GALILEO_E5B_CODE_CHIP_RATE_CPS = 1.023e7;  //!< Galileo E5b code rate [chips/s]
00035 constexpr double GALILEO_E5B_I_TIERED_CODE_PERIOD_S = 0.004;  //!< Galileo E5b-I tiered code period
00036                                     [s]
00037 constexpr double GALILEO_E5B_Q_TIERED_CODE_PERIOD_S = 0.100;  //!< Galileo E5b-Q tiered code period
00038                                     [s]
00037 constexpr double GALILEO_E5B_CODE_PERIOD_S = 0.001;          //!< Galileo E5b primary code period [s]
00038 constexpr int32_t GALILEO_E5B_CODE_PERIOD_MS = 1;             //!< Galileo E5b primary code period
00039                                     [ms]
00039 constexpr int32_t GALILEO_E5B_CODE_LENGTH_CHIPS = 10230;      //!< Galileo E5b primary code length
00040                                     [chips]
00040 constexpr int32_t GALILEO_E5B_I_SECONDARY_CODE_LENGTH = 4;     //!< Galileo E5b-I secondary code length
00041                                     [chips]
00041 constexpr int32_t GALILEO_E5B_Q_SECONDARY_CODE_LENGTH = 100;   //!< Galileo E5b-Q secondary code length
00042                                     [chips]
00042 constexpr int32_t GALILEO_E5B_SYMBOL_RATE_BPS = 250;          //!< Galileo E5b symbol rate
00043                                     [bits/second]
00043 constexpr int32_t GALILEO_E5B_NUMBER_OF_CODES = 50;
00044
00045
00046 // OBSERVABLE HISTORY DEEP FOR INTERPOLATION
00047 constexpr int32_t GALILEO_E5B_HISTORY_DEEP = 100;
00048
00049 // optimum parameters
00050 constexpr uint32_t GALILEO_E5B_OPT_ACQ_FS_SPS = 10000000;      //!< Sampling frequency that maximizes the
00051                                     acquisition SNR while using a non-multiple of chip rate
00052
00052 // Galileo E5b-I secondary code
00053 constexpr char GALILEO_E5B_I_SECONDARY_CODE[5] = "1110";

```

```
00054
00055 // Galileo E5b-I primary codes
00056 constexpr size_t GALILEO_E5B_I_PRIMARY_CODE_STR_LENGTH = 2558;
00057 constexpr char GALILEO_E5B_I_PRIMARY_CODE[GALILEO_E5B_NUMBER_OF_CODES][2559] = {
00058     "C5BEA11BAD79BC7A91DC8DBED9A215B8FFFE4978A981C98CDEECE756A364B33FF851B7B3A"
00059     "FDEA0564CF9DA207C784E13542404E3BDE45C1DD70B4DB6B51C88E3ED42A1C524E0BA5441A9"
00060     "C8766013E712313070BDFAD3699A226B719410EF18EDC843818877FEF748A0D7C6AA366163"
00061     "30A087CB91058D7F1A245311AC571864DF734F99E3E643AB7627B4CB57DADC23E4D22C5F2E"
00062     "3D2FBD4F04DF616E8110A001565B030C993141FE038670F574DCC4DF46C8DC18C84B449106"
00063     "9D1CAB4BB923AE6CD41BB9528ED40F7D139CD425416EB2D6342AB8808009A40C042F153DF2"
00064     "3C8B14F3F7235FAEBC9AFE0C845AD41F40E2FF7A7FB82971E014C9269C60AFE05F694AEFE3"
00065     "B23194956253AC39AD42E0B7575F96CF2A094011C91A2FD3A10C02CB42D3425A3D6D85B404"
00066     "CDE1171ADCB1B5EE752F5783DFA14A7E26507C231FA5541FD06330ED44E37DDA40008A91BA"
00067     "74EFACBBB617C7EDFA3DDE592EA284381EF17477038A8B7ABF3C5BD673D658D11A22B97DA9"
00068     "F247FECF2143D69468CF8A112A69FD3E9488F14AB1506DB1240F26C52D2F19AE7A81A39E9B"
00069     "9304F65CB7FB3D9B4404679701216BDEF9F9E576EB36F7C9D77847A8FC40A7D7C4C65C8D78"
00070     "BD82FC7E5F90C6BE1CF19A2B7576DB1547A26C24E5F9BE97D2AED0A71BDEF74B26F92BOCAA"
00071     "A683DD3412BC52C7BD443F32EB3E39B2C9B1CD6B274C825AC7618CADE08D9991E5150281E2"
00072     "D9076EA58B2A99C54236A1E93F4D467692766080113C58FC53093C739FA8B0CB85DD0B6683"
00073     "CCDFC0907E124E79C33C3EAC7CDF8E9F6779B87B4FA71688BADC1DEE44FCE4952CAFEB0D3D"
00074     "64326F07DE3B6F734D18A24233A39C0B59A1F458B9F28B958539193A054DA2AC02B60D1146A"
00075     "1C6D5B86DE1BC424C2C7C703CAF92997E09A345AC0243EEB72468B161E5CE268DFE2A76B10"
00076     "F84DEC801F5F20419D9C326E7AB0BA22BD4225CDD990DE4A0C7E8C60E24A385DEC7F342E8E"
00077     "5BDA079ECA35F7DEE07A2E1690CBB8C28247B6E13AB795D6DE97FA074596DEE8B5846E8904"
00078     "39B61BD71C228D10C7BC45886288F33654F787B4D336950EE6CB2989EF35D14200B9082C08"
00079     "5C6EFCAD497A12A053543DEE352E4AAEC83858967441F7023B90E263E8476D53E23FCA4805"
00080     "99FF7FA69C006271E1353CC546E04FDA1D6E2EE8DDA422F25216522B9657FD3D02CBA10247"
00081     "98C00CC2396154E2DEEEF4EE6E25C8980C2B78991C545C09AAD403A0D3D16FFB8C485C15"
00082     "D93A160AB2D8296B811A25030688196C7208D66269EAC5C17CC4F00BB35CF85BC73DEDFE1"
00083     "CE6C6547DE6C7873135A351EAC4335C257694D766CFE21440BE04432BE32583A166569FA7C5"
00084     "ECAODF8D478ECD43F44D914A707902234E9A9E5C68FC472C8DE48C5E96071A1999108D6F98"
00085     "00F8D1141EE18F376C9A422A73C21535D055080845586C931FE8600366FBE3C80522D32EAC"
00086     "06E9005B545F59373C64AA152226F66E100B183690347C53698ADC5B8FEDA8AAB5D36823A1"
00087     "F3775B295EA8FD2EB09C23B80A3E96AC1580124E027004CB0A488FB2DDCD739E8BAF4B47B7"
00088     "539D2D7E3E79FB67255F5CEDBA801026A6DF3389A205DCDE967AE8F0BC766A53465E46AB20"
00089     "A67B379B11546D66E385B963CEC31C4B832C62F417122CE3D158DCB5D042A7290371BA414E"
00090     "9B59FE6379BDEE02FE476C9194F27E67B6953CDF9FE738A7B8256B2F8789D5679A5A172C46"
00091     "971176955E98BE1FAEABEF94D2B7E08ACDD9D9FA341A47D211123C6DFB12330142B8171DDF"
00092     "404225E20A9DC94DC00B1627B704BFBF6264A74C", // PRN 01
00093     "F462483A58B2AA45A7F0B344DFA1C95568646AE79869337332CF0FB02D7A9A91633F160EC7"
00094     "D50EB74D2FBD3050B16E2E264EFC804070A7845AB5903A986183DF0F7146DE5BC923841446"
00095     "46BDB346E3A6436FFB19E16C324B001C4CA6D4B223874D15032F743481B2E4D8B0D77B23"
00096     "BBEDCD2ADC458C23180C1632887694F12949B95584ADF4040F21E2F47F1F1F5E4E7FC3AD65"
00097     "21F1D8CAD7FEC9EB2BC1CF3C04AB70E54E55A31F654634A1D29005CDF05797D3E7FD5B2B94"
00098     "398BCEFF5A6D280BA49C79066A5866FABFF8D59582FA407E795189B3AFC33B798F672FD41E"
00099     "6806D877ED31A4614627F6D46011FC6153A9F3143018BCEE9A0F98F748194AE7C51539C175"
00100     "B66C5AFE7C9BCCF8D6B4926122EFEE9AEE07CECAAFF783BD40AD8376A95A0345A3EF7DC8F8"
00101     "6A1F3E5BFD6F509AFF85DE7C7101CF51766E6B4419424A10F15BF4343BEB845E5F6279FF54"
00102     "4577F26DAAB7522580AC9EC833D3021AF903BAEA8CC53B5F0A630639789CDA3CA6FC72F96E"
00103     "F17EE975366610DBE3DA5063955CA0DB7FD3318E14FAAA514CBDC864F40261D54EC284F510"
00104     "6A1F385409C8DB91D535B470A883B0D82167435E2533D123A7B71FFC8914A370923B121"
00105     "F45341C2D65BC868570185F247E93E77FAE64BBB112A2E7B63F2117214B9BD0B2D9029AEA"
00106     "EA8F6E6A70F18D178B0B4D26AF7F5FDC8CF149B6744EF28F77B7D87DCC92D0E2CEF3977D38"
00107     "3E8920F71D89846CA5DE3F4FD76D51A0B5B69A6589774EC227CEC64B16217EB55B37EFF5CA"
00108     "78B5DFBAAD361DE5E606627EBA04F5FADBA5224A6C3BF63289E9874A56180B5C57DC188A62"
00109     "E1BC314488F06E728399066FB9EBDEBFCB2F20466CB0C6EE1B2B8B158B769C40409838D95E"
00110     "C5EB52E169DDFDC0F946700852FA1F10436E2C093E257DC969C5D3195CC4DA6E376CB1A915"
00111     "A92519EE1EB2034E3C55AF3C24C247D74832D46D532F9797268C3B0DAABD705945F4EC5514"
00112     "19E94E84A13E75AC428B2559C2076333DED3A6E9CB604751E2FDF379B9A1BD6099274E90B5"
00113     "23A19010478B0958A5C53A2E9A18CEC53589AFCFB075387425FB3A24C2F66B40E55DC3F52F"
00114     "C4048AC77730953DE72BE2E662AB41EEAAC4A661A6E43C65AD368FC2AEBB7F658C2D8A46C9"
00115     "8234071009ABB81C7E33A75B86D89BB3484391D210582A9407637D82A9DEA9E2BEC0B4CF9"
00116     "CFDFA6E0B7D377E2EF3E91212D212B2F9515E6B541046EE2E5744887EDC1A0B149F848E7D4"
00117     "EF9CD5AAE0010132C6C5A8A5E1082CDD46EF100C82AB553EBE5490184971B1F08D110156E"
00118     "0BCD083B81F94D7F19C0F41C815333C34384F16E9B34D659E6A65CB9B274725F9C535420A"
00119     "A730CF0B1497C96EF000D5227834F7C8124B114F9711A0AB19D7A56367902093D5B8DF349A"
00120     "F5AC30E87F737DFDDACB3E95981837F7C48F97F3E34BF278616C5A14180D81B3BAA654206"
00121     "42B74606D76C67F6897BCB2A12EA2A07F0DAD3DD1DEC00E265C0D965EA6CFBBD5E3721F4CEAC"
00122     "C09B3B04E423873AC76E64B17C071CE9A6E336676835E854D1E0E2B2618D918E41808FF0D"
00123     "B2296CE151A2883AA314FBF2E9A300A75C841AFCB3EC84DB34F392FD4DD970F5ECBEF9E81D"
00124     "9DCBA9EF78AF41CDFBD9D77B6EDC92795175D805137996CAB4F91FC8935EFCDD788AE1F5F"
00125     "C650D4840EF222A78401DDD313EA4B4943A89DCE637F1F81636D06C6E1CC1062162320AA07"
00126     "D01F679F57384820286F29CBE9881FBECC3E1AE4D7FAC3A143046E29D9D4A19B95BA2508C7"
00127     "030CE0846E358D52284A921ECA6F2C7890696F6008", // PRN 02
00128     "FD5488290C9EF7CC7C4E4F2A6900C2F9CB24239B239645B27D990BA8878505E1987F8DF3A7"
00129     "1105499EED4410AED6FDF8434EAEF0C2C0FCBC1C2508710CE37EBF740B273BDFE45F4B28625"
00130     "9021B2182D6185078F0C7A6580B1371F023F07236F1655C9BE30F35F2485C9EEDBF03068C"
00131     "660A9E754C0B6F949CF5556C133AD512C65DD3E4AF6A01AC9F4B11F107FAD356FE7C4E0E8"
00132     "814527E486226D9A6FD2D9D0F1B7B1A0F840F99C83F8483E3DBA2F1B576634BD5ECC49640A"
00133     "EC7338185D72629A03F12CF53EB3DE8FA86112F5D0E666DE0DC3CC772693897690EA298B"
00134     "2B9F02C213FD46ABA0A2BA2A2D90CE15BF1E6069347D5A1E78BC55D13E2BE2D173003FF68"
00135     "317BFFC96A9BF858A07A4D2EC5CAD1D4FB21CDE30D67B1DE383096987B93F8C709E3033772"
00136     "32EE269234D48C58E35B3E9978B13B8579B0E759B4CA76B6E69CFDC350C1FD02F7088D0BF"
00137     "BA5633A8EDAB83871C1693BDCEFB6A8D0E39A78DBB3D190196FBAF7033238792139561543"
00138     "8C75D59705810BA4CA380367AD26B08B85A19D167B0ECB27EB8600D9F6FB96375684ABA1E"
00139     "FE9460DECA0690FEA707B2410781C571ECCF567569E07B78311E5AFD6A3EABBCFB6E68068E"
00140     "365517FDFE9E1A15FDAC6A000C324D82FA7D451334B873888B6DD164C89CFF3253BFC2DEBE"
```

00141 "4CC3AF78C66EC6DCA4BD529C3DDBC8B932120733133E754F235354144B3EB4E37E30038F90"  
00142 "73F211C432431B192E9605A63B39DC6C31AC889E468DF3CAFF2A8D63683235B935784F6055"  
00143 "02A8E63794FFBE27C7A11502C99A834B0BBEA2AAD4300405F9C373203DC05D28CBFF5260C"  
00144 "001DB0F963792F6D1D1BFE1345A4084A8A0FEA3183A6C67E4852B90A426A257279EDFC751"  
00145 "E4D3963816D5044B16248366BCCB874E9572DC9CF65FEC89F8D8A29F7E2328A0AAE3FE1930"  
00146 "1C7A440115F881A3808B64B8E3C17EC0F225F1AB1CA4824D54330E504F562B9981511392F4"  
00147 "715BA02578130DD776C8023249FE78FDD66D92569D9D5E652EBBD89A5C55C6939BF99A3441"  
00148 "E6E3ACC944033FC5560A4512C3EA1F78890064C1663EC86BFCC9626C30CE67487A0928E95F"  
00149 "A0957B774DA5F9A6730BCB7022E74D5DF4A6EBBE76D27317B972CEA4C0FE3FE1517E88361C"  
00150 "32E253C025B4F16F55D532A75C3F15FD1AD532CFB9BFC20AA18B5D957FF969185B55405AC7"  
00151 "BB185CD8F3E5AB99C31D475B264954130BFC6318C437E5B437109818AFEEC7C2B78DBD7076E"  
00152 "DCA94F0EF2E5EEFBDB65B22BE8DA21CE7B83A018E78A671A7669B1F8EC62B7AC522886AD8"  
00153 "4377108E5E5FBC79A64E68CFCB535538F892F0278CCD295F266FD1C84B33DAA3B340128"  
00154 "BCBFA8BE0BE4AEE9E88344E5861E14747AB0F9DA7EBBC32CCF2424CB8CFC50627532607D47"  
00155 "DF5E878537307FDB54276B92C3C3B0366623F2777C7196E3B6D879A5E5A7A70D3C1CA7C583A"  
00156 "4935150B37A8A562BAAA8C5B5B30107B1D887499A4629B1A6621F5D79F9AC3764C42B763D6"  
00157 "03AB28A9C921827149922D9230581BB8F05079E1EB37215A45498B20667A25E613CAAD346A"  
00158 "248F01401775E9A0CCD06BBFC25B8CBDFEA11B126DE738240AA33AA0B248C5D4AD67305E07B"  
00159 "C8BE86464726151247DF7D35D08B8DD373FB6A26134846B4DB32DCBF3053E5001A50B625E1"  
00160 "2A8E480043E053793675D68D2CED8AE3FA434DC701CB8B9B6C666BAADF5E1E70CD94B308"  
00161 "2C36619E689ACBA9CCDE701664DBED3B069242EA13EFCFFE6084846C096D03E59C7451E2"  
00162 "23622864163749D6B7D938FB0524B057AC679D3BA4", // PRN 03  
00163 "86277BAE531B790D4F98842B2F52D1DB87A7F18529C775D7B06A93D731CDBFF92DA0214527"  
00164 "7AFDA053D99D96A7B14A3E7AA5657175CB63C2F4D724E4D66415F60390683EA64AB62ABE23"  
00165 "78AFA4DA2830EDF2D10D97468F10FE82506175DE9AE325381EBA4C16F9AF3B59B6B8C74E88"  
00166 "CF81460ED6495CDD869E5E64517FFCD6F6936F0BB490A964652B08052C742F66631708B303"  
00167 "402481AD37387A11A8897054B9D5C8DF5EB869A17099361646D96CB62DA504AEF3E18ADE08"  
00168 "0997F3C17AEABC5FCA015848CE5F4237F2D194241D6C9A93170E990C15FF9EFAC8485E8A5D"  
00169 "458D42CCDF68DD3163711FECF460A7AC531DED2FA12F81EDD46395575B7CB02BE0F8AABB7"  
00170 "8D7D5587D4CCE7F282F4FF0DC5FB01AA94EA0E94AEAA813F5E88EA2C57F265B8DAE1146AC"  
00171 "809A2629F37B2A9BFE676D6A33F5C7B2A24A6ED8D05FE91708877BC53AF036D62E4F2A0425"  
00172 "0EF4FD62626E22EA6041AD77B83FD9F675FEBC17C2F17D086949EBAE2612565FC14154D1C7"  
00173 "136E7F2B1599CB6E6C5A08A438620A2143CCA82A40B2DCDAE2E7A342DECE1A930BBE99A26"  
00174 "B23C3285C8F3C41F5B6694F9EF10F3802FE51E3234C66894665362EDEA755195949133D5D"  
00175 "08DDA19154DA4453A02F51AFB275E202FDB549E2AA72AB4DA011E5FAFE35AC8BE25CD4CB4A"  
00176 "CE313395CB912ABC8907185960C7227CC10FF9D471B19B6E0B27A74E0579E262E7CBD62516"  
00177 "1C45527C88A2AF58B1F9DED566BF97CB2EFFAB8EC777A8E38A3B643B6D90BCC31A89E07AF3"  
00178 "A84E31204474487C0813EA7C8781F45AA81BF5A49F9159FAAE985C882AB9A411F1FCE97DE8"  
00179 "D4EA4A477224267FA6CB1DFE0DE4EFBC889E69636CF51C6F579B3B782FDEA3ACEF89A0254"  
00180 "68AA1BE57920A5A1C2D2390834D4388CA49E363C25E0B4AE63B421B0D713EF2EEC1388BAA"  
00181 "CB7876140DBBCF88E6EFC1E9253534E38E94EECC9FC2CD6840FD5E7EE81B0838218CAC5138"  
00182 "13601FB2B1599CB6E6C5A08A438620A2143CCA82A40B2DCDAE2E7A342DECE1A930BBE99A26"  
00183 "2235438D5EAB5162F32534A67D99BC30AF66A97DC975C93B4955B741AC0A1B3A033B765D5"  
00184 "769C4BC0018D5AA4FADFE595AF6C79E7D253539E3604DEC60B5E5E8CF3BDC59B185FD35239"  
00185 "927D2CAB7CC319A20549A8CF7FA5851DA35876C95C65B8DF4EEC2166536831579A8E2AF3D"  
00186 "5BF01F6F14A2EEE03B0A2497C17A36FAF23261FF04A82B8301FD5108E0B12D910C5EED9772"  
00187 "4F9C7BABF42390A0D603E25EFC734AEE71B526BC4B01AAC7890C77D4D260E23A86BE22D787"  
00188 "9B194AE8E94D3E89A5FBA7A64438F541A6545CFD3C43ED1CF805C40289A6CAB8E879D7B"  
00189 "007E43090A9CB9C9D463991F1B14FA4E0EEC840DFCC8293267835ECF3DD145465FFFE397697"  
00190 "FCD1D71521435BADD8388CE0FC07B458AC0236C2339C53CDBAE4466C7FB94F507504827789"  
00191 "268726BB2C5D9DB3A3F700F0ED66A412C05EA6D3CC03C900479EA252D87832010CA8E6E3C"  
00192 "2FE98141B1A6C7A4FC3471E308CBC24DE9E134F6EAE1323AC962CC2497616E6FFCD4279403"  
00193 "CA6388DC1902C0B41BD7EA8110C44E60FB5C08261C42735E91B2E40122BCB2966D22723516"  
00194 "OF288E2AF24278926177258EC33CF02BAE9628314A0EC6262245225B2CD49082B05491B00"  
00195 "DC0F7FA8268448491F02F292E13CA2629FAE7ACC2617300A776483F06738E4EEF4BFE7443"  
00196 "35980CBAAF37A4B9044B50D7EF8C5014B9B022D69F021BBC09B223E3C9C06ABB3BC7CA7F6"  
00197 "25E3E69F498C877008F9A53C6A2D42D1115105B860", // PRN 04  
00198 "9E39D51E661E0C75E62D1C3739E6B454B4971B299A37168E4D2B3788859300A530A857A078"  
00199 "7012210813B9215A5FB377C42F6B198F057BCCEA4D3A71DCE22B893373A3D342C10BE1EFB"  
00200 "B764C9AC9B1F762AD0C13C5872C87462D71F7B013C77A4EACC79244656CF890A2B738B586B"  
00201 "AA846476F5ED8C0898491E96ECF2A563978EE78C084482F1E00128339A07A511FB2A7E532C"  
00202 "426A147A6D9AFCFB9E9BE2DBAEF57E9874B93F43499E3C2B46599DEFA6847EAA820B98DCC6"  
00203 "6D1922428604FBA7A48C810CF6C9EBCA4A7455A39A88DE2B70FEFF95CA3A68973E0E288923"  
00204 "E29761E73E73FB18B97057CF7C98996E91C09299F9ECAE5D754CF2C472967647365AB7A4A7"  
00205 "65C7E3331F42C0E0F9CA5C3BEC59E19056861F11C68A36309C99B341142B2B66503F22C767"  
00206 "0BCD3CF3D9548C346ED08F4384D5DFF49C69A9A8F99EA9D1157CECAF140A9B6EDC8E279289"  
00207 "8344F9B76979409C8FFEB6DA107D32DEA5FDE3393D2CCD5BBCC8ED8499F58A611167AB89F6"  
00208 "1119CB827482A81212B8EB2DC2BA0B67C2B84996801052AD41AB95B0CF27CEDA284AE01861"  
00209 "153C41951E09152766BE75C2CFCCF447B30D68146E38C7A38A923CEADFB997998970CC56AA"  
00210 "710976754762A9B29E7F860961659AF938AFE5A85B8FC169BAAD1F2126193643C05D07A822"  
00211 "E30EF801BCB836ED8DC2A36427F773114459592DDE2EBC95C1AD12E7C91139CCAB8F45C2A3"  
00212 "6278E93D4F5DE7D999232C9B0A69535A714E85393A7E2C05E956390C0710477A80EAB367A3"  
00213 "4E389C5FF74CFAC5A940D64C35BDD458BE1EB10C484B0A25CF8F341ED44496A578B266FA89"  
00214 "110DE962F2290E5A49DB622629919D7E81092CFD2C3C918D23CC80330830A904DE725E0F8"  
00215 "5EDB93CD9F253AF2703646E921FA79C0B710AF0341E555AD2AB5B818B497DD6EEA7D737155"  
00216 "03080745DEBFA72FC86C9C966F0C1869F69CCA5942787EAC83D26DE003A39959D0C5975A67"  
00217 "51AF27A7EC5F14A40A2E1DDFDB0FED0A186BFF687636416638DB91805B8371DE10B7A5983"  
00218 "00E36168ACFE39D1A98A15DA450AC97B59A66AC0DC1D4DDB7D9D3851674E1C00B9E15A8E0B"  
00219 "3709C40A30F9C5CE51A41907685BE086A69B39BF5CB085095FE242768664FD2341EFC24710"  
00220 "014D6806DB1EC351CDD374285EDF813E62E6263549E7AB98CDCCDD8778E2F8379494DDE05E2"  
00221 "053C53A4E6053382E9049EECD45945DBF4B6DF0CFA0769B538EF9B176463EE8A9FB6C8F0AC"  
00222 "C9480DDEEA5AD146491F29FB5B243974B6073E8C15FC3672146069C1A363418C5E4549CC3F"  
00223 "F3792540B1E9409291BE4956991FEDF1C6DDC18ED7966413A67A11794271FB56A37401EB3"  
00224 "4E4F7556C8A2DD9A9E854A43D29B35569F465ADD226C3023F698D7B64E8E7C29C2DD57FFA"  
00225 "AB06ACD7F383E0C2EF8A6DD13CC9F1B629CB4751D159255115291115056FBB615C7B738F93"  
00226 "0A3C8ECD6703837EB33EDA12132B99C30388C5B8838671A315F312DC50997FE0A77FA7530D"  
00227 "7F66B70153AB60D80FEBBEF5747FAD22CB93F743781D35CC85F4DE1819626C4829EEE0881A"

00228 "EE1FE4D2BC59E65113DD8F1103C762B763CF7FD211D1AFDB80A2BD9AB3EC93A139E2EE007A"  
00229 "3ABDBA21F590245898D25AB87ADF63BF0B3EEF5B33F4E18F984D27A73B3D77B14D178C949D"  
00230 "EDDF1F18E01BD332B490BF43094DEC9D0F29EC5CEC1B65B8040B7BCD04A8C2A7AC065ACF"  
00231 "F8FEA5BE160C56E7D8B7B1000EEFE0584C02E2870B86D32F2946A4615D8C73C8CF5A634A64"  
00232 "0B65CB8A2A1276AF6C3D6715C4CC088FCA08BC290", // PRN 05  
00233 "EA7EDE7F64B6B80F492DEAA6BE36509A1DB8225760CDAEF69D6E682C6B0679FE860C2C047A"  
00234 "24D7CD35DE983BBED286F2400E498D6D1D1C929DC0C9F379B312B41E293FFC1DAAE0D8370E"  
00235 "57D41A8847926E68EFBE114508B67A92A28BF80D3ECA1B8FA23E5B3D19908AB6DE56A856A"  
00236 "D876019A6621B933AD563BD522289A8C13A52509A875FC8E9BE87AD629CCF6272B89005A37"  
00237 "7563347D0B971947C5D13736983AFDD5BBD923297D34AAF4DB3F63EB95AEC43D56945C76A9"  
00238 "B31A7765D92F74725BCBD81D76D103F57D91B4184B0E52CD9612E388954C07D97C28A1B1CC"  
00239 "BBAD61BA8EE8A8796959388EAEA7F36BEB2822CF5C650963463BA8C8F56862C436C566CEEA"  
00240 "3F96845658A0105647E1C35ED73BFE3F04000C264A2B607CA48A9BED86BB1C75200323A525"  
00241 "4404E9813EF14223F0D7F66E52AFF952A903297A8ABBC1447F01F89DA82D403E8F9A08FCC"  
00242 "A510D09AE803EAE71CE25866435A9A64468C2F2187AC3A307755BED43B97533C07416141F"  
00243 "8E5D90B421A087313C7976FA1D99DCA7C15C4FEE4F753ECFD4158FFFD4736706F0BB6B129B"  
00244 "54450FE398F6E317706350A0614181254A839767DF7F30B31EA84547F89A3CD9458D2683E9"  
00245 "46D2D79A1A12739251C86BB0CFA6BF2AB70674201B72ECFD0025EB8D24C872FED7A5D052A37"  
00246 "B67EA7A02225EEA2B7F65EF2727B5D26480956EAC7479048B92F273ADDA3E92D152ECBEF51"  
00247 "0804631247F634CEA988B7D58D47002D00D0923AD04B2D561A8751F947DE34C7459918521"  
00248 "8804079CD7F7DCDC2C4406372A7EB63B91832EF3D24BF9C440FEC45CEDED9D8C5AA305B3C4"  
00249 "7A8CF0687E08A94FFAE9308ECAEBB86F3F51A34C1370EE8A585BE01463259B409D89FA011E"  
00250 "6FC64C4AA9E059AEFD7F43A2193440AEFDC0323D2F7294F2CC9B85F4F5DA336EA25DCA666"  
00251 "E16370CD53D3624674F097BA3D15DA34376CD721F2CB4D0A7FC716C16A4FA243DD8D89C22"  
00252 "3AABBF4C4A4B45254EC40AAD1D15648DB2FC63E6A70832FEBA9BF10C1C19CA20C59BD7EB91"  
00253 "D0ACDEDD004259AB134AC1CF90E86FAD39D630114A218118D72FF790EA29E906289F8BF4F6"  
00254 "477DE0F1A585EE7054CB3195BAAC5B638061118FF5BBD6967982A3789A455FE7BAD5F0F1E"  
00255 "48D7F254587DFC7C7809A8967FA0F54729E0A86BAB5B2DC3DAF8AE12207CE7D968838B5B81"  
00256 "E90B7E94BD88A79660E4107F970D3E86F77C29399CD2EDC31AF16847CA35FA221F6C113E85"  
00257 "1728F056ABED87C1E424371408CE13EFC3B620ECCE09E774FE88DD8B97438F9B236549D7"  
00258 "E2BE8DDC002706326F14172C7673FCE8797D3E424A8F84D63C79BE8686C68074B0503F096A"  
00259 "F9966C8EC55841F2122B8C1FE6070E487132AAD79E7790C70E4582823DE647E9B269C9F90F"  
00260 "BD27F823CFA0422835943E9A0801D4CA513E1DA51B22CE1533CB2977DE21AA5ABF436DFA"  
00261 "FD4ACA03602DAE22D85672AC8995DB1C8B2E1938E29E892C0833B5D95381080EAD71CB9BB"  
00262 "9CADB337091956181CBF92A8FDA413DEA5255648D4BE7585F94F216ED82D3A4C31441F176A"  
00263 "155E7F2FDC9DEDAFC01F7631FDA0F16106CF2DA29361F39E1847AAB0E5BAFB897621DB63DF"  
00264 "B66A8CB39646B859549101533E3F94DC8FD26C272CEA922E6F901275BF96446D660788EF97"  
00265 "EEB116A54FC63C6074D94518F48DB4AFC39465ADBF030F0F092FCC8E6594C2314CC366E8A0"  
00266 "D876A2F39ABCE218F6E66DA47FABF34EA0F7FE464E30C675AAA01ADDC26359E219FD52E90A"  
00267 "B254D9A73C74DF9C1BC4C98C7E4863E33599389980", // PRN 06  
00268 "F283216A0505319CD7685323FD568BB1D0AD9B85232764959973B2DB55620E2E9BC31B189A"  
00269 "013592E220DBB8345C1F5105373D9786E74D5BF99309B25D1DF61800F2685815DBD1C8C01"  
00270 "DD9DC2D2BD8417AC887027BB5B85DA397D966FAAB1D8A1061C1D6875B5925879BB3EDBE91D"  
00271 "8DEA9F874F3708262AC41FB31E1D58B892B3BF9C6AEF7C8C23EC2A5C6F84B53E006575B7AF"  
00272 "BBF45B23F99B539945A93CDC6122312DE09FE27692E9A5D06AC135EF0B24DA853A95B36E"  
00273 "7E9AFFCC9586F65736F54F8B29BD3C8058D89AD336C8CFC758F3EC083C851F0A737069B2AD"  
00274 "3C8484EA141ECA1F36CC0665B83786B83381395654512C94D9CED2D944A0F494CAE3F0D636"  
00275 "A0D7AF68AC9B3E5EF120951E3C31862293AB7A9C16F0BD267D77082A507C858B83EB893"  
00276 "83DFB15CFDB2415827C574516FAA0F6CD20F9F7D2AE801DEAAE54A9A25DF9509B659DC1A02"  
00277 "A8DEEC84AA72635E8FDFD68E5AF04F7B980BF24FB44322889780796C6183963F85BBF38924"  
00278 "90C2F0FF4B1E3CEFA2EC73216424BAE4F5A3DC13B5B75434B03A44B310D0DBD3B6DDC9312"  
00279 "1FBA802349933E54C239A05609C7CFF25D71A0FBCE0359CDB352772E60166B1E8E65D165EC"  
00280 "FB7B9E4090960A2056D8068B2838415204E3D75B62617AE0ACB20BB296ACC26F33D5C8344F"  
00281 "1FA89F40B960D0C22DC87C55A3B0E0AC8914F8C7B830BF71943B9DFE52B6BE1C9A44AE6EB"  
00282 "AE0082B9265FCAECB2A6199B0F195BFB65C215BF0829F4EE3267A422D9ADA363D8161070EA"  
00283 "B8922BE03E6089BC3F579ACA97AB7F1D02EBFACD87EC57B640AAF1FC5160F967CF8BC87A6"  
00284 "3DD3311C18639C07DF20EF1C3C0C5F61805ABCE74DD7888D4FFC650FA223F7C892B9ACA85"  
00285 "88F6B326389468C542262EA2CEC5933494733E2FE01D18811B0C869AE2EA3503AD1A2296C1"  
00286 "1D61F68964D5258F1C0E7B7EC09F7B03DC51C98A32F6E96B8BFE256B2181FDFD632397F63"  
00287 "2F252CF461FAE2BFFDF2A9AA44E02DA7A8678057493DE24EE3E11A2847D84B3FEF26E867A65"  
00288 "109EEAA51746FFB0D2D2661812B233E5BE21F06C496D21047EB056FA7D8C97DCE5D77A400"  
00289 "81FABF2423E5C92A0F4C45907165CCD1CF51C56A2B2728665CFD058999E55FE8FB68A6F2F1"  
00290 "ED9C649B7531273EF5F18CFAA384755AB1D73E9AC6707A7701C6A3251DFC9671BA29D025FA"  
00291 "72F36E581762F8C41FF6B9C0EF006A1B8BD2166DD9CF89851A92C8BB15A27CCD597622A51E"  
00292 "A718BD78B35BE1748466A0CC2464E0497E562F898C45237007152C8603582D50D05F9B11A"  
00293 "3FD6BF3CF8750A301051BE191F36617D66F67C98B221F942D1F1C03BCCDECCAA82D566398A"  
00294 "B534DEC485840C3E22E0516D9205B83EFA59289E66A747057AD0304E6002B14F6BD9EEC45"  
00295 "8EEB6319EE9DF6F9EB4536FD39D0B9FAA0F94819A9706190821EDF74FBB69D893EAC0D3D49"  
00296 "0B936D6A53D0E56696A0050AF484B7C5DFA6A5B45998FE7374BCEAD95C111EDC742AD33ED"  
00297 "64C48F5A5BBE257BDA93B941F8412A04BEDA8E888FCAE00F4376F4E53E16C782F45BC3553A"  
00298 "FCB9CB35230EE9C37B423D22CE7610730C4521ADBE4578A34B3DFEDB1CBDA875D56D3135"  
00299 "95B83CB8D3544C06B7CF341E6BFEEA5AF5A50D1BC439647739A0AAA98ED67B7103A33B9A75"  
00300 "E66662F78485BA9AF2E1F2D15D7F9C13E12E77F871ED909643DB41ECE09AF25B11291DA8A87"  
00301 "2146ECBE2D3C39CA3A6DE7AF67599C422279E2645A238799D5AD93917C15261C71FE9D6F37"  
00302 "7595D14198587B145E49ECB9D04C07C9051FA4CF4C", // PRN 07  
00303 "4FB0C9E163CF95D9330D8B1047AD6FCB28ADE705EC3E8838152105181C8164BBA0580529FB"  
00304 "5E7BDBCBEE0B67CA9A1D8AA19B6846484B126A267D7F9AA3FFDE6CF3A8261F435A483015B"  
00305 "75FB39A6860FB9B7CC69D146ECB0025DC7E79A9880D0BEA3AE8A396B201202512C9C24F3EC"  
00306 "278031C8BB04DB9D2323592CACD7E93D5452359D920C51A9B8433A1AD42623641C52A1866A"  
00307 "05B914B2BC7AA0E906F4E4B931AB759546F4B793F5C006F60A7BDEFC6D79496E03E403D737"  
00308 "2EC2385FEDD2E227D7E5EF1126DEC4EC14BD6960C93B8FD924C2023634C22A2B16542B9EC4"  
00309 "1529F4C9354F4E2B047A602FED2BE9FC91B6F247A2291AF02B9F9E3C464A8046F0F9FD0DCA"  
00310 "9C918D3C57C076B498483A0714AA401A504F3EDA93BDB22450D81B2CF97D855384B823AAFA9"  
00311 "65A3FC320CA3EAC08AF2E7D00A204C5499B7A01DCF83AB47E161CF59B45D35E5D9E47F08FA"  
00312 "60738C17A106A1D4BFF74BCCB4A897C8D3A99D1436472FE24A9B5BA56300A200221F619C92"  
00313 "D298F908BDAB6325BB2B3BE69EF7FA959633CA6169B2DAD376F212C93CA1439BBFF3493D"  
00314 "40148EA772E191D1B5F90DB31B9D910521D57FBEC93724AABF55786368378ECB95C2993645"



00315 "FAB2AD13DAC66647B36EBDD0BFFC6EBC8BCAB78AA9BD2D7067D0ED1C5A190803A7D1F1FEE4"  
00316 "CA5519A54D0D7C70404DBBD488536485A33136D2BD2FD1479787FA009F3EA1235CB451BDD2"  
00317 "6B384EA79D2FDD6D95ECF2365E5A2EE06C8B52349872D77B32CCD53E5A28BDE3986EBCEB57A"  
00318 "059013153761EE8E783F5B34E088835D440758D1CCB8A96A16EA6CDC373453A584C3F06844"  
00319 "C685CA3F2362235452317AC4F6B73AB623F9A6179B94F2A2AC59E3D105091EA88C4E524915"  
00320 "5654770CA613E0B1F7D74C709D61A090B22A53B16D05E02D04442E92A524B71584D37E6B"  
00321 "1BFF1841FD15379D5A4D8AA2FBDAA078663408433234D89A95C5A4252B87CD12BF98C681E"  
00322 "7909552699657DB90728F021C5B3EF0D59F5D3A0B2ABB086373E35FB935C9A1432ED932596"  
00323 "FE8182B668EAE731AD081E473634EFC212EBF0C8A86841BCF5819CD429203B0CB548FC2AC"  
00324 "F54B25C2F1A9B3F98F75CDFB0C0D50CF232158FA44A4CBA828CE2E2F23B5CA8E80EDC4F27E"  
00325 "8CD0C0879A84DC821F9A7EF37FF285C599BCEDF34E4C07958C90865DE349847CE454BDC9E2"  
00326 "3F977F58FED34CD2801373FB7B8E4299E90F88CE41E73F47B1C2B46CDA3181281C2CF6432A"  
00327 "CACFAB5AFAA83BCB9ABB002FDBC0A0DD132F08BFE7276BDFDF827E36E4F7F1E4BBAB62045A"  
00328 "420472F251FBA2432B0701C0836F3DADCF6864C1AC31F849D353945021F7231A09612D46F4"  
00329 "2C6E9A63C247BCD01688E7D6DFB455A344C6E6D4183B6D497C837F6E04C40407540044B57A"  
00330 "FEBE2296316CA0E2D3DC0F196248EDCCAD4BF4C8F2ED45AE2EB8EF352619EE6F79D10CE1BA"  
00331 "7A5D6F34762D9A82AE98ADE36E0BE4529E081A11450EBB4F70AEA0378148F38C50DB624631"  
00332 "2C7E77FEC63B8ADABD0FFF3D1B2CFC7384D0337956B098FE667D1A86D6CAA1C24156EE3881"  
00333 "EA7EBEEA9BD13C0D5CA57B44008815D32F2F0D30C41719510B4B167E9E1419CCF1E4FF8FF2"  
00334 "CB61B051A5A1A3267D08D106D1B15F0F485B44785FF5D4CD5D95BA1CD4FB9CDB5DAD46A099"  
00335 "7A3A4E2F8C0DF2C536AC3A402B406F9E522D925ACE301274A2A0BED02F062F8C780A42D5F0B"  
00336 "A170F4CCB36930655C4B825634B72D2B254DFEECB9A171A97F2EA6E2BB43D00F89B92AD5E"  
00337 "141818B58AB94573472D636ECCF4868B7A761CE4", // PRN 08  
00338 "F0AB64492124B092152A5EF9A1D3C1EA9ABCDABA0BE2B6E833DE5D291E7ED04B1E103AB01F"  
00339 "ABE4FDDF722BCBD8B7BB3E451FDD3D241019AEB2596F99632497E3A9D8212E4E35E523966"  
00340 "AD1A028659087E2B2414260B9811289130ED5E999E123E739755529C59C6D5B53F24C912D2"  
00341 "7757975FAB7C2058398F13DD51BDCD27E8A8EE45BE96998BB9905972F919E8F06B7BBD76E"  
00342 "C00229F58C2BCD6C50541D16AE8C8E5FCC411E5F9D1BA8BA4EE839491344F22FC41B4B561F"  
00343 "F43076333B16737866CB58EC2F5A228F79E05E0A49C364347911F66D2B48754A56C992765"  
00344 "4615C47B096F4CD3D13EC8F83778369D86DE64B95F1BAE796510195D83A3CD8370F4D51079"  
00345 "2BC2751A13A3EE000449C995463CDF824188591B453B88C6123E5063A1A9674B83350A07C5"  
00346 "7EC68590BB99A2BDD3BE8753B17EAB98174DA3A4296037435D2A35A67C2D44B40AD017AFD4"  
00347 "E32EF35D982A2BACB7D3DE110E1B530BFC9A03FF8ED64E130A050FA21CF9A8CDBD9BE56F4"  
00348 "A68D82EB0F5F5A816FAC7FC37263B7BFF8C69117443C3060E531B7EBEA83F369FB7F8EBB"  
00349 "F8521E85A1B32982E5F95AB3918CD9CC29B65AC3478A97B36DB893BD29A20101A4B6065CEE"  
00350 "814E3FD4ED5F07337B049CBA0BA7A1E837D64DFF70859529D7C1F835C9294CAE0003033509"  
00351 "4DBBDD0C68D413C893ECE208A9A371BC802B5739A42FBD7772BB5013C1F2EDE8AE138CDF00"  
00352 "658135B2BE5CFC6B115201C8F11818D97E414DB04AB1CCE8DAE9B2E0DCFCCEB968617FD53E5"  
00353 "195BB6260127449CA276B6A1D1436C028194DA0CF1B1FB2ED8177645F543CDF6693B78FE4"  
00354 "789AF52CD6DF731C258827E65C15455D8BD8EDC394112B63159E60194B3828F74E70334ECD"  
00355 "EA8FCC80C0045536235806D9AD863E4E57432D82CDDEA283FEE84D43C5725F75C72468115C"  
00356 "A91E29B13350D918933C19403EAC3CD7AB637139E5D37CA0D6ADA2587E03D4A760A792A50E"  
00357 "B0D3B78DBF96FF968A07259E83FEDC1CD30D2ADDAC577031821708CAC3D3AC54D0E0345965"  
00358 "214E94CAD858813DFCBABD5CA331980B73C83828984A3CEF59F05D466EE6FBE6E0FA927A4"  
00359 "FF2EAB5D1F28564611C0239F69FD1D5DE3378C36B1485FCFCE8521B431620FE85873F983887"  
00360 "29A33AE79174827B75D148A85D52C7F91B9187ABE72BEF39E5509C0D2B70A5E8627227E58"  
00361 "86B3AD35E454866675DD3365F6E627869B4CCC0536C1B30F563A9AE01757282CA517041DF6"  
00362 "3627CB991F73E287B51BB5F0833FDE7A88A2BAB3E18A3399D93A4CA71DC935086C21EA4"  
00363 "78793FC43079FBAEDA897B326302FF4BCFA5C5295137ECB653035CD6C12120CAAF6F2CFCD7"  
00364 "A9DE1B37987AEB506DC0243EADD301CDA8F313101DBFDEF6381A40BC5CE19F5569D2459F18"  
00365 "08A0C406BDBEE678111E1F882839E1FF0270323658DF7B3628FE6037C68D2D174AC5F611E2"  
00366 "13CF0372F3CF6B5444F9BFBFB0EF0DECBAB0766C91A84A597A045FE874D4C18C72035E2FF48"  
00367 "1E823E18DCA610552FAE185004D90D226592E361E98F250ACF0FC3637F51C993FE5263A305"  
00368 "7A7D0DE690F11A83653E023C12EF263211A17658F78D7E274490FEBEC35FF38313B1C860F4"  
00369 "61F21FC4624B2885D180110AA5AEC4FDB886ACBB1986A7BB5B9B15A7E35CBDC81931310C04"  
00370 "7A14BE6225BFC647DD634D1D010444917AF27768309E3EF1870DAEFDFA23788868110BF223"  
00371 "12C2573AD326F7093F91FE73B6084682CBF19BE60BC0B1303A4D36303D851EF82AB38ABA"  
00372 "FEA16B676DE05B5488E43D0BFCDC2FFA46E5DE866C", // PRN 09  
00373 "79833B3A3E625E3B24FACC84F7C58A25036582B49F69EE62F0174F1A006431515E9B77FAFE"  
00374 "E6A4FC8370038FEFAF5A3F71A7C7251E264A2D445C90AFA91A8CAC47372FFD16A8428C5CE3"  
00375 "7E3BAA03F3D9111676E455E24F75CA70562D1ACD385EFF6121CA19577670CA0FAD805E776C"  
00376 "94191238DBFBE00CDD037A5AFDD9CC972B3C1CEDF3EDA8B8965AC6077251D219872F301BEA"  
00377 "DA10D23A7696C5B4B41F3568BDC7823183E18BC0E700F9639A91DC4E893881015286169874"  
00378 "EAF816F240E001AE137B20B9EB54AC541A0DF264A861B7FEAC185E794C367F2859234FD224"  
00379 "67CB201F83C5C5C27F9E4DD7674621FA80397AB67E236AA01A18E0D99CC13EED9EC54D4EAF"  
00380 "3732E02D6932AE9119384FB7A1CE111FCEC7E93EF064D5291974D28E1D77ACCF443633107F"  
00381 "69E343DF308C78CF6C8BDAB65A256B5CB2B264441D2D9E284074DD486E63DF81290C61DC"  
00382 "5D1A4C0B65331F60F1F83DF4FA60A090224EA70B4B9472CA00286667D200E5B0822B6FEB48"  
00383 "2F468C3D44AF45585DF4BF0E0F2A33E4AB67756173D5CD805A705D2C7FCAAA6529B7733DE5"  
00384 "9A1FC15597FC5B96A1B83CDD4CC5D117AB42A16F75CC36341A7DACE024494FD7E518812BC"  
00385 "17ED227D0AB7999593248B6C2C675B0DAF9D5BC104426F115E48364B17C3A83D49B7692A23"  
00386 "9E6C840E12E1B25CEE25EF1B847FEB8DA3B4B783971A27D3605FBFDABA811E6FF196021E33"  
00387 "B34A34F4EC4FDE2792801FC2A0287D148DD93206020BD45D05450460B704225BB2CE7B5A82"  
00388 "AE9D8F59AB93C34252245DEDBEE02E93D6180E82D49847BDAF194F62B7DA2A1D63C2DF7ABA"  
00389 "7977D6237716EE4E7BFADA13A965B4FD0167A41EBEE39C0E5D69281DE2B3F08CD4320007D7"  
00390 "8184C4C2852D9106DEE4E9C2572EF6B4375A37BEC65CF18E1E241EA80413F93C763E7F9A9B"  
00391 "F2B7ADC95129A6BE6BAC24B35FD214ACC820DC4326E36F0DFB6E185A9BE71F7E473FF8D364"  
00392 "8E9930456BE9209EDD822EDEE10222EB01A8E1BA3AF11D90C4EB1AFCADC1EC5A1CFBFF21D3"  
00393 "B4E54EA74383138D1FF37FBC9F1EA6BA7AC08BE4DCF18C67CA16510BDAE8C5F0D058B68D86"  
00394 "719C96D1C3551A0ED7C14E5592C81813EB60F9D5F85B6078A5406A9DA454FE3E46093EFB27"  
00395 "7B7DB99A07C342D2663DB9C1969B50197F0D0175425BB8F2A1F7DD5A6615C49409E2831324"  
00396 "D18D6743206BDC379FB5C7CB44ABFF6BD1D952016AF95EF2E385E6488C482C96A5C9832CA"  
00397 "38DC27176D58B453DB4D2F193523B958FE4EC7B6BBA0C2CBD693A92E8D98EC8527396FF58"  
00398 "195F872CA14F0A6DF0CAED78CF1F0E3CB4C9169DC8C9491474786705FFA996A6052B72BA8A"  
00399 "2AE5B489DFF56B57A5CCD5A0670BC46D64C78D26B4EC6747DB0C5548DE1303B8642DA9FF35"  
00400 "B2EA015BA154238B0CE0F8EAA7D7C0D7768368781F89FAAFD32BCAB1A93A6EAF02442507D"  
00401 "7C14702F7584CC4B17DAAF7A0AB42499ABEE3ECCB80DDFB851EAE421E7ACA7841D0E52613C"

00402 "777912B3C4FA7E4D8FB19662D680B7D361F959488C2EE6778C139887ABA8DDCDEF0F6A89E2"  
00403 "D032C1491349A0EAB14BD50D34135372DD8230435430BA0B120CCAAE4F26C40552F6C671F9"  
00404 "6852E9FBE48FEDDAEABFEE3EA44907E35B6628C4B90F5FF795D873D345E8FDD01727F262"  
00405 "1EB1EFA4D1F5E53B6C7C2A9E674B6185B4A259CF7DBD9F515AB26D319818C786059BCD4B39"  
00406 "80C8C14C4740B3E532C363A14E23BA470C4C13CFB600B8342D70D41F4EB6A0FE2933959C3"  
00407 "C6B733C35E64E0D77EEA6BB15548D7B21ADCC6AC60", // PRN 10  
00408 "EC2D915F0070493562F04F3549330482E0473736D3443D68D8D7ED6F0BDD523393B5FE4790"  
00409 "EEB4C928046F05DAA556EF8F39556D48FF434563834A79F72AB1F1275D2517E7FB29BC64CE"  
00410 "16F0B465920C48719622E618CC3D7518EB53A2F728B335DF34151BE0E239A3747072CD0A78"  
00411 "88C8E7A4EBE7DD38FD84ED0E9344EDD98F3B3820BEC4E90909CC1CEA1C0EC2A63D139598AD"  
00412 "F5618C795B1D5F0F86C81CDAEDD7651B2CB67DDB7198D4A5509B98BECE7120AB6F28977D0F"  
00413 "F04B7C307BF10ED7BB2C0BB3BF14C138FD191D54D28C77A0A88575D7F97372FC35066ACC0A"  
00414 "26A7AF5B816801B0718A37D666A492ECC2CD7255AA076F3CD30543DE651DB11C9FDEFFC6DE"  
00415 "1A611BA3322B1A19ED1B13F78412322A0452B3D2D7C88D3D4574094D97E5D41581AB8BF7318"  
00416 "D4C5B76D214EEC890275DE64FB24952F1C8EF54EBA8C7C3557220B4DF722DFB77544AD08F"  
00417 "18DA398FBD1015B130C47B26181B67AB8A54DCB069F3715D168653D3865FA8EFA344ABB826"  
00418 "6F7600410321A50BDF753335F804E7B95E3C0484F1CEE0D94110548669E46E88F6C94466B5"  
00419 "1FC990F5B34206691AD89B76D6C0086EB50C57B6BAA273014458B32F7ED4EBCEB7BABA9295"  
00420 "2FAC7A53287BEB1E060FA0583840BA971E4F9DAA9F0B5EFA1048D06ABA360A4D3206AB5C62"  
00421 "20630B37C21A5CC60767BA826A6A638F9E31B623E4A4F03D93F610129DF0C5AD8D37AD5395"  
00422 "93BBC37ED9174BC4685457F7D9DDCAA4D4ECA24201AD128D5F3A934859561AF9BB450D200C"  
00423 "CE3F545306D2301D6C6BAA78AB8B7D0B07AF521A5DBF88D7536239E13824BB1F32773BE663"  
00424 "D4FEFF49C154B81B468F316D573DFDA280090D72A7845A6AF918472E284C295FF96947866B3"  
00425 "F9C4C0EE8FA4C0A95943570079FDFB49CD3F5EAC5B43FF4D95F685230409A9071151D86909"  
00426 "6D10A75A50BD7576729D98FDD5257DC835E18945199CAA55978C7CBDD0A897E2E9F09322021"  
00427 "E0EDA0BFD971E199AB3FE147FCF62598CC6FD9B34266E86F540BF85F0BA93D1441FD45DF47"  
00428 "84028106BBE50BEA5EF5BE9ABF37EE04D380E097423A50370BCC8BBD1AB38C772551978112"  
00429 "14757515577AA38174B0D75746FCADA5073BE4009F7CE732E0EC2FFC40DDE7B14B79B3DB62"  
00430 "DDD8C93CA73CED3721F2853865D1AE262766746E7806F4E22DCC2CE04FCFA415F157A36AA"  
00431 "EEBF3A5A80BF9A16114CC2D5CC734B1D0F6CE593768DBFBC0858D73FC761675EB153457B90C"  
00432 "736EF866CA1A07EB98AD056B73177A8ECC4E1FD771D996ED6B0B30B96396340DF66EE20972"  
00433 "5FB9C521A832ED90AE4E810AEA8B3135114F21BA188386A979D08BEFC35B156B3B12E5F1C4"  
00434 "84028106BBE50BEA5EF5BE9ABF37EE04D380E097423A50370BCC8BBD1AB38C772551978112"  
00435 "5990A1CB4E2699CE59A4073DF795577F6988DBFB6C06FD3F4F49A60E2D46ACA0415DF52765"  
00436 "B03B1B210DCE28765F7AE7F4B27735EA5289700B1FD5CD62FE4413A442EDD15A726B9002BD"  
00437 "EBF2CF8B856195988055CAF3D4C2E4526CC193B1A73CD1643D564C86B3D14E057CE9CDF4C1E"  
00438 "80D5C51609160B943DD1B32E21E2A188E07BB12C3E62C069B312A66F8B3D8AFCB549585D00"  
00439 "008C37BF010886FF12B002A7886D7EFF5D627623E1ECDCD31A55F207EE8C2171166B40744"  
00440 "18E84095F787C24D0E26959DF4E816B1ECF31D0C3A61C4189FDE1C8E13BD058BA27300634293"  
00441 "47EFE80716CE6716C68C304E04E9D0B790B9B3F0DEEB14B9212A1105300C556097E170FD19"  
00442 "F3B8A6AAB5F861EA643AA18FB1CE5E24F491DE3DAC", // PRN 11  
00443 "409B1134F8998401BB764652FEF53D692BC6803D4B470BEF7B23F53CF5DC11118836A648D2"  
00444 "768CCFC08A79C7DFF43E1AD138B84E475E8AD831DD388402DFC10087D79A5E3AD1D80B0986"  
00445 "E1B8B0CC19D433EBD411AF02A7367D5AAE7A66C7E60FDAC4CEBDD31A8220B9487BF09E0798"  
00446 "449B0C658A419489E79366C9C8ED6991ED427864654D4FA68B234917F4D6DFDCE23863580"  
00447 "36DB8E9CBFD686281EA26AB0D39DE28698C9A2D3261DDFE5326766491CD4EBCCF98EFAFC49"  
00448 "0C594A5E69B1001BA0E773052ABF1B55621F86CA12B4A3C9FDA8CAED348F7A9A6073456535"  
00449 "AFF37087F87C24D0E26959DF4E816B1ECF31D0C3A61C4189FDE1C8E13BD058BA27300634293"  
00450 "0810C51156B11A5FE717FCD813FDA1043BFF3827ADE87E9F440EE8BF90E95857E691B97227"  
00451 "7611F1C270DA24475586309A188C832435F7B1F1E78EA8DA17DEB8C4F92374DE573A6CEDF7"  
00452 "1A1EB102EBAF07E7B58541A16E93C077E141FDD9AC50FFF7CADD9769F9E572B3DA71D0C9"  
00453 "614C8AD0AB73FE09A2A1D2C999225F51CF0AF547DB8D7693ED0DA5F11288BD3C6F37E1C492"  
00454 "64F243CB3BF46AC35E86DD2B55FBA7D0F2A3D7B172AE829F8E80760402E075F81EE9032376"  
00455 "39B98ED1DA870613F9E3C075820A7FCA02A5585982F5B9989FA6F142CEC5500375502AC599"  
00456 "F5C98D587559E47CE6D3FB1D05B110D2187A113CB787E0BC69AD39678C199DA82FBED77C64"  
00457 "3CA165D349A808F953E4C983B026E60FCCB0CC5AA1B29CBF8D6329B95D3DA0D0DEE4CE5D4F"  
00458 "D264B087FFC66E0C3162809FDDC4B1AE7DE9E358DAA7BDF9A65F98F0C4BC9C39E93F560F05"  
00459 "913BF1106C7C07F52FFD07DF7A7F2C1786541175437A599B9063B1FC6722118CC4822A2291"  
00460 "3F565A398D96D5B11A0D7C661D5A8A8F39757DD0EA5C98CEF3B9A037CA5987A05C7279C9D4"  
00461 "D2D36B386B02761BF85B21E7472FEBB9A99355D0B8A6D688B80EC37156F2A6AC04D532B060"  
00462 "9343C77B0F9EF08312B5323EC4E07D3CFFEB6230B468BED5FD9F00175790253E4F3BBE057D"  
00463 "5930D0815AFA8F7A4237F4B315577EE080D6F220BB3FBA999925F1A005F552575F4AE63360"  
00464 "4F1150B5C805C8D80C702222C754860EC27648C07FC9441E9975A8A9254264AB8493BE99F"  
00465 "7F2D33DC4F805BF98A485459580C3F478D01AD0B13D01C09F4AC02BEF02C785F0244C0422F"  
00466 "842FB939858B3F10D73299BE79CA375FF18417057B9102797154611AD8DAC02179A053804"  
00467 "D9D3A7BEA40675C83852C78B2A5C0864B8A5D0E435FC20C003DCFD5E7095515A47B308C7"  
00468 "EC25969D69D95A91D65CF5A4505E250240FFB52E603A831B40808158D2932D3AD31E27B08"  
00469 "DD09512534B59BFC2DD1C816A0D09C3B17848E5D63FEE578E609A5B1268CAE52E66E1435DC"  
00470 "A7BF12A4FC1529D5AC9CD624AA19FCC934F2E1CAE19809A7EDEE2C6C75595E125C1DA37BED"  
00471 "8936F10D4F67C00ED83F3241142B44D2C1508512AE98DA632857732869AAD5AB692019EDA3"  
00472 "6152A656BA90C586590617EB9D7F3813B09C840B3E741681E495D650BC5AEA51FE7303F74F"  
00473 "F700A8D19F1E672159BC330889BA1A9B570649F15320D730F5923496FCC3677E2BE5506C55"  
00474 "FFB2D649803F7C3E9AA7FF000D9F3F039245CBA83EBA2CC8511DD411B4C696BBBDB6E498C2"  
00475 "722D0B8E597729FE6FD2324C14F802ED50EF06F25830DB70D7F46C82DD7DB1AA0B68B706A"  
00476 "4F19FC0197009C497D4296F1247417D8994B354DE873977EFF02C4676783BE31B734972BC7"  
00477 "32C70EBA5F6FAF125535FD3127716C67CB7F3E444", // PRN 12  
00478 "397E16AE84EAA97262B8839E03D5DA85030F2EB3EF773BABF2D7BE79631BD7F699727F539"  
00479 "FFCDD05E0881BB23BBE8EF901966AB33D42051CE7398012D8FFF17E0D17D4035F0E3CFD0F9"  
00480 "E94134D37B275D05E589F1D4BBC6E892D4EB2656FB7E5CE6294F1D870D6F096DCEAFC3E893"  
00481 "0E19E1D9548A104EE4AA7173DFC8AFDE66579DF4F961891F83E9F455DAB2A83C9691689459"  
00482 "1BA3F5A8FCF1EDA3E7B9A4B86DA95DD065D0961F60363C00C61C954BC2004F383CCCC6054A"  
00483 "846FD7ADA6CE3C425236CEB9A374158329EF991320ECEB3BC5A4C272606466EAO4E2B4D50F"  
00484 "3A9FEB73A7EBD946A992EAA8A5B6E3CC5527E2E25857EF19B5C0ACEE07FB77C3D63792BF"  
00485 "1759B63AB024E6850099B28A855EBCADD17A68145C8694963CE125A08A24E699256434AAE"  
00486 "FFDF8F1C9A3EB70EE1EC2FB0CB5BBA264FC210659A47A5C93A0222F5DABC9464640FF649E5"  
00487 "FEF1746B5D497721FED69529C02D3D412C073D91191711F52D9DBCBB6C219AC4FB60696516"  
00488 "CDFA0A66539A333DF3A7DBAE2EE466BA048B424C438E40144EF9BDA2EC89FE1E74B4B9CBD0"



00489 "FCCA4523B5E2A8C60FEE2875541684DF536A4AE32CDC7F6A7BF087A4885F3C25341BB44E3E"  
00490 "CEF18A78513F7D89BBD0896FE9017788478C9C694C495683650E39B8AEDC4A3F7ED279C879"  
00491 "399A669914FFADCDFCC470DA5E18949A7A036E9A4F17DD4F08D3111CEF5A30D2EC8EC4D149"  
00492 "FF1B6731DD63D1DEC8EBFE98E6C619278CF1352F63797FFEF9FD74E8F986BB74842A3FC09"  
00493 "2E76869ED8660C02AA9F829496F6BC3E2EF78C61A9F6990F37227CA0940945FBC4A79A63A"  
00494 "186F2EAC6AD6494B78E2ADFDF4CD0F17B846F1A88B92E389D7A7C31BF3A2B32285CB6A6DC"  
00495 "2D27848BF9966E702CABBFC64F84092B47A177253B3C3585E0F7DC5124ED9EAF05B7FC8EB"  
00496 "7007F18CC9CCB1943DFA4C9B101121CF6DAC0B75B480B991FAA25B0170A3FF8924BB145018"  
00497 "918743E49C34154C59291D02CA1226900BABA15D475EC07797B48AB24E7668A35FA5646D92"  
00498 "570A5ACDF2A8D478B763154F7471C60811E003E3917C2CD3353800D77E9981E3C6B4439147"  
00499 "342A838B40B3A472482E647F446F29B085D9C8C7B7C5B58053AD6D825976FA9D2DC007487C"  
00500 "6938C75EBD1FCBDF6475A43474E246FA1A91EB68F80E2EFB6B7B42239684DF2211451241DBD4"  
00501 "51853F5632C887AA3686D8211611440277CFB01A28B212F88B0F4864BC64F407B5137F17F5"  
00502 "76C9011444FF2461B835B20CDBF170A37CE4CBC8442A7212125A660C5A1B2F783FE6707584"  
00503 "F07E724990CD04808B55DF432711E9A73AB9046CE722B635B5BD20492E0B0A1C085E8F926E"  
00504 "98CC54A94D821DBD6ADF91CA9706A049140480258C11A4CE7EE66AFC56C2A21B4784671FE"  
00505 "612D8873FE273CCDEFFD2FD42CEBCAD0FC0B8C2B650876E248BA1097BB0970B5113E02DB30"  
00506 "36F53DF74D8CE6352F98B5B8621D2A35D2259870FA2A6BE07D5CCE435D10D29F2FC197DE2"  
00507 "12FCC1926C7FD49CE08CC492A56960B783183F88C87A403B4D012E18E063F27BF0B5F82D75"  
00508 "688C2207962ED06D711F31815D94FF16DC32F3D7A698F770233FE524D0534F4D681F2A2574"  
00509 "A4D6F38D7EF401767E2670A758C3714A857086768DE08FE428845344D15915A1725CE9BA3F"  
00510 "D0DAF16CB1349ECD1EAA5F2E1F1EA58B3A793A22D1D37EF0DB0F86B8D9075904FDE73104EF"  
00511 "7589E092A234394FBB3CCD9A91CD6B9A67A3C7DBE56F2ABB860E72490D6F756D11FDD7685B"  
00512 "98612F33CBAC9A463E7B76C0D36A3040D44149E3D0", // PRN 13  
00513 "1E0FCD6CF84B05DA800B79CE6A08053DB3CA26A30B8E860D0F744E77D2321BB9B9CD586211"  
00514 "119FE8FC01ACD5935884D4B13CBEAC08B1E1E39E68D6A7E172EA0BC49F467FA9A1FB4A67A"  
00515 "66BFA5D12A92B993C2478F604A4B5FD2EF82F9A3DA4E688FC4EEAF439A9AF576F632EB26C"  
00516 "686884AA162C682F80D23177F3C2268FC9E33A4A3B4F58327E13913CC82FB732551ABA0FD2"  
00517 "1BB9A5939BE9E4A50752603B2FEA8C903ADC9A7A32EBCCB69A2E1541EC43F30FD64A02461AD"  
00518 "1501B6DF7EF8F6B11758BA2959C68D422E9924DCB9F11F3CB669054A691CF11FFB725437AC"  
00519 "9CF73A3DFAD63B0FEEB77BE7A4B27177E2719A7A3DB19CC4C2D280D0C958D617D53A9E3A6"  
00520 "E4D409AE7CCCC72C7FC5B6941CA931453B4D718471C5DFCD5224947DDC569EED813EF0FCF"  
00521 "1B569693974E3A2EA4A8AC06DFBA884ED632F408576383C11F5993E9194FB3D88CDD5671"  
00522 "9C184ABE0BBE8E0470B9A7EB29B2E903F96F8A0FB06F2801C62FCDCBDF41B8EEA84609EC4D"  
00523 "825FEFF18895EE74A1EC69715F432409BCC8A26631C874D798CBBD7A8D43874A5209215247"  
00524 "1985FFE611FFDC3025A8CE37591FF938AE34CE8F5EF9EB5572550193AF8F76EA89161CA2D4"  
00525 "F20309B56F06EAC2F57E6E5A666547E8FA77A50276916AA987097ED51F2A9C080E02B9168C"  
00526 "FD3AE956558393B3DB2F0A7ACEF7E6203F562A6598479FD8A0CC1AAF6C29BFD57C12A6BDF6"  
00527 "7B67A53E6C1A66A9D342908280142B764587B31A788F54EE9C66B000483FB3A592ED0D0B8C"  
00528 "8B37F03782E3A2A3AB3573F442FD88E4276595FF4505C76206DFC7A85BBF77AF88130EF6D6"  
00529 "8ED313364B45FC8E846F4AF9ABD750BCC908F57CABACDFCA674785B0D05079484AFDA8CA5D"  
00530 "E93CCA721E29F05CE5C361180B618E40625978C9A21011FCFCF1D81E8E98109792138710F6"  
00531 "3AB1BC46D9D87AB41BFC15342137CE27768BB27F96A00AA6D802196FD922C7E3FCD03E380E"  
00532 "9E23FA716F7EEB21AEE3A2B8143A844F85FEE0E334ECB9C8454F7C08253E3BC65C69971D6"  
00533 "93ED65A287D9ADD6D561ADA1D9A5A892CFAD329AF98AA5AA3D64DCFEF5671BEDF2E8EC34E"  
00534 "465D6736F9A467DD8AE904E6AF0AA913ACD059E1F62CBA857B355F64A0BAC546099D0B1293"  
00535 "BF63876B4168506E196730C739A5E6EF742BB37DC22F6028AAB82BFB7BDF838048D3EDDC75"  
00536 "1D993C2275C6A89E32A28E16E7D585A72B7EB3078B6C0271B730B4F93B50E9206A24A151"  
00537 "D06D0340811319AC992C1681EE74F4825E7029D02374059FEF0E02716A17A2663A2759F374"  
00538 "965787A9201E2058065E1AA08AC52B830B85BBBF9B6AA8CC050234749FAADCA169C68644F8"  
00539 "791155DBBFE7F12EA6EAE229B65529CA4B7FA78673039FECDD69019D42C3ECF927F2B70A15D"  
00540 "1F249344F7073BABE2374119846940390F3B8F79309B04102137D6E5DDF774863B84950FAF"  
00541 "F25F52B1778DE93C79765E4AC178C32345CA66B0EA431D6F81592A02D9DD10768493573FC4"  
00542 "9BC4DEC8B567F897C7BDF5A428C062165BC723AF220CA169EC4A6102B9739FEFC0EA17B2DEA"  
00543 "B9E66F5DC749B52A93CF873F760B9A2DBE2D5C85208BC0FC8269A91CC37BE3DFF828395548"  
00544 "A5B4B1A6D695658047A91DEB8B4E397E2D2806DE2794B0B51D9196165DAA336FFA13848E48"  
00545 "57C7B44AE51DB34C64002FD02F8DF9FF72E242C344F2B5D58D017E0364B130DEB11863D5FC"  
00546 "C814090DE66F0F1A9913EEB45050BC45447407D801695274E36210DCB3753A7645540BB0AA"  
00547 "136BD572B2E2ECC0C18D46D60E4F52EE5A342B1A5C", // PRN 14  
00548 "CB2F5A4B8600DE451C3185E21C019B8BEB0C1433929CE2139F841CFE511C74257A30EA27C1"  
00549 "D4311599B509929E7BE1EB872380D77EBAD14065EB3536BE0E9A9DA3E936F80A9006A47A60"  
00550 "A6C5B110EAA5FDE98A34527D430E45F247E05294714E747B6BA1455FBE24AB02952DC7E5F9"  
00551 "D302C3056092471430EE366698AF5D791A1286375EDE47996BA4EE98E6864329DCB5A53C0"  
00552 "83D6A3F104C0415D476B8B6242F6F3DEE1EBE4B0E541FCAD230ECF15357CEBE779CBF2782D"  
00553 "8868D80EAEEFD99749918EAOA75DCB07AA5C97F47095D809E33065D8EEB4F8AD52968829F"  
00554 "F1FD0DD6B5E1322FE9D1760218E93BA20F4755F9707EFC8A8E62A4616C5B0E9D109B480F"  
00555 "999DF5E4DA312C479589A9D1BC9B8761C2ACD696C4EF23A2CE9F512F13AB072E19C618693B"  
00556 "FF74FE0019BBA98EA19483755BE6B81EBACB1102A800CD08C47501099AFA78C67844A2B739"  
00557 "AC11B7A8AF6B2B1206EDD9C622D90BBA0F3C3C2EA9A0060F55707DEBB8239DC228F20EF3C"  
00558 "B55C2FE7ACB112E9F6C2132DBE3AC47E2E15C5C437D31B4E2B3EAE2BD116AF856065C9D6B1"  
00559 "CAC28797C5ADB972B770D99DFE101A319BF38A06E977F2076EE754F34004CD31EF8BB86D21"  
00560 "DF12326CB19F1E87461B3C789457ABF1270A573F0A9154E9E0858883D903C7E9AAF30C5C11"  
00561 "30B5F1E6FA784273C476585ADF68BF2697CC282CE68073757F70ED01D20FC72E6C0D48742"  
00562 "77A87B9338291B13CE87478F2218D0089E1D5F2825B7AE7784E2BFD291741B2AD82D201FC7"  
00563 "5A487D4F57D6F5A6512C9A29C60E2BABFE00CD3DA6750262FAAE513A9668FF8CD5C245DBD8"  
00564 "95390BD494635B46269FC011E2806B06B76DDAEAE9E26CBF2A6EEF2193D58F6AC8BC49E101E"  
00565 "8677F85C3605939569F16B207BE9D65E884FF8422E82503087811B080E78896301D8044"  
00566 "A890AA724D0EC2A58B54D72E506CFD025C0585DDC7BBEF2E233E025DA00A61917DA5045113"  
00567 "9325178F711DFAFC029D567DB48B7CD4ACEC62EDA5AA669557B21EBC97AD815FBB028DAF36"  
00568 "976F6E3FF4E1EA838BF54CAD0193499E3FF2B58A44B5EAD16398834894335F17EFD8B579"  
00569 "988CB3CA9D1CD7940F5365A3B71EE7B2DBFC35B853FD8D28BA3759D1CE37542288DC910D40"  
00570 "6D2A8D8064054595E23DFB013E0E4AE2357ED57AF49672D07A14C73B7E528ACB4CB3BA325"  
00571 "748ED135DA14E1EA838BF54CAD0783772E0651C8B9D903A0F22D0E077F38D74FA47AD64BDC"  
00572 "DEC92A9401F780053C0420B21CAFA92F95923E2764E0B5F70AE029B90FA6D1315C7DBCC53E"  
00573 "944B62FCCA0E32AD55BA279CB8DB58BE2B18EF82EBA9463654592CD8C8FD4E5CCC6325B2"  
00574 "F11C8475D7A17121FB52E9E925F05C1CE3DE1D0BF7A73C291EBD881EAC6B771C3EE9FF1E7D"  
00575 "2372F18CA98627D166B7F280A7DD20C634AA715913ED551E049DA98BF4871567C295BB1653"

```
00576 "EFF925400A1694ADE4F681E307F009CF2097D45B3B92910376FDEF17E58E02B67B06C7D5C6"
00577 "29206F03392EEB69349C6178A2AB8C575FFBFE6CE62DB14AC1E1E9B0C3D9FA63686F3F7B6"
00578 "F69E9B08C406C24EDF2FF5DDA3CD2CA735C4F7610D734A914A6FB8934B8A3FD5887808D236"
00579 "406E3A187DFB593004CCB6C8E74F0C3CE8F09B65C8525BDF0D2B5C22593DF8D00D7F907B28"
00580 "164D1BB312120745BDD623909060C5D0E0EDA4BA7994E5D0AA095506A0E8BE8FC3BD06B37E"
00581 "4CE83A5BDC37C269F035D3DD904C7870FAC049D777511CBCD4A3E13B896A47A5187C410397"
00582 "D1DAD5350242D8CEB657BB087B7AABEAB40F5355C8", // PRN 15
00583 "C1079AE9F30B67267EFCAB6BA16F939BDE6B605F19B8E81F4F7A826C194AAD2C121735FD17"
00584 "8A0128C4D56398748AB280B71BADBF5A4B721B0B252D678BD601EEB565E4574283CB497AA"
00585 "18D06E125D31B480F1E9A0EE3E1A5923B27C62884C7A54FCFCF52BC29F6A0F05B1E08ACCB8"
00586 "629D55CABB677B61FCB4DFC3EACDDAA6CFE205E8EFD5EC90B0D91EF96942CB656BD0CCCBED"
00587 "451CA9B5F48708FCED5625E3A78DE96D8B29DFA1EA3854FA08049B5558100CF396FB8895ED"
00588 "D66C45CB005A9C9A9DFDA3CF145B9DDCF4F1CBD2837B5394E43207BF3DFD0D7EDDCABCA69A"
00589 "6D34107E6CA7CFAD0F9C0CD2A9FFBEE9DB853EDC98082744FA1824952D3D0E246FC17A409"
00590 "64D3C81A28D5206AA02D7275DDFFC3A47EA05340E52E8DF0E49CF44B831A5445492D6B21325"
00591 "DEA58B6D702CB5C08562FE3337B5FDDECF4FC921A2D10C3482A7E3725D58D0DC03FF80D03A"
00592 "3B07199D1B3F745F59CCB60271B74B8A83407BA0AAC982E1796CF0F06FDFCC87536AA2F411"
00593 "EC474FEE06A73C2D1B1D6D1AE56A41A87FB5E17A6A2CA02E8C95AC278469603D1DC5842801"
00594 "A8C778FE24D1290D84B932AF7E392270C3DDA8CA9BFC672C45D340D4A19E7CE329E0232410"
00595 "345C328A9032EB2536608DBAE21615FEA83387C4E40BBDABB51C66965101FA6CAB537FE2C6"
00596 "DE465849320F5FCFE3BC42AE050AB039EF6F7DA5B8BFDE31FC27725AA90A5477684D72184"
00597 "A7F2040410C7492B6C8ECCF95896B0AA4517F269D6E890387377AAB9C2ECCA9D84B5A93808"
00598 "30B40639D4590B37537D2FCD5AD78622DCBFAF60C98F116B36688B80F8BDFE7BA6F6837DE2"
00599 "34F5E68E6B052A235DD666983E41346B341D427CF897F5182CDDC810A50CC9E2D6030F10"
00600 "45FC2AC7BD0589C552960487887B41AA6F7B856B57EAF114BF46F2E6D9E41A49C33622FF81"
00601 "F0FC4816AA9FACC4DB8CE1E4A3299376CCFF374506BC9993ECA71FA9A779AC5645AC7CC6E6"
00602 "6CB5F76DD2BB922D9CC050C74A1F0E8DB81C72A48E1113F2DE3440DBAA7AF07D78E01F9D20"
00603 "29EAF04EED1736D48112273CBCC6F6A07DFB669A33CB169B48614CB7FD22DF040A3AB992FE"
00604 "D2C91F8FFCF85E8D805E152E975B4117AB5A95B8DFDC990A2BA5491784CC4959EEEF34D69F"
00605 "D1A9A6AFF427DBAB376B0CF81B80250DF79F7D83ED9C0F9A9908344A430E9ADDC4785B810F"
00606 "27AB81BC815B440F1450FA91E455C8C5F177F68A0A614040D85F4833BCCEE54F7C1B1C3288"
00607 "F874B20D9612CFBFF9A03DC6EA85EFC9D79F375A831DEB7366920C2D30AF74B76B5A6C8FF1"
00608 "2946CA1020DC11A918659BEE80C31B6CD73EEAC84E076F5A0D6E8110F22AFA09317D95F385"
00609 "81CC3899A8EBE7AE341395D75BD7B766434EF2794DE2696DE16AD85C3BC368178D6A8C6E7A"
00610 "B3EE93E016025B1369068B37D0C5ECE95FA40913F7D01DBE4C131148E5F4EC008C521CCCE41"
00611 "35568BB78F3D5DB3A99E42164719544FD9C46B81C571BEE4FA275B9177BAF561954383636"
00612 "1D791853D585754AEBDF2CD675701B05BA4E433441C578ECA3D0B6AFEC7A25FB7E999A5CC"
00613 "943F856445C2D90026C84A438F344479EB8012E66656B29EA6F694B7329639D1C7D27D5C0E"
00614 "7226C60560B1E7DCEDCA203C7690A6547C1BD8A20497C6607097C916E1F589263D6114342"
00615 "CE2631DB041DA3FC4E25696EFD1302F7C9A6F534AD67119FF97D450D4BFB5EF2B171233861"
00616 "768BC74798F325569F40A19434964D7F6FA9A24BF66C066C1E9D05BD00FCA9E800ED8EF3CA"
00617 "C84018670B30AF2A0BEE42CBD4B86C8A2DE436D568", // PRN 16
00618 "2D9BC6C52B03EDB2B764893D7286EAFE28A970D4D2A04A8931352C0F75A55268E8B8CDE2"
00619 "06811EDCA7FDCD7D25D4A6D2E2FE5104CDB07C8D38C36FE83E224192A909E46F34B5D19868"
00620 "71B1E3BDA92C076FB2839BD4B126F9D0DFD2075A9FD1ED319F2C27DAB014C15F7F56856C00"
00621 "E34C8EA6C6D2F6BDC884E87E9BE5F9551FBD0A77662F323894E2C257F8179A5A9E04DC6E07"
00622 "42DE45552864C918CA4EBECC4692D9EE6777B27C2C0B5F20CB9A89EC574224E1B555B851B1"
00623 "2835967AE9AC982C36161E27D0BC06B6F28BB4A631C7163A6D6BF76CC9FB6308A2C171A27"
00624 "064C08EAD35F33A019C1D8E98C50B18F53E1D3ADE72B5E85ED8715A38D70F67BCE396BD8B7"
00625 "EBADA1F957C69832AA90BC7DAEFAE9F993EAD3856CB2E8B9D8F62562F8DB47AFB49DBB0D62"
00626 "DF249887ED4F3BCC175CC83465186EAC4B4AFBE9AB509A5F4EEE58885B913E9E8B77AC1DF2"
00627 "931782377B716EE943062C07F25E65EFFF92C43BF6384B2B2BB893455322DF538E6EA00ED5"
00628 "1458135D0FC5FBE383E22A0BD4538E73A2E973135CA1E3CBD6EF4B07CB2FD40F927CA58A64"
00629 "13543EDA572D3E4BE20A662D22A4DB5FADA3D8216B4DBF0A3F0A20FC6BDC18AAB99A60E55A"
00630 "12508B34E33D1DEC21239B011780C5F3D72EBE627F3B7C8743E4A1AB2871CC8DD853193232"
00631 "A0C55236F22891BF8E50C5C2B357ABDE8F342F9AF1A6E348DFFFD3445EA996ED3D4A889F3A"
00632 "09EC301547E278E32D82150467F0B3F746CFD1A9EAEF91B3E16CBB18A49F21B9D6234DB77"
00633 "65AD1975D77A4CB623374E71435DFB5AFA08B6656EACDE013DD1DBB09F4F2D18F79E484C38E"
00634 "1B30E9E20370857B33DF13CC3B93531A0FD8AF033C8431985FE58C3BA62AF728E826A2671C"
00635 "1A63F43DE9BCE537865BB0AF57F90767AD3384B2D4939E026DF389AB6A04C93A1B313872AB"
00636 "554BAF4F151EA00E752914C3520308814C2A7E92CC5F6CD7BD715BC9A143A39BBCD4123287"
00637 "CF8C4ED6D1BCC5F00713C04EE851355BF69C65DCA8C5CB86D46F90AF25A07123C2CC0A7042"
00638 "40257C933B94172B0432E71B489C65F6E9A2BFCB87F3B34C28695F925A534D0ABBE7471D6"
00639 "875A118B36D37A298D9AC9219703D184592FC901148F1CEDAA7B02621A4451345089FB02DA"
00640 "94D8F20CC8EE0E0924B041E5FA477499629536E3B33693A5152CE357512844E5A13029485A"
00641 "A4169F53ECC1329C2B844ED50B32B2A220613B1F42EA888B68383DC4741B154F851A820151"
00642 "A36C17621081096AD3D70BFA8847515989CCAAFB3CE06776D1DB548693322C0908E5B063A0"
00643 "D482EF1EBE75861F2B5A8D1FA764BA121D69DDF1231E769E095EF7B42E38EADB19F7E3BA"
00644 "CAA9163B9595B1A653CD0296E34E3B31F591A1EEA947AA36C55BDC958C8E5D499697DA820"
00645 "F8D9CDDC09B751E3FDD3EE49FB34E9073C08682D1DC7138E2A2FAE8DC4ABC3A28DCB11A6B9F"
00646 "92E2A9881CB731005394590DACC935F572BAB67034B6D9C90E966B7EE7A3379DFE33C7D0B6"
00647 "21817B8FA9A8D93EEC77BA493F8552EABBE2A6904C7592E206A9A152204EE63021CA7555A0"
00648 "5DD827FD8FC70DE92BC7E5838CC3DF5ACF7F32BD70E65BDF4B7C6790118624373F532C7AB"
00649 "82713B4BEFCAFF2E855CB00BE91CC69DD1C5FA977782609200BB785E86CFE21D17DB9417"
00650 "FCB938D0833EA6BD96FA1229AEE293AD0880E3DD287371CEC3F899BCFA9DC9B66FC627307B"
00651 "A331F1020AF6233EE00FE8B01FFAC9CB7C464F20D97418C2AEA800C878551C701CFB3A9645"
00652 "FB70F2CBB8C0CC73F6AFA69460E6D3D630EC403034", // PRN 17
00653 "BC514690938DAC703824A90124E422A0A9B4ED89CD999D8B2773ADFB66A47505F8D7B6E848"
00654 "79EF37ACC61C346C0BCD5ACF4489A07956CDED10B81CC62EB6E8ADF7F4BB0FBEE539BD6FCC"
00655 "B4A4DF88BB3F8FA2D22699966556731BFAC901215DC15D71F3281153939D05F90D79B0FAEB8"
00656 "EDFC15A2A788CE9A405EB431E311AC5650CA6E3B865DD0ED525448DC11BC7E77668EB1A601"
00657 "5351CD4A46EB948BC9C0299C196745094B023F49A07C4CCAAFCB298DF9F0E73BED2410B"
00658 "E20192ED22136F4022E37E6FF7A4399C609CB2F5B51031DC4BEA90E4C379CF0DEC71C232F"
00659 "2D49CCB35D0E41057CA75A6D7063E7B50B2CA0C4C1163745F143201D393EF90DF677A3EF76"
00660 "1A4A9BE62115C51AE02381BF1FFAAE4933DEDDF2B64F36CCED0F218CC3A3083B81740FC97C"
00661 "E1D5DE2CFF334496E5A670AD3EE658AF9427B9D738A9110BC0B45421C857FAFE5B8FDD1E6"
00662 "C876D79F451A8920BDF48EAA982D7FD91401D51F0FD514A44568E2BA8102956532FD3B3B64"
```

00663 "300FFF16A23AB982BA7AF15C99892E4D4C8FEBC79BF5DA567DB0A7C9E781BC5B5D7089056C"  
00664 "13D1BFBB7D18CDBDF15D223F7071B641300818B828A95CE706C7B40FF21DB74474D6412B6AE"  
00665 "F302EB0679441ECA16AA5A5FFAA5A4BFD8E798D8790711B0F785386E45BD78B010D719FFC"  
00666 "32DA0465EA087F6837A3E67C33B9D266683500880F191154C3ABA78944805D4DEA155A1171"  
00667 "2B72321C62591C22C3E8EA9CA1C55E1B2C10AF545C7929A8F035F5C03DBB6B86983CCC37CA"  
00668 "59D58BF0204BB8D101714F7D5EB26D19331CAB21A24A8A655B18798D7C3302168408C775A"  
00669 "B5243DE216B0A7FF61ED51D4F2CC30B5631EDBB84673A1FD3C17006681F05DF6170213AAFA"  
00670 "423CE62761E64821758E9FA8696471298325E8B691D9FACE7CD8E05FD6B6325BE098123BDD"  
00671 "0EA59BA74369A87270675C05CA79CEC1572647DDE6D8BB64AE43A760561B1718124AD77A5B"  
00672 "07CDEFF96D436E208A9D84FC6FB001EB30789FD5F1326A33C9F2D73175D9D6E4F02563A59E"  
00673 "A69490153783A84C2955BC1CCDC0251E041481110D91212CD9582D6D345CDCFF2C71B08CB9"  
00674 "D3480C651619BD08112602329C5F6D2652F5FA770F47D0F61E7ECF2ED8205A71B6667181B5F"  
00675 "79B93D059469A71D2DA196FD98690BB58B4DC218B16DC650CBDC57FF7FF38909197C597BCB"  
00676 "DFB7ED0F883983C18CAFBD80D56C071E2C2B64E33CEC03A58225AF13F3A8517DF3A23D90ED"  
00677 "43E7F429E5EEDBF99F136944B61816EB9E147B173AC33551C4B6C05FAED10FEFB202F4EBB"  
00678 "0F4A73AE53D3E7600F268D557555CB61A823D36ED537AE76FB96462D8B5ACCE8427DF32371"  
00679 "FF4F32E66896D3ED4478943F08D0B7BB528C2A39E7CF77C621577C3DCFFA5BBDC6A4D3D0C4"  
00680 "1DFOA7F141EB60B63750669ECE2CEC7C6EDF0660BF96749D01727040AD51D4869585812D3E"  
00681 "604F94FB8C90925A4AD3859D7DFBE83CD1A88F8E3F15EA96A4D745301B3480A2C6463F885A"  
00682 "32BC173FBD0C8C8BF8768DF2041ECE73D9F17B71119438DF45B99484859BEED1938CDC0"  
00683 "CBA05B8E117BBSF2205FE6B5062B429C8133FE52FDD887443EFA8176EAA21FD567D124"  
00684 "D7912599DF30E3B76079572436439192C714F45DEAAA892256DF38F16825575F9AAE7FB723"  
00685 "6D1F8860071AF421D5F6F9E75F5D39218726E24C8BB3C9EBF3269579C421DFAC0B5D369103"  
00686 "1F444B9A9E7004A91A6061B3C0CCDE29FBF07B1A0ED82BC3947ADCF3D87B423CA1694310E2"  
00687 "9DD31663C11B8B8E062A9EC4D4E0FC19DF916A6C80", // PRN 18  
00688 "F848B06D24B8741482555C3315EC3D051B23BC975819C7A3B0D152E1970E1FABF323853BF3"  
00689 "7008713374D685E9CD2CD0CEBC6A8D5CCD62FD5603BE5074AADF79C6310542F1747C5C7A1C"  
00690 "260AA8FC4729AF5E497DB117D635DEEC76810DDC5E67BAABB45F835D6580075744E36BBAC8"  
00691 "0CECB358A8EB36F60C3B47821CE2EC482CFB80058E1AE5CC37700AF07895D30745AFE02235"  
00692 "B1E7TABEEA117DB5F2205FE6B5062B429C8133FE52FDD887443EFA8176EAA21FD567D124"  
00693 "89903BCD8B7476870F2A247C5069EB644B607732089818618BE394F3F99FB340828D0396B7"  
00694 "807E5F323884023D371F650D95084D2C44A82507509497A99EFE83B4C1F6546D657CD72DFC"  
00695 "2A6206A4C44D1DA9AB81EEE49DFAE323EEC5FF3BB4CC02F3AABB39975E152327682BA5EF0"  
00696 "EE8286367D49FC824496690189DCA21361372DC48BBB3DEE1CD96A9833DAD1A161838F7E63"  
00697 "BFB67A7A5740D7D7C15051EF9F1902BE1806DA5207F4F1715ECFAAE928035C5661B76B5703"  
00698 "D5B9444D35FE966C353918D52579DD87ED0695740103AE9B49E034C5441DDB8D68274DE"  
00699 "914083AAAF70A2507E725AA9C1B7BEA38E45C85CE08004B2F96B5BA4181E4F8B53DF62A12F"  
00700 "D44800AB28076BD11FE30DD46AF779DEFD25089104156F9B4F00FF4374E6D5B3F40470F8CE"  
00701 "75B2C59BD2B8328714F446078ED2CBA3AD5C34BFD3B4D673B90EB66A67B6BB6C92E5BAE22"  
00702 "A663A7C4A8E279F5604BF2454E8CD43154B68AD5A352087CF5503D43AD65C942B2A24BBFBE"  
00703 "048ED195E75831E58F468FE3049B3BB0A53908066A6BB912ED2F76707DC5AECFA0D94C929F"  
00704 "1EA950F70163F0373A1C450F62C013AA72FD63ACA6D3C2F3CD1DCB571FCA6818F7784D8FCBC"  
00705 "9A3CACAE2A577356AA03497C4811E5418A0A7FC97F01D324BC5BF50025582D33F3E7BE235C"  
00706 "717FE3F8677664800AABFC0F846998CADF955B7F039171A9AB4A5F06FA6FA682F4989B0088"  
00707 "87F46792702BF23E66ED196C0033E225444664734B62FF55B277B531FAB8968CECCAD65E22"  
00708 "4CFF621CEBF558D3013E6F7545FC58FC8942A469F3243321872540812211B8EE2C980138AF"  
00709 "4CADC65EF515F80370BDB8A5DD64A7840FDBA73BEEF284A2B6260BB668703D442EEBC1E0"  
00710 "67649DE66BCDE7894D08B1CBF30B431A53DA99E635AFB5617D72CC4A3F8E55C428D5D7D3B8"  
00711 "E4E152D61460DF112531A5D9CB7C06152E893088D1C5EC150991E46C8D310B3954E5FE200F"  
00712 "B74116E9E37AD62044FA95A011F1D4427984734375E05C46BEF6B65CC59D45BF74FFDE18A0"  
00713 "37D34A0A84BB356029431842EC8EF5F0485E32F0898DD90BCEB47F1782B02EA1C580E5C95"  
00714 "C7779A0FF2A2807A05D62B9A1446F287569E60D73A9695A7B6CEB34813B2EB456239567F65"  
00715 "7A6CE18DA28485850D5ADB6F1756948D4BD4E803574A12DDB47F5D2D164713F3CC4C60FFAF"  
00716 "0B5ECC82AED6F19916F0CD8C1452B41579A8A40E7A1D4EF220E6D78EDC6A592E2F30CA677A"  
00717 "F87BF22707824D597541CE0B44ECF15683088FA2ED3CFEE9D3FA2A3A4D405926511B25909"  
00718 "53820D4DF802D00431F2AC3FC25AB5684018BFD1F5D7D714E5CEDF0F0DF6DCBA6D4B32D88B"  
00719 "B1B740608CD5DE6724AAEA7FA0111854B8DA2BD0B923EDFBA37330144DA966EBCC5860"  
00720 "0724B2EF3BBA076E5B932B34F4E4F4425E2AFA63F57D932545299FC145F004C076E8BB4D37"  
00721 "2F73F6EB6653F779456FFE3A3A12760DC02CD58B5B891436406A7919DFB9AEDC7D8A82F4C5"  
00722 "85483982DEB086258474302E7EB26DA4183D363D0", // PRN 19  
00723 "4F01E8C687252CAAE1EA80DF4411F92015BDD7DE90F317908FABE7B15AF8F7DCE47FB3B3C0"  
00724 "9DF31F58C41FF10A2748B2DE8321E8726F5FD656B7EF7F3154EA9A92E459551913DB2A69D2"  
00725 "AB35CD69242229D5D7E40D18C7EE9B2560C7A9972D382C41C3C0821FF1F25DC86C8CA9F94"  
00726 "E2054CAC07A3E0DE5CD84127C245586A136884E15573E8711BB8B250C91AF5F78D8701E014"  
00727 "F0D3B52951AA4EC7BE99F4DEEA5DA94227665E542108C2D674ED14A129031F6F6B465D5CCB"  
00728 "450491F555FD32E89909C88C3CA00FC6D2AA72E6E13C6C5590546C8A75D8C4A14F3C5B4194"  
00729 "D23F0DFC0B0A59AF83382A9630F81AD5CF275E1703EF17BB96D0F678396C898F462BE67793"  
00730 "0D60CDB0423D8AD9D5402433C8CC32AE542DC3EF514DD880DF842082F2074249F07AE3EA1"  
00731 "D4075A87CC2F6F869A216D099309182524A528C2E3C472A738F2061C87C93E4C0AAC9760CD"  
00732 "0F837F0F2FE9624FDD46E58A102A3254B4B52A315E09B0D5017BF456741E542084E861B882"  
00733 "8E7360F988106F258BD99FE5A89B61D3399771D2A9CEA27747913CC00DE32E749180E4DBFB3"  
00734 "E7F7AC12F0ADB0C2BFD1CD116E9628664318513774592CBA8AFE0C0FE31384C7453C07A40D"  
00735 "AC6BA47BFD19BC91ED5E808DDF9664567A99DCBA919045BB79B68CC9037EE8C5142B72B65B"  
00736 "A2E8671D771D91BAFE5864D63FC72DECD8ECC441C03BCD96621BCA1CA20A81A4CBE03F20F3"  
00737 "259F49835062F42C247364ECB33471F0D3316E911B35F80A6C8FF8B5C0938F34B1DCECA59D"  
00738 "2F3F44E32CF776A3BEF5517010CFAFCCE3AF65029C3B3D97C2857CD384DFBF2AFC9A42829"  
00739 "AA62DDE01318CE940C35E7EF58027FBBF795833285BAB3CE220DE17CC1710BCF64F53ABA6"  
00740 "4016830445C7C4B483B469640E2FC686FA23BE89B91C28B3D6750FAD9DAAEC3E654B339358"  
00741 "A474F49A1199DB9BA703BEE4ACB98A95F16A86CF5044F0AF81F5B8DAC4BF807DACA999FF3C"  
00742 "77422FAD89CF07A726F9854CAC24F3437D03AEE6DEF8F54642BAEDD8998D5B28743F71E551"  
00743 "5C2998CDD3C880323A1EB0728260B0817E0FD0C8D9E8C42B43245E43A2C18131476F78EC78"  
00744 "A8BB7EB3142BA8D4D66EADF2A61B7E0BCFE26CD1FF44E7E796E0A917E4A71EE2746CFE2B"  
00745 "F313B4FA3CA922327D9F819A9E8E15BAA7506C1CE653FD44BEA49191C21157CF60A9E0E3"  
00746 "A526AD9E954805E36B5D731D5A23B73B9974390C37A681483EFB649734AB686A49B51856A2"  
00747 "C8F2DE5B5AD8B51C7448A98013E7D106D153A910AA814B2E28084B6E71140B30F39E1BF1ED"  
00748 "F253476215F80485FB5D7E57AB816F956D61EFB7D6BF98EB08593911D2A81D55FEF4382DA7"  
00749 "B4BF2D3F5ACE9B1F7C749CA05C3BABCB361DDB89DD5A1B5DB57755AB8FB6570E98C69E0F14"

00750 "2BBFD5E505AA06D87210E2439D2E9CA9DE0952B093893DEE2A94E5BC77B65AC8C713223097"  
00751 "87A70A5D64B0858CD6D137F78E5B0FE02A56F75B6B69C24E87A3DA1A35280EABDEC95E8B158"  
00752 "4889B6ABC004F51FE5631D59FCBFA156D40805B3A5F5F68AF687A81E17CDD77F8ECFF3CC86"  
00753 "99B379D292A9ABF8A3359BA6A32E7D0261720B8A3A836B081FBB0637B5AF6CAB7862A1E8CF"  
00754 "1C1092924733D530C38D4F77C842D2B0DC97AC52A832B124BE512E336D743FAAC4AF929242C"  
00755 "CCD61532596809D420860847D5E270FCADD45521C34481EF5AAD4D025E63ECE511363D9941"  
00756 "5A6D8FAA80CCEBB2E0C8A949A0D5C415F0722817EE147828551B7ABA696155B76632EA94AA"  
00757 "FFAA69C5398BB67D6754B6BFB7A351631CA179C1AC", // PRN 20  
00758 "B16C9B01A00C67FB6F3E732B0B428D301E56C43EED81DBB504B4DD2A7E4B3180340DCC6B2"  
00759 "D457C56DCB8A98E1BBC98FC453EBD696E9D640B8FAF7EEFEA4B78308290B5C6B643F97D02B"  
00760 "689A0FAE44BBC9F8DDDD69D81D58A85B5909520140357B0823E72C7CF345CA0ADD88FF63E0"  
00761 "049F206F03103B3D402DA4E719C0946B882B551B595A1771BA29BF91651D54BB918475A72F"  
00762 "91D11D9D2B054E312F1E243A24F4867B599CA4EC59D88789D00381D872B5BB14CF44099927"  
00763 "78C9B0ECF170076D1D28EA18DEB86898A5278FB285415C591D4A694B5C5E34C766C9BBF147"  
00764 "F6E780F7CEE9F47B411C1BAA8F2B669DD1DEEEA817D2A61A406D97E030BA1F6E9B2A9485F6"  
00765 "E1F30185790895FC968EFE0C274817E860E2ED408A6F34DAE84B895B19F0D75A961C993781"  
00766 "D3615762707425537717A28ADECC678394E9182543C5914593CC12CA6101DA188A0C3496398"  
00767 "28DC541E84EA87BEE8C088FAADBE209DC45C8E9AF542C9485F7804866EA49F42096E54E272"  
00768 "D393FBAAE9C0464ACA0BA08FADE18963CFD2948DA2D9502E759432145D327898724317EED4"  
00769 "82B04EC24613979DDAAEAF0EC1C4AE757494F06138613F92266C9C38F063B8C6277E486DD8"  
00770 "DE8C01B5EA87BEE8C088FAADBE209DC45C8E9AF542C9485F7804866EA49F42096E54E272"  
00771 "F6F91DB707E3A129A22C2C4D4726B8CD13C9924F388070200BC186E3F0DB92195609A064A8"  
00772 "13CA6BAC734C35B1F3AC473B73795F6ABFE0BFD79A046102394AC1C1C4B7EBFB26ADDCD3"  
00773 "59B9BB151C81AB86DC5BDF05723644039D8BB1A7395789A0D10F7466788106187A8A93F388"  
00774 "763FAE0A33D77979B59B24CA8B8DD62A7FD57E1F11003B3A8E0F30387C6CE86B7B1C06DBF9"  
00775 "E1881EDC0AFDF04915E1E54A392396893D3AD7A56464AA860B61BD7D7C144DB048B959EFD2"  
00776 "B7C5A1E84EA6D03CA15103C7F418C919EBFDFC5180CC0EE39266D57EF24B335F4E2B5FC3A7"  
00777 "B4648C2A814A03775DE6462ECF0B6EDC8299AE153F7784B6008344D3B0AF4A4B3D721ADF48"  
00778 "56A5B96AC67BB26092FDC5A793DEE80ECDCE63EF4B62A3035477F8D27211EFEEA68949F1A7"  
00779 "2F8BBB048A8E61D19D5663CBC851AE96F090B10C40126043952F477F7E025157F07524955C"  
00780 "7B9EE83F5375BE3951CD9858C34FEEDB8128520BB24111C858ABA3F2EC2F0C180925C2EA9B"  
00781 "FE80B9CF140CEFF4199BB6870730CDE78954525ADF7F97002AE2C56A098FF8C7FBA7C8191D"  
00782 "7D5E923B206145C8619D4B26C587A15C00C595DC692BF9E1E9852C9B778575E189636B85CC"  
00783 "29C63B49ABFD6BAD698F89EC4EA795BD54C2048FB687E283495617B90F925FC8B6FA71B73D"  
00784 "E63C4E00777B41C3C255E3D42570AD4568FF670F864D2E2BD0D906CF2F97995BC669723839"  
00785 "1BE458711123E656E153B638C02D565FEB6F8734E5AB231D7A3AB3A8DF368011093AFC5062"  
00786 "5C27B6545EE7D8E6E708D7A0AFFAC25E27B6691EB3C7AC0C661555A151458C91697DF976F0"  
00787 "00828F39FB42FAEC5C7CFF5BCBDAB1B48EAF35132F3E89567954981D617B000991644159F"  
00788 "41BE834FE42053EF6EEB51BBE36BBDC1B3E5AD9FFB24707E6E208979E8128B7FE97B7C99B"  
00789 "E82CD130AD9E061EFB11069B32CE0A165717475B1874A4873C29A396A983B0EF949B65E268"  
00790 "02D13B1F0503617C9E1CB6050A6AD6202DB54DD94C5941485FAE5575B03E445EFB80DE5245"  
00791 "6C373D9C40DFBF9EDB4172F75CBB72B4FDBE8B57A768C3D723DA35EC6D79FF48F94211D431"  
00792 "02CAADCEC758748447AAD12C61C651316C23C99DE8", // PRN 21  
00793 "B2827D682D4012B8F6E4C6D5C7A047D69B1A34FC25715BBD5CA1CF71F88DFC65871D7E9D49"  
00794 "209D1748F1B48566AC6B8D253DF76E027090F5FDFC8A0EB82B4AAAFB9393F13F92A7902E2"  
00795 "9318A5A23A4443480419A0173F846FAFE964C7E7CAE99229D6821DFEFDA8B195EFB3176D07"  
00796 "9F11507DF23FD0C261EC61A0551136F7322620E671BEA5585A092DC613F79BA89D38BBDCB1"  
00797 "7394A580FAE9634C2422C609C3EFF7952C04CF103D713544E592BDEAB6DF35826CAEC60F2A4"  
00798 "A07F8B463F3F09F9967060DB17AA5B86BFC695F20E0C16B8500A68175E7671E442E83764B1"  
00799 "CC114C90855E5B02474A24824AF91B377C311CE30B0CC39F934C3F659A6B7053C9DAE76C21"  
00800 "8511E083B94CC15ADA4FECB8B03DCC7D6FD57D22EF851640B050C36977170E7290F0802D"  
00801 "B760E7A1E4868D1E0C2691A514590DB206AB6BB868D4D8DD26D3D769EAA0A950448861D8B"  
00802 "63D03ACF5A6DDBA6CD79724B656F8D02F7317822B25B292B202ECE2F841AA007FC9B740D746"  
00803 "DD1D20C9046C28AB6B8A1D6424A096FD4D2AE871ED0FA56F544AD760D798F167DCE392C184"  
00804 "BCA729C4A10D98BCE22E7671C93D2A6A16679E8F09F961443A9C09F08AF2F8573A0FCE4F3F"  
00805 "2C429C3AA5737E5D9FD515A6F4A270719B036D6857FBDEF2485DC2C9E55F4530B65B090799"  
00806 "D4499D18224414110C28D946A5EC3F950EEF9C4C19BBE28B5F539B50DC2A6D5102243815C"  
00807 "68396972A7E8C174B1909E38FB010805648137A48B71FBACAA5FA2CA57BB6166897EE52EEB"  
00808 "55B12441F7FDDBE24DAA6F3FE75ADF97BDA227E7755B3AFEB342ABB64ECE740C3ECBF59DCD"  
00809 "36BB735931A7ACC963CFF4E694C8A501994082C5F68E756818AC2D919B8476CD933C110DDE"  
00810 "2172FB4441EBCE3DD21BDCC3B0EDE857D7DE44ECD00EB5ACD8B6847F0621EC24357054FCFF"  
00811 "57E66B3D1B0ADA0F74CBA7D510F70062980E7B0E0542CFD044ADD438A0581113CDF941C9F6"  
00812 "840F41D91935B3FE7FEF3F163BDEEB77A706321E6F18DD1B7457D387B2EFD40A5712C58EB"  
00813 "A2BDEA94C738907B11A5C5150BE6EE643D8A1EA1A0A565D559ADBDD3D872BCF8FBB40452"  
00814 "5F791B5AF7E3CFB6FE10E833F19A3F7B48CA9602212959FE244F3D3B2B95673EC9CB8EA6"  
00815 "D356125F4A6F8232D39E95B80F1C057CE08E1BBAA8569A0F030227BA8E5673BEF1B5FF19FA"  
00816 "80A8C2AC57A1953729881BAAC48ACF026D9F8B7D4715E16A14A842F7BDF027CFAC22C3597F"  
00817 "81A065E985AA225EC7F0EA603995973212859B1072E081577C2EF6F6E883EE7F495BBFEC99"  
00818 "0479018457F50DF81CA020D3C7F34A0152684A31E19848A1C7E434B93E4221A77D39061430"  
00819 "FFF183D86F41FF2F1CC96799370173F71CFFDCCEB0EBED8B8207856B514393AE1709DF732A"  
00820 "55983E9FA53AE9AC855B70762F9CCEB49A14DBB200AAE612D0479DE1B09DEB4051C934920"  
00821 "F0BFC6720FBEFDC6892E37D8AD3E46C1B41C8CD1F85B833068E70E0DA5054FFC9A4B87D49A"  
00822 "2B082A201D0F67FCC2A697B4C5E03BF826FDA853C7A7096906B3F169472141C3B355D25F33"  
00823 "5C4FF43596F6B1A7084B173CC50D68F393883673BBEC015A65D9A729D92EA8B0369B465751"  
00824 "30E9FA4C997730B7A8A8B77824DBFE94A58BAA56E798C4AA866366CBB569751C89658280E7"  
00825 "ECF0136BF6592F3BC89D2FFEDFC0504F9AF6F563ADED00D7307526283675E341A8A3CED3BF"  
00826 "371E48DA987BA3C343B5D06F3A1D41E1B61C86A9CD85DA34A21194775259150595D9FE6E3"  
00827 "5B705CCAACE25DE1E669AC6F7D9DB5E59537978CB4", // PRN 22  
00828 "16C80914164E3DF298C8C6BC874DDA6804D47E82B95B17BB0FEEEF153E7CC64E5AB8ED6A4C"  
00829 "7485EAB7E4D536EF3FF2FD6BD69C163B0CC41AE60D8B4A21E1B0B7754E075B642937B63FE"  
00830 "733D47FDEDBA00CB8E9739B47ABD1EE0029E962EAEBA5268F22CF02B638EA8EC73444AA98C"  
00831 "909B19A5A8C5C6BBA58BAF96AB42055799F93D71D631B708E4EF134BD59390F84FEE606DD"  
00832 "04C106C612C075FD31A5542DA5D3CC90F4CDE79942D1CECE5B93DAFBE7B51065D3C7C8C7A7"  
00833 "38B11C2961CD0555E821EB7FECFF00595B50AC4EC507B0FB01013DB5800CD404A53B99AD74"  
00834 "0EDCCC182C12C8062271A15DCA67708091455CDC07105029982DE6A1AC2B94B1AD5BFA422"  
00835 "E496C906E9D63738BC10798453154CB20525FEC9B7F06541FC9AF94678C6DF98226CD62635"  
00836 "6BFFA68ADA72BBBDBF8E626F1B6DF5545B865103F402E54DE75120850F0C4230B423E90096"



00837 "D517E19787AEF96FAC40363469C5D5DF8AEF1D86CF30D10FC805F7AD52555D3EC3E9E0AB68"  
00838 "1310EEF51A2E5922D9740C972C9E531B90F68B4150A39C8E7FDA224256BACB84D0A65D2419"  
00839 "87B9B6A4EA6726073C8E374124497FD1FE998FD268D4F0FB6031FDB3E17863D81C418F8BE"  
00840 "A50963B565E6FB80DC674F5602CAF63F4C149F5FCBEAA29B36F038FB8CC7C9677190E1F07F"  
00841 "D114705B11AEC538A2C59DF951B95D07285D1C67BAE2ADBEA859AE553270C26581C633994D"  
00842 "5690BF4A4721E297D1E5B4E1CC2948564801AFBDEDCD3F3715FFCE54B393E9E445A26FAE2"  
00843 "EF1720A05C9C774BEDF251AD010D12BC4417E8FBE9FE17B8881164677383A44C1167BA1C38"  
00844 "FD8BE5754D4E233D12DCE77A697970C8BFD3EA11C8DBD2C66AA8D9FAD8AC8D5F891A4AB4EA"  
00845 "4335C485358DA5089A3E141C221B0FB0D9614FF15269183ADA3A205942B8EB800964F69C34"  
00846 "CC015BC2779BAE3E1F74FEB81020F1B10267C802178DA05C377A433BCE4F2115A0F60621C6"  
00847 "23F3435F5D02B38FF97BC3816E6CF29AFE7AC1F00C4D8E0BF15A3EAB418A41A55E9AC6B908"  
00848 "E3A894077D6D7515CB342B8502B5F956B4EC258C883E804B451E518E5BB9EFC526CC4155C3"  
00849 "ABB9296E8FF8385D7420962AE89B6973E82AFEB915AF9E0AF6B48BCFC4666447452F0972F6"  
00850 "25909505BFE9E21589C512CC0AD7CCEB52C12B6E4482E69A0C394F453046DCB4E7353AE007"  
00851 "F60D893A988CD9EC6B05E38EA856EB827007BCEBB678CA3DA0737CFA8ACDA7BE04F0690201"  
00852 "F43B9DF94900C801958FF8C986ED9A668FF9352F26A59C333A2E9556145C79F9A61DA5DF02"  
00853 "9FC931F12A2CB0C1143A04307A97A20025EA277C66D8ED1CC0ACC9112BA9836627AF2088C8"  
00854 "322CEBBA70B9AD231EB622300268366A89E0578C6A1343F616EA0D15EC098BE2C93B5A2A1A"  
00855 "B5D49C2DEF7A35E708A5B28E6BC6EBBE823FDA837E7253859A5D915F9DD238D8F858471E35"  
00856 "BA8C0AE9359FC07258F02BADB9A71665021966783BA3417D19362C6046831B7BB6DEDDFF78"  
00857 "C5FEF387743C9AB78E86C1C3639D2A94FFA920EF4B1C0F1E59CF7E36472ED1DA506847EB9"  
00858 "0AB79D09FBA416D46B8D3B172C088DCA4993C9AB7BB775CFEF639EFC52C8934DF85E598B13"  
00859 "60001CF805F99DB4B5F1C2D28049013D4FE693E64A61B1C60D86A5CFB57D35AA8667636B5"  
00860 "26D3C1BDFD1C3DCF977E71CB929591311C8C82029105D32169231690EA44A37EB4C66A0CA"  
00861 "D5169FF9FB4CF9BACEBD65B13A082636A7437278C916B33CD36BEDCA17A09F603300759DA1"  
00862 "78896C84E154CD3A3A527E57685478A99B5199094", // PRN 23  
00863 "7B570F8F888E0526F19DEB03BD80266062810BAD10649EEFD052E9BF70C29151F09A63D86F"  
00864 "74B43AC98B1EF5989CFE4A61B440E01B228C47A89C658503F02C375F5C66DC2EA97E4EB7D"  
00865 "D74393F7186B6066C58477C81C53AAF90D62BF28AF85F0BC689852E5A9016D4D1E381D9746"  
00866 "E65F26FB0DC9E0D4BCC9030EDB64540EBDC953573649B5CC607080E11305CDB6460D26E9F86"  
00867 "A83546057F33438393CF7FA700CA2ADD69DD79432A3EC0F4D1DF72A9BCB7E531323BEDA11F"  
00868 "D0202AD8A2E746C3A28DF68E9B846C8AEFC952EB6720CBC0D23FB503ECA6905AE3661A96DB"  
00869 "B09B63982BDACE71FCEDC2D0F7DA43963FBD0674719213176D002CF6F7CF46A77AF805EEE7"  
00870 "B98A23C31ACBC9DFDE9F97A57B61BFE84B62A8482E0BB0EB7AF54E05C07C3E2C63F3B9FD46"  
00871 "3679ED2DC52198AD3E7439466E7C1C0C5E24D879DBF34A5741FBC03A4C3179802A5780FBBA"  
00872 "7E0FBDF99F0EE6E19E926F8E98C1BE65A38ABC45B273CD542CD10EB90C440037E90E97E4A4"  
00873 "B66AA84458A94117F78131363D62CB3CB62DC32CB9C5D94B7A565B75D0FB8C8AC1AC5484EF"  
00874 "B83F7262B55C19F8083904B70CA777796CBB8EAF62AF899D2CD41DE92B0FC573AAA27C5DEF"  
00875 "62DF29961957E4E75F0100731604A611EB59BCFE6B371D58FA92034482AA43A1C88A73FAE9"  
00876 "A0880287840FFF447FEA1F469DE526DFAAC6C0672AA3DE10CF9D78BA236A7BEF52BF0B8F15"  
00877 "F2A33BF1C4D7ECC16B4C20198475714E620F694637F7A0BB9F29C76404DFC147472BBB8107"  
00878 "629C141A7A69CF6BAD24FC7CB321829477541281A0DF4C2CADC1FB1818DADC8CE4123BB413"  
00879 "564EA8D98DAE969E3EA57EFD3C1CEF4BA51C001CEA267033904831B96F867508F51094E5CF"  
00880 "E6DAB34F7879F57BDDE864DC52EFD32C73C1E4B26EC22AAC6A9156DE7BE2DFE520F432548"  
00881 "E8FFB330E196B283DA0DF20E3E4A1D7A5EC1014254110AE168A22D61AEF83A3D89E5C73CE0"  
00882 "A55BB5E55A975FE534D0CB6D2A51143BA2D6E8A10DF83D97A2E6FF40F68B39B72A347C6C1C"  
00883 "D4F92493E915CC10790D7B8FB5A4316EE504218B02FD2940953DDEDECBAF9CE150405EB2B1"  
00884 "947D20645D9FCE0D83BCD530AE4A75651CDE9C13AA161602545DB39FC747D917F943654FE5D"  
00885 "93CDEF3F8E647CE11167B546688FD319E0DC6B38180ADF97E6150DECD83E5BF132809C8C76"  
00886 "579E50CF813D94FB6AD3F0EE247DAC86811CF7625966409DC99D907316C3FBE4BC26B8537C"  
00887 "C1471846584D710157BE09D7BD6CD46C89C9DAF39DF4EBE657E9035E78E91D94263B0D87B9"  
00888 "8C7E4C09F741F2D64D7C84A283E05CEDA7748B74578D1222215A4826D1680B71DA08A57A89"  
00889 "CEE3CC46BAB909C21BD69ABB33A2183E254C87DCF59529315DF3A378C8CE26DACA3DC9DCC"  
00890 "3C1572AE6A3331F369955CD847C801B8B69BD3B28525B738E5B1091668D83165EE76783F2"  
00891 "D585F8F88A958F19F2E0378DBF0998E5339B39FED7D8019CF773C6EDD985A1248827C5630"  
00892 "2776D5AAADD6A3350D844B766450566DC07774BAE6526C5884BD1C12D8137A2BBDB249D143"  
00893 "4EBEDBDE69D770624F07C69E59B765866EA0D2C1920C21C832AF4DE6CA22147E4D35A41D26"  
00894 "F5CD1661E47C0C559B4FAC070D422EC14A0561852011ECF5F57C2028876CDD75E2085A36DC"  
00895 "463E5C6562D277B10B543DCDC1DAB155DA7DB8294CA040D66C770E6D1551C2CD3D55F42573"  
00896 "5055FF22D081EC031CFE8F7B590C3BEC507D160F190B7C0BCF8BC848776123B08AFA0402"  
00897 "3DB0189E0AAE9CF350F6E7A4522BAA5F079300E908", // PRN 24  
00898 "1969C08944104F0FD3DB64D27C76521D9FE8C35D413E0F30B4764CD45E6847D13A2866144"  
00899 "36C0C373D41B8A00240855A32CF0F2130CF98B5006B86720A71D6D307B64173A2F0ED9C7A"  
00900 "3B5F3E824BA4290F67DF6479C01DE5D4D785DA63D039791D2F0C914E0BAA93086052E74E30B"  
00901 "D0D56675A09243B0E5E953685C90A09CFE6F9B1C77774B1AD281010BDB5F8B333758F6C07"  
00902 "695E005D8C004D2958F079929F1E3098929892B6EBEE479A842FDC51A90B477A79E5023584"  
00903 "3880CC570E6C6224B2CF2B221271863E5BC72D2414AA28ADFB349B53BCDE2A2669BCB73F7"  
00904 "AAA21812D9C3A22D29345469BA48C31106E87CCBC14EA7D603367CA4D04D7CEA77A9E89AE8"  
00905 "9252F6584CDA1157A552D51C23331DEB200590EF5788678578056AEC9D099404D05D8D46E"  
00906 "C70BE649B7818DC31DE449E5BE741319C696CFD99CEDEB776D1D88D7540EB76455DF88262E"  
00907 "3D86D7664484BAF9E6EC2BC51EBEB90C3E76E3236D17AA341B3B7F1B3EFF196085B46390C7"  
00908 "AD5741751DE655D347BED45207E26785A65FD8F8E1408A9B8B58D0DEBBB73468B08BE76EDC"  
00909 "DAA55C61197AD8E05EB70D553A3064A751ACE7CBF769AA5306CA89886AD6789C05B7723DC"  
00910 "B88F28127AB5D8905D9D147F77DB1F6749B61625E41B0D81905AD844CFFAF2D7EFE4880B53"  
00911 "7A0C2C35090EDAB69017C443A0BD36BD0138BDA4C7A1B1E54C8D57416C619F6CAB2144D681"  
00912 "3EAF9FB05A6FAA2FF108940DD5D0716EDAE7D004AB7AF21073B76335D2AEEA54A8660F14"  
00913 "D091E46603A748B4D9F3B66E1FEFF2383745FEA9ADDCC12A12C66655669BE7023A01DEB969"  
00914 "3B7FAF5631FF3C5A91BAA85F7DB720E6189EC9B068066D05A6CAE245D6C107993000D8BE7C"  
00915 "32A9B659A02C9F74471196A3EE4774685BFA93F1C2286B94BDFEF6B653A2520ADA1DCE8A6C"  
00916 "6832DF79F46DA068A5DD6A40587CC8A928690EF9CDB73FD0BDAC4A0B790309B6A27FD2F023"  
00917 "AE7877E57BF70BAFB9AD6E50DC4062B2CD393B8A456CD56FD4FAA1D42AD33965ABEF471123"  
00918 "486DD9ABBDD80CB6E74DB20257D066D5AE23CF81AAC322BEBFFEB9CDEEC9317925AC69F0E4E"  
00919 "0E565EA420C682E33F6E6E4986FB9F1F677AF03612AA49AEA27BEFB6160A2391E44BCCB8A"  
00920 "676E7DF729E4FEA84CBC1E7A24BD20200E67E22E071E4C2753E1211F50F1C79CCE26FBFD21"  
00921 "E9A57F809F193586AEEB62EC6B0D232CAB95BAB7CD79A1F3CFBD113DCDDFAC705ED3D715"  
00922 "C36A2D6DD78710427B6A8C331FC525EDE113128AC69C538EA0D1A06844A0ADD091BA3870"  
00923 "A00D359FE00D74745BD7C11E76AB9D22002918FFD216133373C37B229BB2765EF6B67D0463"

00924 "847191D0413433D7861485BA603029983BFD17DA369B7AFBFC64FBC04710CA71C8EFC766B"  
00925 "8D4745B8F6CB78F00973A1EEE7EDACBA7324910A916CD925F3B37DD601DC6482408E964F48"  
00926 "8FF08F5624D279729AC1DAB38226586F042989A480F236A874F1BA682A367F1636F877B7C3"  
00927 "D6498E7FA951D8A0DFAEE1519AEAD4797D378CE77A49F020B839FF01BDEF59CDADCF21CF8"  
00928 "F678BAA531A4B2356EED839C1DC2A39EE53F19FBE4311F08A47A3091760F3E75FBC6DF3CC5"  
00929 "E9C1F23FC0C209B4CA563FB6EC7BEFBBDDA80E86328613EA74DEEB92000868940ED956AB72"  
00930 "69DA335D402AA54C50A0591EF0C21EDA4DAC57E547DB1245BF62069E715FE2174A9AC9D982"  
00931 "379EB9D422A6C03845D14F043DC55ACF989D262D6458D07A0E198FEE889D3491DEDA7BA25B"  
00932 "3C802A537327AB8C102FDA6C033AFC6BE8602FA7C", // PRN 25  
00933 "512FA9AA09712E0725988ECB3E10F8C2A6AB952A9810C3B48A372EACF9FF0E006B8EB2A872"  
00934 "1582320B3F6EBA8928477A57F3EE089F5CE05DBA6C03F3DD5217D4A6A20E3633FE4C74E75E"  
00935 "C852704FC0C8BC005CB79BC7D0B2EFC25740239E3F9C328BE021DD9549276B2C0C8EC3CE2D2"  
00936 "8E56096C722738FD5609801B8497BA11D4CA2CFB2EC8CA16182FA4FEFCAEA1A8C0D8208FD0"  
00937 "A3BDF519DD609495ECA765F66665E6CF33BAB0F50ED40CD59B40FC120A90941AFAB8222FC"  
00938 "1E541423045D1656C4F673EC9B0B0DCA87D5BD5E9C9C21AB203669751C39B831300F92AAA6"  
00939 "521635BD53247B63F1B3BFBBD1BCC0EF22EBDEFC5C5848CF2A73BB878DF4D247AE45B9DC651"  
00940 "7B2173BFCDD045A223BDACD719D55D23F6D40E1BBF4B4AE36DA16B1A180E8535544C5D29F0E"  
00941 "99A17A763BA55ADE0FC297DDDB3C61D0931571F773807101FE225C6310E1D01E320E9EEBB"  
00942 "750D1FADF4E3CF0E2E19DBC5E7BF273FE72DFB86029139F2CE11082D029DDBDE15EA300D66"  
00943 "12AACD70AAAF0D78E4238CEC645DBB69F7A21CDDDEFDEE1D243B200ED84CD81387D7D0F19CB"  
00944 "8693D36F481EB8BEE7E79B6D05406B76EE5AB521D4BB1A5843D87DD789884DD150760CBA"  
00945 "5FF93BCDF78ABD053E9699BC63D8EEA113FD217BC4967AC4BCCEE02B679979AB757BAA8091"  
00946 "51AD0969E7E723223F5F126BF05E3018DF9F0986F832B50197E45AFC40A927CC1AAB622649"  
00947 "DB0B3BDA4492E4650FC8118B3AB22F97EE61FB3D8D76A7C7A317D2BB1A7CB90E48204DB7"  
00948 "D8F8210ACF33E28C077AF20189CD1EC85B8C1A0B3416D4FEFFB6212617D2C1F7BB646A958A4"  
00949 "8A27787563BE6E94A234F3CDC9906F7DBD6E9E49E3891615F5B16257DEEF7255C89A055C5F"  
00950 "6598EC36872E5460C16101D39086BEBF0EC80CF8C9AF4891FC2629CD14F1A39B80DCE89C85"  
00951 "ED26BF28CEB87CD914BBFD0486CBCCA132BCB4CF88253AD6D5D305425D286532166C3DE248"  
00952 "8160694BCBC20EDAC104DBC8C056D6D7420A0B60E9DF250C8D07E070756767DAF7F24F67C6"  
00953 "1767362AE647E4025AA34B4BE95A68BD54EDD78FB45932FD74548E837C344591C8A039904"  
00954 "D69995C9B4BB47A66C2BF8B659CC2E49595639C6AC773DF17EFE96F312F5F592E54977879C"  
00955 "840B78D551C75A7899A2C93AD600AD79BB06FFA680A2980CC6640B3F6647B416B9BEC68BC1"  
00956 "C6A73115484397004C98F9F0637C2D956B8194EC55F47EABCECF3DE06447FED6742F39C517"  
00957 "D0DEA3BC5E849F43DF36251A75C4C4F723A994E92BEE217C526AEA701D95CF2BBD67FC4E73"  
00958 "DEAA2FBCCBCE10290788658471843B1147DE9B39915ED65306B71E90AB557E3B33E28396C"  
00959 "94992B66F5FFBDF0946FBB1956198B78013F7BCBB3F39A65DFAF3E81E7A058AA21536D45EA"  
00960 "46CC12C22CE15756A6E3160CCE5A8CA2351AB2CDE145186535AE86416BF29E2FF87C5742DE"  
00961 "237D3F52F1D96E10EC2E65A82F8E1A842CF99BFBAE0DEB1827243B6E008998CA082851489B"  
00962 "E74871CB9D9F909F464A14C8DAFE4859D1209D81430C3AB52DA3F5124D220124CEFA0F8B60"  
00963 "03DFBAD6217248E45D05B7D26E27C9922A3F186137C6BA98EB6C79BB20E20B77B16E6E3AE"  
00964 "1EB826E8E83323A1445DABB9616DEBCE266D142A7877EE8050C45768DBD0440EC6FB577811"  
00965 "01F70C96D45D8C5001A49CA5CD852E8B20290145BE748D6244A88C2BD8938ACCE4D68D4A65"  
00966 "82E0846F63613F70C4F995F673A98C0B09C29703D3A5D9FEE87AE2B3BFF7B46A674018F146"  
00967 "6C665CFE0C94933EA98E8950338893E6BC9007BED4", // PRN 26  
00968 "73F36B0BB91CBBE1AF4467EACB75C940A846EEB5AA2B2BBA6DACA0DCEB0E21EB94AF736D35"  
00969 "595390978563CF7400A3C87E13CE9AC82A0A92A960BF3DA6FCF59121672ADA1E834748056D"  
00970 "2F490F6695B95EA45B26FBA8206A3566B44DAD1FC6B41D56FC56F6F38B8FDC5E5C67646811"  
00971 "47443A7268C622BDD44CAFAAC9D7BB4DD6E269D80AE94683EE1552329BE51607B9DE76B0F8"  
00972 "666F88B37039FB02421413DAB547F6E959EAC432021D33248ED0ED2046694D1509E743304A"  
00973 "811FDF0CB242ACC460611652784588ADF7527D0A7810482C8C5268C5DFAD4A997D74A39989"  
00974 "988940361B9E28FA53E7733901DEF53DD400E71D21CFA48CB470B41690EDB1E59E40CC1F85"  
00975 "02C47425B17774B4A687031A839DB2FE16B6936D8104D6631E183F7AE305FF3BA5645322E5"  
00976 "10E065E8B95D26224014629CFAE284A419BFF808DD20F42EAE1C4B041606CE1113365A32C4"  
00977 "7DE0A62F38F581SCC472400EA13E077580D3EA2F6CE323684E2E8B4337265DB11EA530AB"  
00978 "582820CC746489BFE71A9B5673787D267CB7C6AF50EBBC37A8E9C029213246D815CFE1B812"  
00979 "ABEBAF66B7AD0D001765B7BE6E60B9E193FC9F9E5B605240704674DFA64C2029D0455C12A9"  
00980 "D7CFA62F38F581SCC472400EA13E077580D3EA2F6CE323684E2E8B4337265DB11EA530AB"  
00981 "C4C192FDC7CEB5627353A1359213EFC8E9D07B92BC73D36FD2A4DE2113AB5719AA7836F665"  
00982 "C44248F1A20718BFE8465FD9AC11323E090E811530F77DA5DF187CE29C16A877A8FDA753E"  
00983 "51365F0425D771C5B310ED2B5A1A27CA21B9BB9F49DFD9EDD6BE88505EB3FFC1857333FCB6"  
00984 "6ECBB10E104DE9CB27BA32FE1F5B6E3BEF7DBC05605B081FA481079A2D770DE380201DE980"  
00985 "C92E6C5D7FC7995418E708B427408754C549BBF351AC7E165FD2319EEAF0E0B99B5FB093C7"  
00986 "715EB9C040FEFEBDC45D3085A8726FEAE937529DD657C72320A3F4E2D7972CEFC39E8A8F00"  
00987 "76286DE2401A7540833792D2AB5DDE950723CF502763CA1777AF288A157E7DE6FC701C6A26"  
00988 "DB34DDEB7E1ADB340B4C08CB2F3ECF53694A8294CC0E152AA6F81F77E8EB0B902A36B"  
00989 "15239D72C50709E81F4418EDC6786A9D4ED29E6CEFFB236B11C026804C0503EF573EF1A26DA"  
00990 "78194EBB438754B48269B18C43AB771424006438409A4C659349B405ABBBBC70ECCBE922FE"  
00991 "4FF3EEA4E9E9D9BE1B8753D2EDCE0E95A51EA839E370908C3D6F02DBC5FC46898F1A5415A"  
00992 "1F7F7D2A842482E990E6BA271A47535D7D9E1CFA8E8C3741249616F321D624414EFA4F4C"  
00993 "7C7EB5869722B8D55D9EB24102689C7D042B7169533827C72C7E9992CE40EC693500ED914"  
00994 "965DAFF96655583FDF93942558EBD04F57FB8A640C5E4C7B584B424FE46153731CDB72D05"  
00995 "48F6988E84B30C5453B7F7F0AF7C344AA42C430ECB068E2F99B37736109E4D9B6182F2E0F9"  
00996 "5846EC6C796F02BAA87023F5ACB5B53A6B561BFC66011A86EDFCFE457CBD06FCA17BE8F978"  
00997 "F15E8549AB737D92B9A674E560B48D04CDCA58725EF2DDDD7A24DD43A39D15B6A89AC8E219B"  
00998 "D6753CB99631F05EB7641AE249390FC4A4E85699405F7B2874EDEEEEA3E778B2D99F9CCC5F"  
00999 "B1A28042088C609CF7ED35CD0896C7653E317A32D86860B2829DED99DFBFFE27A35A940552"  
01000 "1DAC8E422D49D2415CADB642BACAE22E5A6FB007396CB84CB1952D3A240BADEA3F9EE0A303"  
01001 "C72581E31A56B93183F784FC7A097A348F06515CE89A26CAE99D21AD5C17301935564AB564"  
01002 "AAFC6290EA68626AC0A236572DC22048DA24C264", // PRN 27  
01003 "2D531725771175DDB1061663798BD8B3A0782B5377F7488AD000993DC09ED4C9158D498E39"  
01004 "1C9C750ABE7D50144CB976C5CDD80FCDF1732F3E69313BD4D381BF6AEBB5CC3537A2366FD6"  
01005 "E49FB8E77C106A6585630A67C1B4F9A69D2C5C9F2B7CB3E2F3D41306461507FC35FFE83FD6"  
01006 "8ACCCD946D7B8BF434CE4E837BD0AB2ED0E05F726725AA320956C41A4864981E7866C120069"  
01007 "821F90A24E902D3AA2DD3039D532DC306EC284612E9DAB3DE86C29BBF9C9218BEDCFCEB286"  
01008 "5D955E05084A825CB00BAFBD81DB63F1105B02181A5873010F74583E4FE8548403973D9A6"  
01009 "34C6C1C3D02E04A3D9D7FCCD70CDE6A129BD32D4075CC588A2513B8E2E9458EDD6837CD9F"  
01010 "249FCD11D3304B75D2E1682FCBB87B218440748A8FBDD7E9C7F92ECE1895B1DE1FD5486D9"

01011 "0442052322BA16DD2C8516AD915DDDC9196809DB47494722BCD3248865D159127D216A34A9"  
01012 "72B3738231ECA3665BB683A3E5CA8287906DA73A9BEA1D01C35FC56ACBCD28689A11D10796"  
01013 "B3CC2C1F23571F72C5D4F7E38A9E993B675E4D11FA7DF38513ADD163F31B6CC8B9DAE7CC1CC"  
01014 "9C1EFEEF74366EF346F72B31875A12146447E5CD2C802666F67C8C38CAEDCE4DB2F08AD32D2"  
01015 "310D5A7FCE33060639C14427B04867035DC6F735448102E2683694213C490D24AFFAF43ABD"  
01016 "44046B00286BFF4BC7EEDEED09E17EC850C5F12A1EBE4F923C9F576A77069149D456B87CD"  
01017 "16E8138FD79CE0AF24ECB586FE6E2097B6768C41AD8BD39509B49AAA6FC1092786504DF3D1"  
01018 "172C6C4FBE6741B0CEC95012528003299CDD539E3E5DB4A051FE2DED59FB02699098913DEC"  
01019 "F8EE2DF6DD9081F9384260F21D242B92DA14F998F544835D7E79A9E7B99A859D3D6A05A55E"  
01020 "FBAB356D9F8B3BE652A7D4F9E7B082DDCC769CA69A32E92E01AA2C7A5DA908381D8456FF3A"  
01021 "31CAEEB50ACD72477FF235EE56F8F3DB87EC793960522D8AE35B36AFA6460E82DF18DE082C"  
01022 "E3705600AEEACACE7EA046F99EF1328E2A15EDE3CFE3742ABFB142EFD4A02F2B9FC7F8C73C5"  
01023 "E6A16C61D119C48F2DE667FFD135C0DEDF914D1096D2F0D6D571D26D95CDD13E807611C0A5"  
01024 "D03C4B6E0099783568CF23DF54710281E1F688498A87F5D99F3CB7B226EB32623DD1BC60FB"  
01025 "7BEDF8E9A9A48604892CA3781C4829216955A546C6E03B04AFA167F3F2ED8CC830FAA91DE2"  
01026 "2A575DDCACE10ECA494C10541748362293F818AB2215B31A96181EB2251530E5FA74AB7130"  
01027 "4295064C097D14CCA4267661F5D6CCD956656C73F60C022B633085571F5EDBADCDF08C30B2"  
01028 "147D7AE5485EA85B37DC780986304E2E5ECB372BF0304E0240B20D38F69E652D5A96CE7DB"  
01029 "909131B34FC9F8629169D3661950AF536DBBCFD85C1EF14AC7B19031D2A52DD288E6F559FC"  
01030 "F2A99DF11C0BFBEE556804FF18FFED3166BA5EED0E933FD3A3F41071AA1D62A9216851EFC54"  
01031 "2D15C09CE9D7708AFC6DAE0218C08A237EFA099D08994BC0E8EF5618B7A61095D536038938"  
01032 "70F3DEEACF2A5DC2DF218011F2F63BCC868B668C88402C5648A6D76A6E86D1F1D02D4F4ADD"  
01033 "8E860FB73CFAC5882FBCFEF80CC24A2F5129861E2E0A8B868CF8524CE8ACA65D4D04746CB9"  
01034 "3E0E61C2E965C198435796B9E19CB2450717105B9265EC4E334A86EB273E03F646768E3C29"  
01035 "975EF12E719C0BD7E5E024C23DF3B754E5DF6CB3F8028E1CE3B2AF006906FC074B91CB7BD6"  
01036 "5C73903B922D7473AAFD65E79621D0C287FD28B8698D6EE00C05407747993781C7EBBC76F"  
01037 "CE86663C608516564910AA3942CB57037E89D3A6344", // PRN 28  
01038 "EC83904B22E1DB25A4BAB428325537273B908BB5F85BA90A31E93839FC2477BDFC873CAE00"  
01039 "3AF61C40B3F1ACA29CA2043D4825DE28DD691645E3C322B88F5466FC52F85EB5B09B631569"  
01040 "54430C9CE672263A37A2E1302ACCE514D29207844AB02AA23A7BAFDE44FF4D555E36948F0771"  
01041 "F52F1AFA6F8BAF1D69BBE4AB1573CA09C5F4852D16280335FB4FD1D4259989E893F4A017C9"  
01042 "C5013182627665D7BAE2FEDDAABAD4EDFA810D0BB1462161F01457FFB24F5E8C845237EDE0"  
01043 "82F139653F0ED987F9BBA3ACB4A7424B721E8DC8B0E0C824D4477A6680C405E4F92901C272"  
01044 "6C43921321208F08B29B442F01EA554B1ADF8FA04F0EA9992739DB100DFD29D035522CD4C"  
01045 "F946DE29CF88FE047F43CAD5A141705FA6DB068B0BD83F42EB7D0EF8EDE884A4975FBB2C02"  
01046 "B86E15FA4872CDDDD1D5A7DBB6332638A3B28DF5F5C446074E3CD95360EC01021F8A94496A"  
01047 "9FB11C77B2FC8C9D14BD67201C54CDDE046CEE8912B6C17A0CB99C3929488C9A51D048900"  
01048 "809E1B66BA2B872B6C421195BA6A4E38665DCEFD8A6F6D6E5A9C175231A03FB31EE6D2DE4"  
01049 "911D6301FC11727911338896ABB2C44B3E136DD547BE37C5A367DC5A0C9D4F1482B52FFB7E"  
01050 "B32D4749F8B1D5FA808636B675582B58EE1284305259578867AC6DD343F5A58FF56632B707"  
01051 "A0C0223E533C1D2F0862921A588C9A43A1EA927EE8FE1FEB8C8B625EEE9165FC14E331E8876"  
01052 "8F4E26BE029287C5F317860DD5F435A4757DAF83ADAAB3C14339E70F523718510F3BA220FA"  
01053 "5FB884B7C87B1F24129150C0D35B3854A341C76306092CACF7606D24D419763E2BD3B6D3E6"  
01054 "50DF6D608F753E6EA1F7EEB8117A2EA248D8ED2EB51C356C9B871925257E64D729D519BCB67"  
01055 "3BE0F1D23077BCE147BAE7BF6DC26AE81B823F34529569AC48F7F7D161E53ACB246AED762C"  
01056 "2EDE7D65BF395EB4F73DA983AA5E949AD05242162F4C092B9E974BB42A8E18237AB5696AF4"  
01057 "0DD86F3C94BE8185C10B4BD1236B4366C85CF309D5B277ABFE64265C5E8E623FCE2854981E"  
01058 "D73C42BF6CFDDDD7D14570B6279F776116BD3E6900C5BC507767792829389AFF4FD14C532E"  
01059 "55A74A7F52004B3122155EF078CFDAFF0BBC1A141C7C78E38AA175AD27B8A0539BCF0DF0FD"  
01060 "B8E3AB2308916AC045938905CA6509CACEE4F771F4F0534F409E8D07D18E1DE49102E431D3"  
01061 "9369DF8AD109638D2E1DF7C00921C7EE29A26DDC1BB9C83B151DF16BEE4CE6241B23BB4B0B"  
01062 "2F1AB0FDB2BF92CA538853658BACC738A594532823088550F6E2AFFD38A0612FD94DCCC8EF"  
01063 "8866E6F668310996D681E466881E8F2F5E547513C45B7722732ACA968D2D9394CD22F32CE1"  
01064 "E6AC0F73D91C85152D44F796DAFA3D47D7EA7C33C02F6ADDDDE8FBC66BB6FD107FF715836FF"  
01065 "E03E62EE9DA9305FD1E4FA10632FDF8BC58A0ABFD11105E8C21AF37E72E7323D9912CF1398"  
01066 "266E970F3AC2787998638ED86F0FB7DF8F9DC2DBFE6B6CECD3B679D8C8C2D9F4F6FEE3B91A"  
01067 "QDA0729B0EB81A041FCBD5A39CA6BB6ADF8A56BE4539A56137E60E57A6D7C0F449945A48CE"  
01068 "FBCA06746A39960BC24BB3DDC06DD078A24218271740D10AFA5FCDEE923706DD125CEB6F52"  
01069 "A86F4D9A10B6DE70A171B1801AE719D74A31C1F3EECEA312D0436E3DA16FF4D2723A394E41"  
01070 "53D08EAC9950DDBD75ABB77D18E789AF8A17A88FCCDD530E6ABE74CE034F34F33B88FBA1"  
01071 "6BBDF1C54B0F9C11EDD1648478CDFEAB1CACB493A30609F6E76F7A01A857F374C66034170E"  
01072 "817E6330E8EB1B36741ACAD95917486CE0887FC090", // PRN 29  
01073 "380374D28B44F31397486F759F9ED530CCD71F0590BE4721F98C894AAB422D58CACD3EB2E5"  
01074 "6F1CE283C122A91A45E34CA340CCEA1D9ADCC6794CA074D5D71304DDBD5C4540D1D1EB18C4"  
01075 "C8A19BE16107B823C6272955F1A4DB5681F8320E975E28FABB12F8BFC710C97D176B4B35113"  
01076 "3D719BE6570F2FAEF09850CBEA485BF7E5D7F37C180DA94A9566A1B5C46098C68A08BDB1BE"  
01077 "347CAAFF3EBE5081324761746DF895F1150DC7DDB185A4E52804ED72B0CB1C6E508509A1"  
01078 "4A9E68072A41943C3950E018644CB8FECDD125053A88702A64DC7E07DAFC226C23754EBA86"  
01079 "3940DD64456C6F7A504E82998B61BCE0FEF250706938EC881DFB6C6D0AAF82AC842BC717C7"  
01080 "47EDB0157FC9B9AF5C2FCBEOE384A8C21D34C06061C88588DF799933521FE8B0623FFAA6AD"  
01081 "72991A4E41CD1BD1216A4AE799A129DEDA8FEBA5F0E34E2C4BF880DD319EA467790810B559"  
01082 "C98B44BAC195AAA07D7D7BE429EA79900B1AD6B6591CADE647F186BE41C9A0831620C63800"  
01083 "C72EBB52B45C5DD6B237246CBE84754029C2E9066F62E8E814EF6A679F6559723741539B12"  
01084 "9A3BD1097C65AA31A7C7A73B678835F7F22031A1672239EC2327464677A86EA701782910D"  
01085 "A91BACDAE9952C17DF09586604A7F978A42438A3C96CE6AC3040267EBCB7C4B66DBD65FADD"  
01086 "00FECC58CB8DA7B31FF3945B442B6EDC9BE5D72C47117EB4D3774C4DCE0CB369676C4ADC9"  
01087 "30028B9C5131661D826DD57CB5F77FA319905B7DDFF947201159D59528343E6686EEFCDD6"  
01088 "C000AE170F94E78F1F0D1C5C2176F47EB8471E3D6EA4E23EE8363E3CF4F7A187D8F0085308"  
01089 "29673109D8411855ED62F7D5636FF00D462E05B4834D1DEB26739B772088E183AD9A572CDD"  
01090 "D5CA49AE62397E369709299C53B3DD18A21F3A6AAA79D49030657226501154F804306E09A"  
01091 "F3551EDF13DDAE5DBF88212F96CFB7AB31FC53B227F8DD858A434BA1EAAC9C7148F771A3AB"  
01092 "E67403FAE005FB72516E2823AF3664BC62820C2D647F91F3A3563CD4042374CC1B0753A683"  
01093 "A1726CA900F9989DB589619D4E6BCBD3EDDC5C02A8CCA0B9953917E5E30E6FA01998ACB54"  
01094 "93E751736B4F61B5F5A64DC284662DA7BF3FB48AD934A42BCB005A11D81370DE158BBF771C"  
01095 "C5836DC7F2983337E42694988DFB59B36923C5989D258FA9F1720B0BC6317AC9F659E3B48F"  
01096 "D831323CFD3A5633405EC1EB1B5D617C18DEA13DF44A76F07001F3CA6E64EBBE1C9B319FE"  
01097 "2CFBC68DDC8A723840041EF27B235F7953BB65C6C01100BD972DA6A1BF097526C89AD4AAC1"

01098 "F74A0F2E9CCA0BCD83743A963CB6A1F493B8CD5FB468E659D25896F2759CCE33E6D1A6F2BF"  
01099 "5310033986E0FB13A66A13703D7453CE3A95B584BD67113D556CF793D4BE424B3B13E608A0"  
01100 "A78152C370E37F1385C737805BACB654F97DF97A4E3C1E330A864393165833C00A374B9CBB"  
01101 "A6AE1EFA7473BF7FD17C95DBC6597E66CD9F5BBFFAABA7D93592A425CBA62ED5B2BAF5DFFE"  
01102 "59A83F4E3E9F175F12AE98A15D522C5861E30973EB4141841E8C3F8D38A9491E27165B35AC"  
01103 "87CSA82DEE88F5EC8E34105558F137954CEC51138C0670F85762BE7EB5945CC455807BEE60"  
01104 "75A6EE882948D5D392281BC134A9E540FD0EF98616964F01ED2FDE29C44EC6F6BA0718AEE4"  
01105 "40D6D273A0D726D9FE0A3C20E2D5109C472FE65BB155B5A69DD784D25C8CE83F8A38BCD23B"  
01106 "54FC74FAAB1BAE3055E4E470514E7EC5944DD8B3F61086A30771A831524B6AA1E3871BC851"  
01107 "D6717938431F86E4DB31BC0DD663C8267835F87934", // PRN 30  
01108 "46B4DEDB85B3DB87C21C2088EA9AFC4A4D27A43DA786B758F04AAF815305E17831DAA5C594"  
01109 "E3DE143326A607599169482BAB40DB0AA54AB29536DDF5F87DE9EBA5E799433D3727B73EFC"  
01110 "D1692C10C455B0403ADC46CAE3E75A47455D26CC5A1CB0A8F4B8FEF2B8675408D4AD880D4C"  
01111 "F5AFEB3A4EAD5191851F6A9DB1744D846086B6AEEA88EECBF77477DD8212C291473445371"  
01112 "A2F1471BDB8393BA4CA488DED41DCECED130987D7AE96627C1A7896CA61BAFC0CF51C9BBA3"  
01113 "2AFBF2F4382F64C271EE9EA3A44FF8FDC6D503EFA29BF687E26ED886BA0669F57C41E15C51"  
01114 "0532AFF0F07482356F0A77AEC6A10251C0F5875CAB605D7BBBC4BDA946E1F15A737C38D8A"  
01115 "E4BD4943E17253BC8D82C7430DCC05B73D011E97D37805946D54C13C53896FBB9A0669C51"  
01116 "5EEF13D0F79511E3B1F358838731D98C530CC5CA6C20B39E0E284573CCFB106426CF5EDCFF"  
01117 "05BB0F877FA465FE2170C3B8E9F3CDAD14BA1429A916B4F69BEB46C4550860B4525070282A"  
01118 "C4C6911A2BB3545ABBC7BCEFE4D139AEF824564023A98A601437AD10DD8DF19577B1F9047C0"  
01119 "8BA199DA243DE8B133325BF5C39FFB80AA6444C916BAF27E9E20EC0B611819B47B1AB0DC85"  
01120 "79A7F2E034D90E54CC2A551C02616E3AFD57B78D3718D7A29C0BB17D2AA14A1429B3B33E1"  
01121 "C3AD7F0939BCE6969201C107086ACEEDE59AACC515F4CD7F80419CEE8005A856E4C4DBA3"  
01122 "EE594E28FC6FB5717A59098AB431DCE9B0FDA2ABB0190412BAF0A8DCD77E989B66DBAEB824"  
01123 "78FDF0F37A05FF8E918EEF227F6D4966D0897053AE6537C98EC2DA1E9C642F724B500D8D01"  
01124 "81F1613C8C43A9DBD67BAC6214E3DDCB8BFA830126F4F84E539DA744B49AFFCA15C19AD66"  
01125 "F8CFC24CDBCF2D80B6F3DCB43280A5D0977BF9EB98F31DB933144525900770DCF8AD3CCEE0"  
01126 "AFB4673F644F733A1D8E3A400F7C42B860A0069AE8CAB16D0EB598E692DD48232DB5E37BBC"  
01127 "C4D30C19EB930C36B8D9A63AFD2AD6E9026D8796BF190C576CA311C53FFAA55EF42E0C3D1"  
01128 "83805E451924E56F88F561A30E0C4725BB1AEC9BCB283BBED3CE5CF14680495E160D6DA33B"  
01129 "6C84501EB919485D12F0625A6B9116B3C04EBDC97EAE7659F1B456DE06BE24ED6831A2B2CB"  
01130 "B71D0EEDBDC1F8FCE99649AE962A7F7E4E88FA01B0125FF307A20D8ED9A461223AEBD86D2"  
01131 "0E05C19FA0A0C03BF6E69CDBB149A78153403D9BFC1976242A0E04D6BE04919D460111E974"  
01132 "C4B291B666F41713923194E643E2F60EFA566D4924D0398B4D6EC9EA8D275D0AAB536898CB"  
01133 "CF3A877ED1CDBA31B552086FE54958549C86D5A8AB6CC5C342EA04D363A85AE55EC0365A7D"  
01134 "4961580AA3CA7524043B0895B58F6FD508EC9D22CFE664073C71FA9539405C282ACE1E7BCC"  
01135 "F1B49A17E0B7CC9F6651D32D527F50BD85680718ADE2DA27D46647111DC540FE2C8033146E"  
01136 "6AF8533B4C17A5E4D5949C4762BCBB3813899FD1B6368943A57AF41F5A4A8302F730659F"  
01137 "1CB02B009F8B3B334B0B4996F0932E5ED48D84E28FC0001E015FEDC43272683323EDA06F67"  
01138 "41C20381E3C9A8B2A45976525D5C2EF680206BB8A0FA682E2832D7695883F31352C2660737"  
01139 "B71D0EEDBDC1F8FCE99649AE962A7F7E4E88FA01B0125FF307A20D8ED9A461223AEBD86D2"  
01140 "7324092E0D1CC4CE094DA8908A30FEB78F4A5728E633EF43AB9D86CEB9D902C14946FD783"  
01141 "8DBD3C64A3F29A051251344CFE2D28A8CC0ABA9219258C46663B203CFFD7883C058846705A"  
01142 "16E2DCD495B7275E56B518AE13ABE5B7B2B5ADC1B8", // PRN 31  
01143 "6C01D916F94AE76996498DC689DC21382F033B000C775A82FE7555C55A4336EA42BC13E11E"  
01144 "1E57618CCCEA206A8A03A190FF4706C676502A7FB1EE409380702E302A93DADCB2B6563EEC"  
01145 "614309567A8E2BBD3267027D140ECFE30F295AF75433FAC6D8B897B255FAEA971AE7F69166"  
01146 "B9A9B8ECC4CE13E8227A2B0C48E0418C5EF5947F8F1B077F2132FEF89B41F702D966FD8132"  
01147 "483B241FA069648428D358C31B2AD22EF4D2C7AA84AAE5AD4F3E2A299F6FB52B350000DFF0"  
01148 "EABAD65AD5FAFAE468970254F98B73F24C9CDBB5588D8F4385A0EE78E933C95EA35C76FB6"  
01149 "17621116CEF94451645AA57811D06E2695691941827EC6E3C5699CF7AD4336EF26A791848"  
01150 "7E3D4BFBE522D9FAC5C1DE68E1B92DC3BD477EB78F3B2C27175EE8E89063491F2F99DA575A"  
01151 "F8AC42DBA27CF991413BF5029F81C4E9E8C4D6320B931BB6316D2E593972F110386701782"  
01152 "C32605499EABE557E86810B7B13BCECC9393F17933E0D690C47E782F439815678AF96BB122"  
01153 "784CE63D4E4F1A74ED9E026B8582331D044F3CCAEAE05B2EB891B02EC27F0A56252BDB16"  
01154 "B6315858D3D10D1D852AB5E4B82E6366DBF1C14D5F9F10549326E345C1DE94D83851D3607"  
01155 "3A1A3809218C1DAB9183AF0F605A382DA21B4766910004C8D7D1481FEC6FF27C9EF80B8E4D"  
01156 "FA4C5499CC9E67B4A8E2CD42C78F71D5C0C6CA873BE1F709EE911C1592BDCD8C68E2B6F4DD"  
01157 "C012C2B7EA4401940433B24162865D60FB02D26B617CD95108AA582ABF8D64C7129992435"  
01158 "049481AE803796161DD928E9979AAF636D2039BA24E14E7B89D2B7257BFAD10AF8B5AEF678"  
01159 "8FC731A7D9553AC47F72319F98728B6B758BDE633FCE36F4B2117249F5DE167A8814381E1F"  
01160 "3C547DECB751C9A83E711068B801AC6A317F1840F95FB4196B2AD7C3C7CA5B254D62205245"  
01161 "82F25C6CAB9E4E77CDCE15EC40FDB77350362AE60145451C1DA6041B2F088105BEDD3745E9"  
01162 "637D7E1D15B7F1E234BBC1CA693E28B455246E2D972391F496ACE57C91D4D5B5049FD41B6"  
01163 "2E165CC9BD02A1F68F515DB7AC6C8BE5C2A13F75A5C0AD56C9731F191719AC47FBE61B1F4"  
01164 "030823C92EA1A254A22F60F552C7926143C4C3B8D49A8C647748426D3D7A209696293735877"  
01165 "822B2723CFD0937F6A9E5789FE023EFD8E841D8A478A215E74D747D185E1F0B95D87F3E52"  
01166 "D813D11B2CA59485A6B9F42EC44183675937209E3EB1DD03FEB96A3013CC7DD0E24353FEFFA"  
01167 "73E19926A206E85E6311EF00811EA6F83CD137BCA2B01ED116E9BDEF46EECA8BE463BC741F"  
01168 "B3AF8845B93204211A581EEF20B0A5BB39CD436786201E81CF1220BFBF8CDAF8C6F7A0AB10"  
01169 "6236ECD19DC74219844AA0471FD411A093148DEE650C3B554363532A414D08B55660212F89"  
01170 "ECD0DC1783173040F1D73C6B484CCB2635587265745FE32F4E42A4AC792D4189AEED7E932E"  
01171 "CF45827B49696FB5EF98135250B7337E5C50ABE2E3316A8B3CE0D3C1761FD7161D7832BF67"  
01172 "B2CDBFED17968679534797A475CA1EBE982F36A8CF1C4DDEF15A64CD0D553DF8BD7A98311F"  
01173 "5C26B4D80B4826942358B16890B3BBBC3ECF08281C7DCFF62EA27DC1D357A55728CFC4CF"  
01174 "75507FOFF9AC1B7ABD11653BCE119D90B8515D6883EF14B749547A2B908B2ACB7726010A69"  
01175 "755356528243C4C199544BAD1F8174E177236B45B4949C206DB1AD27BC3C3B1B52B5B717D8"  
01176 "F928B6DE9A2711E329ACFA89E1F16AF825EC0E0E97888F759190753020B90F4D704B2EF364"  
01177 "C9543AF699206960F0F339ADE71DB75D91671DCAFC", // PRN 32  
01178 "0E0BB666FB0BE2A72FD69E9D35984BDE044EDFE05515CA4CEC9B5A1D6319DC4C115D3291FA"  
01179 "7EFF5A972182CA148828F9B43D9A3664FA5A0D2F0A787BF5298B5E6CA12D98605F09E8CAC"  
01180 "C18C06BF93A3ABD6A2F18000E9A3AA00CFC8CF1F49D55D3EE0B943083765B717CFAA30718"  
01181 "B5C28744F045E563F591A8AD463CEAD0305BF06CB8D90C2FE1BC929DA8CE85B1A673E5270"  
01182 "4042515949BB706E68A11417C9BEC9370DBF697C615BEE56ECF4721DED98A0CF07D8B97443"  
01183 "C7C84D7E4F48849DE863ED9009AF9E8C3894B12C1795E05F48D02F589EFFF280ED660C44"  
01184 "9210A1B38AFF929F1514567FBF0C6BC6DCBBA3AAD6263128FE04C82D9EC2FB8BB240A55F68"



01185 "9F7AE8EBF09E89EEA815055A3F9740F2C9D4EC812F3D33706E5C06F82B676A83B21EAA7A6"  
01186 "703B4E25AC4F461AADE0198E738739A7DC828151217E8CE8BF14D3E8454D9B9DAFF977FD"  
01187 "09ED7048C7D9C5B3DFC0A1BD169A2ED184C527E042243ADF3AA1EEA77C4123FC77B5B1C3E"  
01188 "6B888B1369F32E30F54A25B04DF5BB4277E714798B5AD040FD755E340824F7CD925DC5DE2E"  
01189 "9E299B701A3C48C28DE5B5066A3033A7DBBF86B5B715BFD8BED11B8154408C5781CEF8E9FC"  
01190 "8C1042B901494A576A04750BBFB13E0DFDEDD9715807F7F44962601D7FE8E7FE7CD4CCA6F4"  
01191 "A8FEE7D88EE07E1FB5CEABFB731C6F6120C8A89DE4BBD00E558D3B71895D357FF5ED081D73"  
01192 "54379B290B1C81B7DC63A846596B238FC198D1E916616576C2B0ACF0DDF7C934D27C107A0A"  
01193 "A9D002F91F6CED138E802CA9ADF6C266CAD94B34C1C2C45B1165D08D9DF0FCBA5261B1C38B"  
01194 "6B3848F68FE003496407F8170A0E7DC4B23CB65BC02726D1A94F3B846F0684517E31EEE1BE"  
01195 "0CCBB1811CF724F821062324D2620676FFAAB7B69E49E32D414D51D5804E5B678DCAB548BE"  
01196 "C889B04E468C91A86B9884C5910D5B4BED1F925600E046AA6B5C7B2206F2F1AC8EC47A7968"  
01197 "F066BAF5B22DE951FF5EEC7F516FB70D8B74089BCF942C12CA0661639FFA0454FB607151C1"  
01198 "45CA5A586242D6CA5322DD202513D347522439CD3CD11E30BCFC9E18D6540B8B7EBAF30A43"  
01199 "5570BBAC869F8D3D0DA317C0B184DFC65AB2A1271733F0BBA0689AFA1FC97552BD188C08D"  
01200 "B532B70E8C9BC811AF2F8ADF8C7A4A59F4404B940F1BB9AEA3DF987942344801FFD875DF43"  
01201 "9A3AC83A263DE7EBBC2921D619A54DBDA1C3F0A5C58E3D6A84E93159DA34F120BD387AD3AC"  
01202 "F89873B12199AE654406E436833C03CA5AD9A2085EE47ACA614697A6255E74FFE17974F9B"  
01203 "8DAD2F61DA7E39423A5598F38725E7D0E044DA19D2DBB2DB8EC56FDD1BE2BED816607ED4C5"  
01204 "A702BF3C484B283DC29D23A1B78EA70499A434DA240786A0FA279EABA37BEDA023272456"  
01205 "5F466FBA00D59E60B5406E5A1E774252FBC2B39559D393E571FDB5981434E079AEB80A01A55377B"  
01206 "9B0A7F0E4D4015341F232B6568474A779A76601AA4FD4A702B7BA4B0FADD21A6FAFC9FCB3"  
01207 "38C937069BA79146C5337BDE084B331462C04EF29B0B2038070642F5149D80280454AA4D47"  
01208 "F2F52121A1A701339C33AE39773438F92499FE4BEA04055DEF15FE22090D27E0A68D357327"  
01209 "BCF61D3EA6B334D9628973CD6B363BBAA9B79574C25BFB2932AD98E40606C72F397E77B836"  
01210 "F5AD9FC040045501F910B10A4762F7C5F8AEEA911FF040816CFD1F25DC59A1E840DB49966A"  
01211 "82B814FC911346C92D8A1011F277C77E974A8F6E6DC3162AA53F93E717CCB7DD9C553C3BC5B"  
01212 "8DA10DB7999ED85EF724156FBFAC44FBAEB6D986B0", // PRN 33  
01213 "2708C751F8CBF4830B2A03CA1BE21737DAEA8728872684AF7F4C3AE960A8D104475FDA396"  
01214 "967BAC4466FABBC0EA4DBB26E58D9CC3F5B9041ACD3CFF08DC81E7FA7847B9698E4B45EA31"  
01215 "6541F392829872083B05C5B38CF188F23E10952C896F68AFEE35D4570F800DB755A142A3E9"  
01216 "00881A897F5D55CC51863C4A2E53150E02686D50E1657794D18326DEE198972A132499C8A2"  
01217 "3278E558665DD3A5BDCAE0A0C69C495014417A7FBB7DC0BD40EEB05699772164C70363E83"  
01218 "EADC06AEAB79F09BA8BF661789E0721A1F9BDB68494F57032F8DF0D329086D3F5765455195"  
01219 "D93DAFA860E7704033B39EDD4F86838F8B8019D4B30DE31D7410F0EC37B55266BACF41E951"  
01220 "A9A0A884B03BC6B6C6E299E991F34C7FA4CD03CFA30FE511341C19907B16799623D6F5A385"  
01221 "EAA8FBCB78DF2C0F57C6610084FFA69EB26FBD3F9C204A9F901B9C28D45E7FE2B5098BF688"  
01222 "33876D45C1343BADBCD17C4565F67AE44C11B7B4F1A8D74947071739912D6E2A3919B8C2FA"  
01223 "7F566B17EBF8B343CB6C92C8C5207714EC2ED4CB4E7B4C65216AEE46BDF1BB6A501F2416EF"  
01224 "CE5FAC3970E8B4A6DB55FC83D05176BFCFF127C29B7E976D6FE6B640564639390C5F9C68BD"  
01225 "160F916B4843917647008EF0E57AF80A5641D4300559DD992F6565B066882B32467FE05AEC"  
01226 "7E81B7B1990D69B062C9E9E6CA4BADDAC09E5A14C994ABB007E299F2136386F9F66AAE9963"  
01227 "A6DD16FAAF35553738FA01E3715D107D1AC57C22F47692315600C41533DC0151630F471C50"  
01228 "4CE648FB0389B8F970435E7AA84E1B27824743DDF5F783EEC1597FEECB9DA4FD30405EF681"  
01229 "E859027E824D9F6ABF9E1F74AB205BD57174960F3619DAB550DECFE4C42974ED12B8960A07C"  
01230 "838891BD4A95ECA6A01A59A052D19F554BCA488D0FB7C30E371C5FF02A31354EFDAB54B1AF"  
01231 "E6904C53E0FB60D5F0E5155811D2F28D65D57E335D5AD7BBAD017D5A237E490DDD6BA86FD"  
01232 "F2F52121A1A70133EAC3CF93012AB8D2326D93A784650B2DD62CBE80B2F74B4CFFF012B69E6"  
01233 "CD10F9171AB683C91C55B96CC56131929492B346D57AFA1D4476F3D0D04584038F66F97A13"  
01234 "64F7C54912E246965EE6813F6EB36E90649E68D7A9EE84E47CEDE3BF1B36A351BC1A3737E1"  
01235 "A1C7D74F063431E6A09B83605A760AE6197C640600FDC093C58F9D5E4A1F4DC5A65624ADE6"  
01236 "FF5C4ABEDA0EBD1E4E452BF3F98247DBD07C51B71C0321CBCA7F96E1BC5161AA245A98E84"  
01237 "B22B4E739099E155E432EA4159C1DDBF6F20A6E66396083893FDDDBE7C6B160955052B30C1"  
01238 "04AB805F68B3264885D18875FFFC9F12A1C7D395CFF3EAE00A3E1110FB9DF8DBE5E15330C"  
01239 "7482895EAEF7A8B537BAC31EE295B5FF274A32B278044DB01C69BCAB1A872C01EB8ED86637"  
01240 "66DEB4ED4D0F3E5E6BC77FA0E27B85B6920A71F712F9778F968D6FC10F9A6E01F13060C1E"  
01241 "6CD23AD468FFAA78B20178C6899F3B0DC9070019EEFA269DDF85D6E30E3C73C6C866E5FF95"  
01242 "0F40E658768B0EFFFF93BC31F0E03E882E5CBCDF57213B3EDE582A58DA0F9F7E574C8AA390"  
01243 "4B06029F2149788B7B32812B898BCAE538DC8A86F02DCEFD384A2CAC422FB9E2AFA8421D8B"  
01244 "4DC6D6F24109DAFD3AF52183A9E1EB096794AB0575A962B437F36008BB3ABFDBBC89464C58"  
01245 "C68DC546D682888A9AAA1DEEB78BBF707002D6003CC1A12F7D59179108F7B53C40CE89B4EE"  
01246 "D08A56E4584B0066414C64AA8880571BF4B803334A19BCF45A1B222AFA9320EBDFDB99114D"  
01247 "7E7422B0A703AE10D932796BFAE62037BF9DB0720C", // PRN 34  
01248 "265B55605ED545130DE90D99D14322D3B9621758F5A4456A1C5372FCA1389D780F9AC24962"  
01249 "B5183D68C702F6ECDBF485A4B3F074AEB2DE55FD592296F9103BBD8F0F7E1512AFC84005E8"  
01250 "AE18255363FC62320684A053D4A0E7438E73B098D1112A40FC62568545509302E3A4908943"  
01251 "2776A00B6BE015340C1A27D6566F91ED110D917307750A1F1C389E1880E3CF59D173B822E7"  
01252 "303C90939EEE42F161AA9334CCFF11E876F83A8DF0012F70FA1DF7098DC99A4ADB690812D4"  
01253 "8316FA7BDCCE50F4ABADF1E89661D0D1A51CBFDDA8A3860FB223DF3746304FC7B58981EA5F1"  
01254 "A00CB707E770A1AFFFB8E58FDC55968778E417F4697A2C2088E197277143A04196F418B191"  
01255 "D6C461E593F32686E49FCA45E032DE38620DE99A1161B591924518A049DB001F14A9B57AB8"  
01256 "22FAEDB2144BD5F76F135C018CACAAB275BBBD42B9ECE505C7E8F7B6ED6735B8397F2C692A"  
01257 "491CCDC59E6CA4AE57471D693D0FA9FBA4F39193C853ACE2D54A31E83951C56565FDB12A2C"  
01258 "F9079C33DFB40431A3A81CB747C8BB286BE9F58F15EA1AAEF77905086F574317C1A3D5EA3"  
01259 "476DA37108C1F0674C6232E22E885457AD2659704178499EC8D265264E63A7E614B83346BB"  
01260 "CA21F167925D1EA6264F76F5E9F29607367CDE50E5427FA1746D70F5C55D5335F2F3DEDDCA"  
01261 "0BC8CE7A6EDABC73A44BC31BED3F06101D2B79C9E9860D9F660CF0FAD4E7F55DA4BA78DC4"  
01262 "B61E2C09135ABF941B13818719AD3E7456F2FF13BBD997B7CF26007D6DF31C3FC51E6695B5"  
01263 "B27DAABBF4443D01D1AED3741DA517E7CCF3F4D31DFC147DEE3DE18F62440C041A5E28277E"  
01264 "F2785C6AAF4C1458023EBDD9F9BE7CEA1CC7B032944131C0DD7CE236F75948784E0D6C66E"  
01265 "EDF3915E1B30B8105CF16E507348CA81F56EB6A85CC933149E26A6AC5499BF7435275476EE"  
01266 "1F2F635E3E7754B2C7D27538FB2941F717C44A7A9A2864ABB83F130F393965DBA7D82B7DAD"  
01267 "4B5BFDC5E3A60C075FA743E076947D873EDDC4F3143A605659E1D41DA24145930428F0791B"  
01268 "B784C98EA6D3897B7D952029A1F5B02A0F13534DC7688453EFE1CEFEA0F9743846084AAB10"  
01269 "69400990E8BEE387472E6A658C72CEB0D3DD4EF43C7B6453581E86A48B152D526F995733A"  
01270 "66CB7E415F1E819165466ED31A46D1C8FF4F9EAEFB13AB11C3D344F33063BC948EDB358"  
01271 "FF436A8D1C759AB2F049A26D474281335BC04943261239D6CF13CE91AA45A0F38950DDFD2F"

01272 "A5E739880D13DEED6093398C30F8C0E5694749B0C98FB6E59D1F3638AD6D6BB638C85D29EF"  
01273 "98DE6CC32E8FDB082D895F223FE966BAFA1E016FFF5558681D06CAFFCDD00E2A52ECB9527D"  
01274 "CCE00920963CE1D3CD7ED2B00A3023782ABF3EC31791E544D11F9DF3E2CB1DCD4BC4D8F2D"  
01275 "A3D4A88D99B4EAA8EF808DC07EE0E89725AFE8E006EF466B48946ADD37DE462772E89EDAFC"  
01276 "75ABED2B942FADE7BE7AF7449852735480178B03A4917A9517D1B7EE3691EA545D21FE2EBD"  
01277 "82B2225E88697C27BC085714A7D0A06A8DFCA205810B3D022619C73C0EBF5367D154CD25B"  
01278 "39508676D23DC6818403F0C4EB81A5953A8C8F4849B2DDA6840E464A69625D02F6CAD2D399"  
01279 "A3C90ECB1BE37CAEDBD45167C949D01288FC417B47823A90F52E50AA373098ED071F70608A"  
01280 "2E1613A6507B1D23DDC62AFAE1746B09992653BDAC57235280F472720B22633AE06026A1D8"  
01281 "6389D432D4FAE6005C19F2D484EB68B3C033F5B89DF23F5554C789CBA7630B929B4D0EAC92"  
01282 "5743F3F16A1CF7D3D8F1AA29F29F5571564FA4984C", // PRN 35  
01283 "A68E1CB79436B0575C3F49F9D7792D1E401A79AC94077CD3A75075AB7CA34EE886388C72B4"  
01284 "FB982A10E1D62E98D1AF55D03C2AF5F008F240543B4B5975CDEDD3EAEF0FB35D2883C1A96BE"  
01285 "3A41FC029B7B4F9D8A8F934BE5CDE5A09963F6EAA54481AA0B0C0E4CF058CDEC2134C819D0"  
01286 "D503FC2B282930D38DC46CA3C8E78980AF6B5EA74A307B4BBF0457AB22F01414CC927F73AE"  
01287 "A536DBF3C044DECEED8E4429DED9CFD013B04844A582915E15706D7F1EF64EFE9F866F68B3"  
01288 "520037EE9465953F1BCA841F5E9420D15C1D21EE71F89EAADDA54CF1931A38F1B9FADCB161"  
01289 "545F2AA6508BD2DAA48F1BD247FC8CE391B5BCB183887E805867FFC6EB9A97072A98EAD15C"  
01290 "202C16F2CFC5AEB50964A42E50FBAADB88698B8A2E2AD0BF307833A0FD8A2F5AEA9C562E6D"  
01291 "7EED05B5B33DC639E22C3994EFA7277161B84C84E3A7D25BE76580F3C1D30DD7911CA715B0"  
01292 "D63E4607B5CD81F2B9AAEE4FD92477FD3F410973616F418C0EAD9350CE238CD84935098A9D"  
01293 "76216CFBF755C610E29C0270B710230961D960BE716259C9330B80DEECB7BE79F842EF0DF9"  
01294 "A72BE1D200A8350B80F698DAF0011038B05B5802DB1AD3C511860F2A7C7D3D20F0FCB3F37"  
01295 "8D5610AA23105C985740E243BDA7CCD95825B91CB40B22858FFE20BE59626D327FADDAB132"  
01296 "91152C5939BEE61C8349080F6B4F916063EAB31EBCB492966D8BD3FF9F9F302E9D18561C9E"  
01297 "77383A80964EDFC521965D36A8CBA9308845FB1BA10C2D818AB4057B058C5376E1350A1101"  
01298 "A19247D0DB5C23AF619BD601650CA27FD0E540B692CF2497428E11EC3BB94DA045EA76982D"  
01299 "EF702A8FF046FBD959F4996922751E871F46756BD38A0B12134DDE60A4DFA3C9144784DFF4"  
01300 "8B5505FEC3FF737D2A4641D82C95D4334CD2110F3E8A5D1537B3FC7C040977C46B203DD15"  
01301 "12E42F4607B5CD81F2B9AAEE4FD92477FD3F410973616F418C0EAD9350CE238CD84935098A9D"  
01302 "D3668B25CC42229288FDE1669CA26F15B1CAE44553106100BBD8D3179F41C5625DA3D1031F"  
01303 "C66EDB8C68B707AAFE69CE6E59753DF34B2BC5E43AD0219268662ABA189E844EDFD78F5D7D"  
01304 "9F067F9396CCF41B5FC373D09368BAD3C8E0A7C579347097E82163C0B19581EB7FEB4CE7F"  
01305 "EE9E434E4FF63B27D18DE0F406DF53BFB9DD5B0AD57356A09955D4D1106F4AF4AB3989BE65"  
01306 "22D2590CD7FECDA82A638DCBE8D236D17277695B602C1A1BF3A0FD16C585B64E37E275097"  
01307 "5726341EA19E6EE9B5982EEE46177FFEC4EF0B5BE15FA4BAB3E0EB88FEB8EAB01277BAF5D1"  
01308 "48F033B1F8D24BCDD37171DA60D9323BA0A9D4F5342490E2443006A585A0439A221E1BB1E"  
01309 "F92DADB8639BE1A0CCA82861320DFB99119FE1C4D00BF629BA5C8D7E14E046D610BD534DAD"  
01310 "73ED77E66ECD3E7F0B93352D9FDEB840837CF898FD2A5CC7026597B1713CAFADDA930B24034"  
01311 "57AF0548FC02922CA0DDE58C956B6EFD50BB52C9142C35EE7E28C9CD5B60E1F4ED96F513"  
01312 "CEF6B1C25DF8AA793A077A39CB55D0F4E6B31CE4C94F02FE959C95F72688986B4A7FCC29E2"  
01313 "A333D56B82DBB02CFBB0C8765220F6A90A3EB3684BA4D8E22CD99C19B6034A6A60342FE12D"  
01314 "2A87814C7A2695E31E545C4FD4FA67C171350EA8C2CBA5A9CCA5C32E917DD96B3AD0E8020"  
01315 "AD09AC44FB87A30FAFAA189EBF2AABFCE084E457E42A7A1EE077C92CECEEEDC0E3D9C04244"  
01316 "67359FC82B3296710DCAC6BC23C5EEBEBF79CC94178A3771C11A337863A024A105A6B63DF3"  
01317 "A60AC8FA355E38EB54086CF638F921642DAC166CBC", // PRN 36  
01318 "C3916EF4D0FDC771D077AE01F9F5E078E91EC0976659A02F468794C8CCEFFBB5A466918F67C"  
01319 "68B8DE976346281A2D3B3CFD1B8DB762585AB799B51A74FBEA008CF72A2D8230646E876D3"  
01320 "F8A7FCCF3A93B29B9E70131B2D4B05769AB350E4A4FEA22FBB8D5A16CD0F4D9769F7206BB7"  
01321 "8194600DFFEECCFEFE5E6C371BFAD352D743963754B1A19BDA1E2C727F0B68ED60E82EE39DF"  
01322 "A90574C806BA74FCD316426F51DB2F44D0C87B505F72C937871C2BFAFC079814FE9477F6EC"  
01323 "BBBB13E1E8BDCAB5051254F40024ECD5B7565100F141E7982BECAEB0E80B763181DB10496"  
01324 "964ACBE480959295F5051456A62D67A2730A741CB51C6C9585CCFA9C6784D5185F15CA20FA"  
01325 "C79C18C7D590ESDAC7D7DB2C496EF790622669DF19C5D115A3D7A66717028CF6625C5A572"  
01326 "E51FF5085BFE804A915A3F9A401C10D640AB0818940B4F5BC995CD5A713B196031F5B8A0B2"  
01327 "6B4A1E2CE7059526DA01549FBD1B87592745F7F8330593073587AC464B967DD17FBA8B8854A"  
01328 "DF626503106D36FE959626FE583880D9C7600F5045BD238A785239E778BDC56EA9D21DC9"  
01329 "7C572C4DA832BFE929B0E1499745378EA13E760E8F228728C7D8DDA79FB8CB9BA135EF728B"  
01330 "FD9C804FB1AECEFF929D0FF0F375AB23CEAD24F05B47D8A7564C89AF207B1B3E911323AFAD2"  
01331 "90E7F655E9502DC960528E6E5C58E0F92769BC1942E3156C8AC0487945EBC2C4F11262F"  
01332 "0BFF45594EED244D6B8B61E03B5A7C90EE3B0E710097C051649ABD1615EB88803DE26B83E8"  
01333 "66468F14FB22B5653CDFAC57646400215B192B703B659CB892415F001C2457DE8EC0B08F39"  
01334 "69582D6C7CF26DB7E5B58387A82C76F7C016B27D2ABF355704C9892EE667E1AE76EDA4E58"  
01335 "DBF4C3F388423C15E63967D1E52306C84E94B061B28BBB9CB2EB6E04440215147B3DE26C24"  
01336 "852AE0B710122860784929C932E313237471768789FCB9AFBAC5D7F72665D6D2C51FE5E52"  
01337 "0C4ACCCF2C9BB15BCDCE756D357127FDD9FD8BB5A008EFFF7843B1A6310761E0F83FB61FA8"  
01338 "E30695135FFCE7050D7ED09879D3CAF36F3B990FA3211429AC2084D8AC40CA4B49FE83BC53"  
01339 "7FFBFC06386691254DD47D9699EBDA13CA00AD9F7526C54553511C14CBBB2D150E9DD60351"  
01340 "67232D465FC39053BD06C4FE0A0FAD212B950DD3DC1B717F43838AE12D322A9EA1D378ED0E"  
01341 "12EA74641C0E0ABC9FF3AEA2B2E61CC523AC801E61E94200152F0B1C4B35AECF0AA175C8D65"  
01342 "C45B2002706BF2D9A9A51ACEC1802B9059CE45FFE67B69F79AA15A17686207DE4D67054FED"  
01343 "ED7374A46ED3679156ADAC2F71242173AAEDB7E004B45298BC6EC6A0521EF47EEB804B86B0"  
01344 "78C8BD6A2D0F1239BDDA751C96526F1CD51F28D23C4E4C653579CD37A89CBE86355B0872188"  
01345 "56F359A702436A7DA65747238BA4B94161CFEEDA0922BF1326600B7E0E67FD2475BF43412F"  
01346 "E5277465AFF9E367A902D1904026790F506304D925CFC0B90278D587F91D3782DE0001AB9D"  
01347 "4295D67F7BD0039BA2917DC567D280E1719112A7B3C41254EF82B42653E5F1FC5631E8DF9F"  
01348 "E55F862E42A34F4D8BA19B76E6624B71F911C051DF63C04BDA0B0F1836FE3E3F7970C042"  
01349 "EED4EF0C4AAAE9A490B40FCD55765C365FE1535D99BA5FD887E70529891E77809D69E35F5E"  
01350 "9A52FCC12DD6032B11F7E25DF8D00FEEDC19B985C57DA531EC7AB2913870CBC73A6F38BAF"  
01351 "55B5B6F06A6AB853C1B84D2908EEDFA98985625C54C5CEAA882BD8366346050CF004C64642"  
01352 "647C89443AD1013BAFE6F61F7AACB88A3824145138", // PRN 37  
01353 "BD5C95588AB0103EB2D3C065D92577A696CB28D1A17BF96CB61DCCABBE92BFF6BDB9C278B2"  
01354 "CB77F6ABB0A1FBF516395DBC894E98702E5A18EF28FC248D10807CB9A9BDB540714BE47B3C"  
01355 "A4EDF433CE84717C7C0EA4B12445F25D3EE604AB598BA606C10CAAADF538756710FEE64D7C"  
01356 "79D3ACD204F95798946A01870F4AB18337EDE971C3F1958CB6DD51531CCCF357232814D6A7"  
01357 "F35163A5E04C28DAE0BF8AD281620176AC90D583CA8B63D8A846F5FCD928662BFD1E083E2"  
01358 "72D5D0384AA4F1E95ADA8A8445D5492F00BE98A69370B887044F757318761244BE37A0E25E"

01359 "8A0B7F0B1EEE6B773A9399BB42633FD4C155F4E27C8EAED2140C9AD89D238DFBDC3145AD7B"  
01360 "13377C1B7491A968C5B7BEB6BA7C996EE394124FBBE7DA26D1FDA241FD9A9F70CFDAE8FCF3"  
01361 "85E5BE9104CA2FAC46664DE8D4A9083B0F944CEDA34A4FE578ABCA8D5DF941D1C253877176"  
01362 "27046DDA105E76F7EDB619028726CC5A678ED28F93B7781BC11420B5307B22122A3114506A"  
01363 "48451B2BEA9AF10C02B9E02F60EE608AEB57F91D345B2DD3D8F1D11788E47E38DCEC02D867"  
01364 "8132984EF775F180BF7CD102AAF80641FF627BB6D3E29E9CFAC77D650048C182B1E9F5B2F1"  
01365 "E95D4078F5DE78E9035BB8A5C755DB2497DC03EF268E1D9ACB631D7F03E449A7F865AF41B8"  
01366 "D75D6F40120793F11B879F6035BD9DCC1DBB38BA47E1C949891DEA435494966DCC1303BE40"  
01367 "0D9DDA0745E01600E35BDDBE018730B2AD28C58C375341BEFF914C55CEF195EBEBFBEBE181"  
01368 "085B5F03D404EBAC001CFFE43596541DB6BAD58879D9B4E29156B8F7A2B17985B27583E233"  
01369 "5EB8BCDF0C606DC1856BD8421C812824ACEE86A1DCBEF3F44E60E4DB38099E59D5C5733311"  
01370 "27BE650E6877E609C819A80F15697D04F1A3392CCD5A021AA83A06A2AC2D4C18C5B168806"  
01371 "EDB96E4283B22DBE71FC1FD43075E73E5787D6D02777BCC5E25FA414110761ABAFEADD55B"  
01372 "F15C6829898DC70FB3E37E06D7C8D1AE6FDC71408869DC97B256D565B035EB29D13CC728AB"  
01373 "2310B137A361B02B772231E199A4582E08999F14CCB8A7DCBF71D88D848FF827A17A756F14"  
01374 "078A28545D2403FD5A7EDA997AE2EFD48E891A116C74D9F70631F39F70604B88AA52232E40"  
01375 "5CFAF3DFAB862336EF2436CCE616705CD9A871D3E25B16F3B5D54CC876D38E6F1499E52081"  
01376 "A5ADDBCO0B744CA190C5892A43B39A78D22F57C9867BD73A468FC0CDF0FD762282DE881A"  
01377 "1AF674310DD1EF1C7EA8B8C23AFEF00AB5FC18C71CE271739573835CC0ED4D1E60734E9E92"  
01378 "600BA19139D69D88BF7028DCD26BF816F22F5490C1B4AEE92AFC0B77DA3DA9FA0AD459D5A"  
01379 "602A09847739816B574F189E5E9C8FA87A20B8A9D0DC07FA20353E72062F295A636AC4EC3"  
01380 "64DD718B618AD64FAC6F37ABDBE9BEF09DF9C8E8F0FB40EA866F3566F978F907EF2FF1D30E"  
01381 "F313E0B317E873CF6B18A0CB7303538167C155C2C0473B85194AACAF5F6FB6F7E4691CA69B"  
01382 "AF6BC81C6A0DFB8CEC7480DB342D452FD7284431ECDC1B206FEA1D831062F3842EF08BD1D"  
01383 "EEC2F51064194C7C0D764F2CA7A6968E5065341C4E6F11759E96ACCD5AD736FF6F0F471E93"  
01384 "3A288B5AFAE36C1EF1F117D91BD72B1A33B8F68CAE179EDA81526BA4187AF6C10776308D17"  
01385 "9089DDCC292DF871DA31E08E738041A3B2FBF6FE5B52774A7D0B6713B8ABF0A4A5D9A3E2B8"  
01386 "B8A219276D28A5EA967AF6C95E1C56D7B7E3E877AAEEDF70B6985E35E4E0811C02EE4B95E6"  
01387 "4DA809A0A378A41399D504C4270173682FD353C588", // PRN 38  
01388 "1327D094F98FC13CFD8FC87EAC26CD9BF52F8844EF532BF7194E4F4CD8DF9158FBC6A11D9"  
01389 "A6AF40901A6FB4FB82B205364A878A433040F9F5687769C7F1CD3CB8F5BF9D05526FC5F7A9"  
01390 "52D702FFA9CFC6D30508F28E0C802020944BEB591297786344DF8B057092A4082DAC163E95"  
01391 "AEB74ECC623913CC3226C095B9D5FA48185BF6CB515502E9083E1FD67BBC749777CC98BF20"  
01392 "5F35621523D8DA72A674BBE1AEA89E0327D8EF6A26DFEE142C5CE70716D2DB232535ABAB9C"  
01393 "619EDC560504647620C1DB5D91ACAC289AC2EB8805EDE5496F51BD13D90DE4366842109A0B"  
01394 "E405F1D07D03CT719B321806A9E7ECB5FD70C9E80C17827552D443DE9A92772239FE52854"  
01395 "43096B7C7276B767E48F7A6FC57460EAFCF6E12155D6427807E152D435E00E305EC7997AD1"  
01396 "6217700C8B859F89F3391007DFE38AD6B34A8769644976F2C2BCEF4B1272540BB0ECB22C93"  
01397 "CC5FAD1EB5A1B1178504F7A35E0BB42153CDF5E4184E63D5C9AA140A66863B26E6773FB38F"  
01398 "BC3EE829344E55F8C7FBDCE663665617189886216DA9FC6E64781A195EAF8E2705DE5A9095"  
01399 "B93E90D1A31EEDF0EBED20B2906E6ADA6B42522093AE922FA4A530BAD07E8EF0A5ADB4D49F"  
01400 "AC1DC4A5A6DC083CB38DEC75EB4A9257BC3A9613FC6993D499F5BE62AA03A1395E8D8F566E"  
01401 "F09D95A58A4C85CF6F40655B8C3453B2C351DC290E5A356A3CE3D7A56EEA3CC586ECAB353"  
01402 "20414036667FF6BAD06E471A1363DF7BD65B360B68D6CF7FCC47875488A386C9A77839008B"  
01403 "596B3ACE279C941A48A6E751EA3AE29DDD423C454D0032EBF6E17EC497AF832B9ED80D4B4"  
01404 "FD4CA48CF2612EE39EE8804D5E19102244609063D9EF9B01106109F837A857C4879B01CEE4"  
01405 "D2CE94AEE2C827BDB6732DED9A1DFD4E691EBE956979368B0E8AB626887FD05BC95AA671D"  
01406 "1711A688E3AB37827A7E4CD9D0B9B4D596E67099966DAB4E25040D9059720F651C063DE8A8"  
01407 "90224052EACE6A7570D36DABFA0B6BAC8A38BA99AFF1B6E1D8689FE22D8194F6D96B4223EB"  
01408 "41EC4FC071C47B4CE1845C40DF584A18668BDF1A00A36CA409E47357E910FEBC3FBA4DBB7D"  
01409 "52A7976E9D080B9B0FB530456BB0B328FB05FD223C65D24BF41BC70519BF6C599E059E3"  
01410 "51E3E3E12739E198803FAAA48AB0401826BDC7168236DB0301587924BD2AD5A65E148858DCA"  
01411 "1914A7A6E5C055AECEC8F2B7C114EE2668D990309BE76573AD8FE078639497C7D21CEEBB3"  
01412 "585AEC412B8066395EC69E2242FC348E2A6FB22D3A9ED5CA97D71076C190DDDD08078E030"  
01413 "0A202B342030D015AFEEF2B43D65C29CBE8706CBC341FA08D69D2DCF32AC9F5E6387060B1D"  
01414 "CC37F8F30C5552DFCEB1BD3C79E7B47B44A80EC57A9859BC7DB605DDB86886DA64A2C8BF5D"  
01415 "AE05B4A01DBE4879053C8E1A38FFAB041F0B5405FD04AC8AABD57FA3A46413C21C320F346C"  
01416 "7CFDC19B631B29812C4E215288D6F9070876167A7AA68FC4862577DC02E9FD50FB3A3FB61A"  
01417 "5530E1697BE4A3ED6C4E32242EDCE457923B0023F6712C72975AF49A1E73565A4F17552F"  
01418 "3C9B4FEE16072F65568D78C51E4BACD6A849C20425F32185F79E1E301FFAD704E2D885F2BD"  
01419 "CF152AB4B067B4B382AB97A307D92EE1FBFC1C10AEF5090CB8B5BCB5227FC3207584D0F7223"  
01420 "916E1EF2CACD7C028827B2CA5146D33965F9C4EB431139611B6FC2427E5FBA87D968BC853B"  
01421 "DEED3C33A3788B4DFB59576983A3951BD5D85779C76419C9AC0A54A3957CF3C0974F71E93B"  
01422 "622AB66CFF037C91020904546672DD0FEB05BCD490", // PRN 39  
01423 "EA921FC3D3DD155F48B3356D78FBC9990FEB705BB3E02295DE80D7B39FCEA061341D7C410"  
01424 "803C284424A13155F0B6F15D09AF71D12A66A24C5C152EF830B95AD3C0F7B8CC8DFD33104E"  
01425 "5C6809E22BA86B45078456A848192784A85F4939D90C9F53BBB31A7141F91B5140ACC1CF46"  
01426 "34DEFDC070F67C4041F15796D7C2CA4E7554B12361F2FEEB8E7A28D0CBA2F1A707C149A30D"  
01427 "DB3FC0C4C964D9A8E06BF872940C22E6DC33058BC59EFB6ABFE6B2E278084ACB53CF8BC2C"  
01428 "96AA5830973B2063045510E19CD315144F51ED99419257B5B29AA177F1625AECF95925947"  
01429 "DD67C5FF6ABD704AD3A2DC456A78FEB2431CE62D037BEA00FD8C25DCEF37CF5579A2EE1C3"  
01430 "F1465BA89EA2CBF2EBB412907B64CFE0EA7624D8A839E733912E7F98EEA283C87AAB455E51"  
01431 "7F8F479A7AE8B4DE47E22E0AE8DE0AA71B982D321A3CED0AD27045A14D202B191953CA21C7"  
01432 "51230321E184AC73C3049403A2D423A7BC8BEFD783D5D3E5BCD7CDB75F8EA10205CA5B58E"  
01433 "CBBC8564B30EF045C0B3215E768DF4D6420EA0BD956421065667E24B52ADF162ACEB33D508"  
01434 "2C211B9379B6438D7F00A8CC2D247296B570D779A34EDE8D16EF70669DBE7EF52CA48BF3CA"  
01435 "6783539DF84415D804D404F46D91743B417E4244E923A01802CF9F5F1ED27F252B991606CA"  
01436 "46987F0C49B447821E9147355FB20735EFC48412ECBB558F04A70E871DAAED3A2FC8B7A23F"  
01437 "F915C88AEAC4F0CDBF12C845136C4F623566A3BAE280371F602AD5F3FBC040E58C913CE5B2"  
01438 "8370C516E7C7E08FC21CDE8461DCD960138723990F50DB0D427DF0634490F99ECE15FBA33"  
01439 "8C3967E4715FF15578538CDA1027B664656364D58B05B8D3A06A4CEAC3F7B49AB2D6540F9A"  
01440 "54B4E3CFDE20F44BEF251D5565909FEBFAF44F5A23759BBFC4BA6021EE7D7B3D0B5C0393535"  
01441 "AB331CAAB665D91DE334B854B23CBAE5FCBB9AE2D44E527E6B49BB082A0C3EE960D251AC7"  
01442 "9CA0CE4EC8DA79349CAC602276B51DD37450B95629CC7120D08E3BCE8909D17BFC5EEC278F"  
01443 "6BDC9C95C2708B2064209B4D99A12826A98D13844A8AF00454AAE910AA5145C3B82289CC03"  
01444 "4E7C82E3E2ED2F8E23E8CDD5FC40F97FCB35EF5AADBB7EAA95BA760471711911995669F43D"  
01445 "73D64D07D56C69A4B2B2CD41DC03ED9EC6A92241D3FD084225D38FC625192F5AF85D299FAB"

```
01446 "778CF6E0E0D3FDF3A7F79942085CA592260C3458AADA2F776ABE79F372AB0BCAB56A8FD04D"
01447 "8EE81DC0C16BB569099114B91B7DF5E56C374111DD032663AF48EED3765CF698E675B930B7"
01448 "6475DB9AD8256CB7E4A52C5BA9FE2A3C5808AA2A43A013A9D2A1976C4645952B43BE206179"
01449 "863A740FBA8239AD363440ED9CCBAE423F22FCDC16C48C65EB0F484F38EFBF8F1E1D7863E9"
01450 "6F6B82F04F99CA434831DFF74CF080DCCA900214258CD747E96C27F1066778A5249576B7BB"
01451 "12FF31BF8A465C2B04208233A1FE0875031264DA6FC8AC54F686A7D40DFC65E2EC9A78A8BB"
01452 "751C072196FC9C238831BA2A9F798D7D3D0DE08A7FEE095236CDC8B9D77EEAA0529C83148"
01453 "5CACA590C2404D4C3FBB36A590D9D6576DAD650D4BF6A8609C85D9815314603C365C07AEED"
01454 "1FE424C0AD100780C8C85D8397A18C8591C26D2C2E48C64582B9FDC534220F12189ED2AB30"
01455 "A972A647BB60DEA0724D4BC12F64484B647DA7097F663BFD0F63A39DC729186873B333A92B"
01456 "0787F4E8A335DBFCAD7E399272194F9FFB6ADA78E296F634770F69FB851E583EEA8C9CEB54"
01457 "97A9A643044C777581F09A1B3FE6DC46290F7DA584", // PRN 40
01458 "D4586907EC637459D2E42A147B17CCBFC1A79225D518E2515F2C89E4AA560FC958DDF8F81E"
01459 "726784B064F97EA6C6C853372225D4C8CC2B3F5CC2CF46173E83D09F68A5E3CE78A9D2EC2F"
01460 "7AD0E1731D59A86F44121C0455E850FFC48CFDA027A93D74C088C94FC9477FB04F28792C5"
01461 "0F714387DF767B884D00C95C0F07E13241CA878A324B4F08986FDC52393E1CAAF82D298EB7"
01462 "91FA9DD823A12AB51E8C9AFD7A2D0A922F694BCC931F62916A4C348DC4BF2775488B92343"
01463 "4DD905B6DC97AD331355D0F442E4444ADAD1B7B1D630A8AF4330CBD9FA96693F4250C2D2E"
01464 "264B6EE736E368EACC2CEE260F1D7EEF4F8BE70771F4C3E31383268937E722E40A930406B4"
01465 "BE8AFED2D8138975F3813C6546C5DE1C72510203E0540E55D4EB7A14979505893DA5545458"
01466 "81AD9D4CBE2EEBE5781D1A0B0F1D1444FB1C5E600C6EA92814E2DF764F123E015A1DE147A3"
01467 "F472C5D5ABC7D0C2DC631765EDF13769ECA5489AA0747338031401D56431AF6F8F0CE2411C"
01468 "C12AF4F8C1D265E4FACFFC71F432388A39D622264964B14725C9858F8442E0781561E0AC30"
01469 "3A79BFA0FD5DC07B71561265B9DC3ACA8C68FD7765B504B78963FAB0948EC9B90407897A1E"
01470 "4966DE16CC1E332DBC44422C040C617D15671071359E93959ABBAFF81AD147C5359E3734B"
01471 "EEB3D4C904188762AB326B4CC635525070479EDD06B67772A652F467F5E95A10D79FCBA8E1"
01472 "A830D659D0EBD39DF618F9363AC3F96E5E4BDE7B33A4393CABAC960CEF4E5E34AEF678E07"
01473 "3A0C87E80F357682DEA11BFD6D6AF70AFF4C7F3FC6109891E8C0AFBB24136F2EF7C976A9F2"
01474 "911FE14C87E5F6C09439417118CBF25983C23879A254B96613F61F56021C039BCAC420A120"
01475 "81A0D49E6EB460335620AF6C359A28FA32CC1EB2277211739DB1B002612DEB2B787875192A"
01476 "4C41236EB20B133C2781F7B97712B6D9C399BF21EECDE6B09566086F309373F56335F4039A"
01477 "100B7813B2E048B90258F880D13F78B6BA20B4B699A2DA8E983D1E2CBA039037B1B399229"
01478 "37ADF1CFFEFDA8B70F1EB5028AC1A40DEE89E29F01031F1ADE0A0FE46D8D9CDE478311AD2"
01479 "1F01B79F67523EC39AE80FE40D5EC2FE853FE96A4267C5FBE26F1BC3E77975A4FD3BE9D295"
01480 "2B6D5EB0A894B4C9F0A594075C9114E93844A0DF7FF53917DAED6E30A42ED59B704AD4382E"
01481 "85CD0A2FC977F37F303169C22B0D79458085898F47A8AB0F1007932EADCC62F0D18528C48CB"
01482 "393E94012FACBCB00F692A85FC3E1A5F07924847BABB7043A7BEFAD5F7920DC75DBB1F8F97"
01483 "F9B2B4932AA3D1817C26EB85812B486BD7E3FB0A9647C57713C5412D564B0E3002D7DE3DDA"
01484 "3F7296879CAFE0470FB6CB81B67EA98F43DBC3024291212B8849953D5ACB20B25EC9589E7"
01485 "5091C6F9094E58302E87FB7DB179F56322DC25273EE63EFCBD6BFA3DD57DE04FA3A6C3DC46"
01486 "8B3AA47280DFBDAE95C1822311D42D33BDC0FA216773E8E3016F3270478819D9E2B532C697"
01487 "757547CC0B4203EDC24F2600D37FDE321D69267B45C5D4F7A52BD9DF858A11BF649A41A2D6"
01488 "191520D3A29B66812A02E4E67A13B553707792332F81787D53B2C56176F63FEB8C1F842460"
01489 "864761F9D90AEC7A990139D057E0A82B7C17F30A37315AB2DF7D76CABE6BD748E288AB6F57"
01490 "6744581488C0757F6FDAEC4E3D34F6AD5AF0616D7436E92BD4D2CFEA92B586755AA582BCAC"
01491 "95A392D496F7EF7DBBE4AC50E8994D27FF3A503EAD5EF1F9B7A84634B8937601880309505"
01492 "1C0D5D165F9393E3B84C511099A58D4D7CADA3DF20", // PRN 41
01493 "3EB98A46F91DE7E3870144CEAF95CB692B022D2E19C84C36BFA9FF42EDC5C413940812629"
01494 "87361544A4CB4ADED1CCF52CE3BD731F4EE873CFA0FB6900B0348397F86111313558F297C9"
01495 "B82CB185DBDEF29AC2723B5E0E6F24FA0F456A126899E99814F45C40F2278845145D8D8445"
01496 "CA88BE687A3DED47604C121E82FFC55E9DDE0797AA7CD9F9F5448BE6E200119DC5AD582665"
01497 "3BA1E3ADB8593A79B466C5AF053FA32D57641F51EFB15DC52F5150CBFC28528B8B1F3FCC05"
01498 "57EA9E2D6CA7D84ECAE8E9FCD8C0A7FDC9451A0C9F0E470DB496B1BAD754CE861F87CD77B8"
01499 "00D7962AB8B55F5347EE2247058D7DA26CE7A926E7F8139EB2D5B1DDA6F5C59408B48B886"
01500 "3BD3CE3294B13B3FB35D71A5B6C057154D7657C87EC3360BACF69EA3CE7D840FE64338A91C"
01501 "0B2E6D568DE270DF5920AEDAD1016150624CFAD6098400EBEC2092079E3053E2DC49666B36"
01502 "AC49273B5D1E77ABEF9E5C21A30740CF91B42443F4FCCFB4B812768A18BEF06F809CFBA221"
01503 "549B7EAE39B5989A8688F12321E13633C331BFC529B69877EBD10053E7B35F3F35A1D62F5"
01504 "25A5FDF014BB74E7FFFB11564E776FC653067A0610766C34B7BC4C562F825EF73AA1CF5B1D"
01505 "BDD08CFCFECC832689E6B07D231EF35A84970F81692E8805B1784CA6D79DACA06219DC6F2E"
01506 "4A00A12B0CCFF3306071EDB6400DE599D9B5B661240C652768BC6FEBF1DD7725291FC2BC"
01507 "424B2F46643EF37469F15441AF27B59DFD16D2EB31B45FB1DEA9888CB1F96C6FF49918706F"
01508 "39C301D5C8F47E647A8DBF7528F7D23F51481B15F69604C72409E518A9FFBF680EAB4866F4"
01509 "5D81AB651B45AB8E0E7E0230A3B475C147B546657941341426D20A7AE124E622E18654589A"
01510 "DEA5AB06D643D1366198AE4CB77D7435BB61C090120104D56FC19CB3294AE25C7F9469B45F"
01511 "1F68D3FE0A7923683A0C00D022DAFCC6F8BDE4F739CA0B586CE55A42B851A812EDE84431B"
01512 "BD50DB77EBC018DF85AC0C129389B7C02A9CF59DDB6BAC42028081B59C2AA25AFADA52D471"
01513 "C49EFF6535C2AD8290CAAF4771844FF82F5965F154F3B1E2CBB622B50D0636597F9091FC5D"
01514 "B3E0377E8EF33AFF204B7B0D072BE2C7A35580204FEB394D3504693B6284A441D121D0A91F"
01515 "D97432EC7D4D2270BD37148065C46E568898141E11C36945FD6F547DA97F38EF7F610CEDB5"
01516 "D3DCD2A7BCB2754A5B443E21DBB0ACFF71DB4F9A787A94185D0C98C2B2654E2430A3FE43AF"
01517 "4B9E816F0C3CCC13DF612590648C0DFF408CE967965D38AEBF813431881A051AC9F896E508"
01518 "2238E8B5D2965CFA59CECE1CD7170D08879C6AA1ADB4D5217E587A3AF1FA4461CCE3D4EB"
01519 "7A3B82432C90052335366738A60CB2B1B5E90D2B39FAA80E3052DDF2BEC9358868E0F97E4"
01520 "FCB707E15FC051D21810E8A38C6CB5B0FF05361F225E8B8D10C958C7BF84EB31813DB1DDC2"
01521 "E8D898058243A9A2F41E3B55B33B87E832ECEAF80DBB33E34C3F830660E6228586F6E825"
01522 "F874623EE48FDBE8FF1736627881A3F1F06B498EC6D5FB0CEA203D5C7C145EE6F09B0860D"
01523 "D86436B036C7CFD873A245D976C63BC5432B487AA0DB087A62BFAACC6575E265DBAD89D41C"
01524 "FF89E626ADDAA66835CE497C17E1440371146EFC8846C7FFA13C4D446078324598E59A68D8"
01525 "DB15F2D027DFA3DD2E7123E380B63F01D8D07C5B42B43AD76D95BD7B57A5CA2EE971F1B9BA"
01526 "4DF6F8DB0C0E1D02074C070A745FC09D1BFE1900D1820CC42F66FB97C39BBAFC3C66FDF94C"
01527 "09F64FFD20430BADDE7A84C3EF0D82F6E50B96EEF4", // PRN 42
01528 "9FDE160085FE8806B43B4E731B2026CE6395293C827458A52B4F92D94367A864C1948C7FB"
01529 "165D041CDDE90CBD1290D664F2D4DA43674073E420BE61A4D4B0B20546A64FE8B9B1846226"
01530 "98E67660C364BD2F97CFC5176DF4E84AB5215F02DDC3C47FBEAA523DC8CB14193885850ED1"
01531 "03109C3EA9D49422968FE54CD6ED29E94187D73B446C21F369D506D6178C91ABDDDFC6736C"
01532 "94C7C5C8410AD4C8DD6198928FAE1ABDD426402810E6FCDE33D835C40D41330642EB2D7FEA"
```



01533 "1480A517E39B78BD85668C2E0B6B1E30AA9C1614B07089577A20BEB06FD51F5D73FA78FA64"  
01534 "34E7A1979AA37906E421EDBBF9A2097202F5A63CFF2AD0061EF94BB830689EB5A057AB1962"  
01535 "1CCB551D75AE0BBCC5F031C7870F35CCF8EE8457171911E198F58AE679283DDDC57AACEDAA"  
01536 "A0C5D0B7E4B22D68C2F9DE42316643A720A8421FF8E6BBA5BBC639BB235B2BFA48F69E973B"  
01537 "BA14F30078B178CD67FAB150DC24390E8672B3A92B5FE92D30D70AF6A6318E0FDB466F3CB9"  
01538 "BCDC5E8E48BD8A37D18768990C4424DAD0A7A269BE56A7C58F6191B797AB4E019C77EECF9"  
01539 "B79BCBF74616E7957BFE239E47F6F9AD9F4FC1FC16BF82763F34609BE72A4E6C90C18141AB"  
01540 "A71A16F96C25AD4385340FE67FEF3A90D6436C3455D39BD16C7C4E5C1365B3FCD110BEA8E0"  
01541 "D2FFE63A3DF63CEBBD2D7FDF7680A67C83CF51416B4AD115E7054A5CDAA6443476E207D5FA"  
01542 "24F87BC438A30A5B078F24AA5F966DBB87CFB2C3FBF63BF84ED3CB4DFBDF60D64AEA1786A6"  
01543 "2E80176E3D62AE69059086D3B3D71313A7B0DC288FE63226570983C85D857862716B6002B7"  
01544 "2DEAAAB8D4842AB531FBCDA7009996B6A8788E2048D7807BE6D1C55C45F8AF816F0D1BD633F"  
01545 "80F346AFC8AAD1A350E3A13FFF3B1D16F600A1985AA43C5376160150561FFB4333974AF3DE"  
01546 "GF22B24213F35E101638C93B216F5EB484FEDEC67FD38F4159DC521A47AADF8E42B6F1EC43"  
01547 "DB52AED253B75883393577366084F90D6D55A56E6DE457A3087020DD90753D84008EEB883"  
01548 "10534B4DEA40EB4F5282AE9EF2FD20EEFDEC1EBCE713629113B1383827A381BE262F10F097"  
01549 "1D45C2597FD9A108A09E32DACDC81783F283C44049D13680CC7C07CF03279DF86A22D7FEC2"  
01550 "42A7A26E63019A6F3B703F84310B62282C3C46A9E19E81B474341D4CB16F763A2B78FCC4A"  
01551 "56FC128048E8E61CCF1DE77023048150FB3C96D4A0FD7FCCFC8855219A30DD2A3469D5670B"  
01552 "465A5D600C645C802D1479298DFF8818C152A21208D520C301E7FBE5C55F744E6AC45F56AA"  
01553 "31E3A596B31AC4836F53A78C427D8B3284A34B77B38B6AED44362AB36F72446AEA179A22F"  
01554 "45E93607C3F77F44E420AADEE877A90AB39E5257B6A5A801A04B3ECF2B78BFDE9089C5131"  
01555 "D6C031DD3D02EF426437502339139B5484C6EB481C3D3337376842BEE792F148FA3DA940A"  
01556 "Q208AB3AF8F6F96F02F64C42EF0DFBDBF178493BD724700A65BE1165FAD8706FA529DBE891"  
01557 "A95B27C0D15C8C3DA300DE05DFF7B03F7D542C1BA5DE76CFFE23CC9A265F69B8D0167F25F0"  
01558 "6C5C5918554CB342EC80CE1B5D671E99ADC2B1EE0F93DE52325CAC69BBD25647E43E8D30C7"  
01559 "96845B0A656506D9BAE8B9B43F1086223020703B789E5F34465BB2A2E044CA2644998AE6B2"  
01560 "99DFBA547F0DF087CE37614088DFFA6D1444D6639C3D7EB8FBB6518B3B6C2E8C70BBF46D53"  
01561 "E982CC0BBF6648E403BE6C7C50717BE49BF4BCCEE20C57569B41B40C666BED36B7E986E91F"  
01562 "7765DF50E36C75544A766E0B83CD899C2095DFA70", // PRN 43  
01563 "2A04CABEA481FC7FFF3DC02DF99CA09574A05F592C273C050A81C787EC8B7DAB7C42C553EF"  
01564 "9FBD8A8F9EBD77E0115A205194015E35455E80B48EE8750C546A6BCB6F782F1D6231386B1C"  
01565 "CFBB1CADAD8CC6565DDF39995B4E8404FEEA9BB1EF537C67C7BCC7EB820DAD3B1708EBC9CEA"  
01566 "451F6F87B6369615F39F0716719BCB7E4799C8AA568DBE32B3B279C1CE3B48B205D23CEFF0"  
01567 "1869039D9A2469D5C5A757E8C3FF4978E50A4FD149E85CF51D9C2942F8571252B6D76EDD00"  
01568 "CE538B2F7FD2944709BE2AA58B801780EC89B596C8FD2C938DAFFBE05B26919BDFD56D738"  
01569 "5F8F0929001D3BD96C87313B98BAF1FD1F350139FC32A7E84351F052868B55B1960D192FA2"  
01570 "0F9F7940FC75B54031B5BD73DC501C86F253BA07B0DE481D97F251F8E2789D47AACA0AA654"  
01571 "730729971AD5BEBEA7F98C32B3D61FE2B84D5535AEB0D38DF4050CD886D7FCCDA95480B3A"  
01572 "7657A8EB3C518FAEE316322CC33A4AAF71AF867BAFF4DAF11BC70E8DAC32CEE8A6E9B2DBEA"  
01573 "A34CAB6DFF3973748F1C24D8EFAB30BE1C21BDD33375008FB39979A9125A3CE7544ADE5806"  
01574 "99CA17533752F483970B3F5F0240ECF71CA97F808950A85CE993513889937F7EDF5F54B95D"  
01575 "4A1D09044E210E284C0DBCBD6DAA44546C9C9812464A166418A1E45FFD83DA719D65FD313A"  
01576 "67012AD8F710B9A74DEFB72FF688FE69916C0E2824375548455EB1A81074610E48FF80C25F"  
01577 "13EE7B098D5C4844868019511834BB8AE6822F1988EA6EDA00102B679756CA8B560A75F62"  
01578 "E74F35B2FFC31E0490760C0F5AF6E6769632D2C5F479001DBF076453D3F56A935E809F2376"  
01579 "E8ADE65CEADE115E6A41F294BBB08B4A15DF94EE426B0B02A367FB261A0D3D321CFCC06803"  
01580 "62COA1B478B34873050C2FF42CFC943EA973BEDEB99639FB1F7D6C88E024518D7C99DE7886"  
01581 "E5E17B5020ED44F10C7842766B8E13F485A2ECEE3AF0AB60D549A420B166B8C5F51988EA15"  
01582 "E47A6B002E1DF56CCB7E6BE805012A2CC5D00FBF02D905537B99F7B972945B4444DA4DF442"  
01583 "02E5D406216F5D05E8100A099743C3A0788B6E3B2747861D7A36945C2613422AD4875D8BA6"  
01584 "2E6A08C60A5E93249726657014C0D91094C3EFD4109D2CBE40214840F49D258FFBDADE593"  
01585 "9B73DB80D89965D5DD219EA58D7BA8EAE202BF9CECCF9DAB3F1685DA45CD948E0ED96E6DCB"  
01586 "EB179BC1577664E4B3E1A8A6D645A08E1E481C7E2902D16C79A7051F8D57DBCDA2A66F45FCF"  
01587 "9F255599887B3E7343FA48B1D32994E8549AD88348443DB287C51C170BC2FB01DAE1EA09F7"  
01588 "DEE8EF145EA61A48EB94A4B555C79C1742ECC06269F9D5D819D3C1E8AFB6E1954941111096"  
01589 "E4301B7FB83C3C638FAB53B6208FC4A4146F183C4FA61CABF9E1479701B555294DFC59130D"  
01590 "0FC3B9EBA0F1203D7A1A87D38E2DB07E469A25449E1AE16B3B84BC27E6E67BFF93D3BE05FB"  
01591 "B2323E566362D94694B6004E5C336D9600C7B2DCCF660341C035A456DB70D4048A872D3EC5"  
01592 "79773889A2776AA85FDA398B91A5269C60FDCBF33B789EEFE9F5EB5ADCBAB1F3C6280BE20A"  
01593 "54D5D0D62B9259A67FC97A71424DCD4E94C0CDBDAE07A39B3F4F81B521E375D80484BE055D"  
01594 "3296BFE5AC1E752A3DB73C3AFE2350E9316622285AD2DCE65851E392858939BE3B2FC25CAA"  
01595 "2C0016E2FF66086E0303B8C681B84DAB2DAF1DE22B36E6BBCD31F27F233FD08D3399EFE0B"  
01596 "E6C054F8BD1EEA98823EB7AF305916536AB0EA077F5E092466B4417C97CB6789FE34AC1BA8"  
01597 "1F3EABBD0A9F4C8D3F32D4D3F126B392CA4B18D3B4", // PRN 44  
01598 "3CA56F1313E324915404FC17AB31C2BE7B841795544CB86C08ACF2AC69BE7DAE0C4A644763"  
01599 "0F3BB8D84656CAD18C74FEEC701806C9E649C942245FCAC57819125D7834A9BF749A7DD7E9"  
01600 "847AD34D85FE0046F288C72DD8440534D94324BEFB896315E232EDB57E9A4ED4CA39DB9"  
01601 "D72EC99907A690D6FDDCCDB744770FB2F8B3B57C7BAD559502E04EF3DE53035D3A38C8CC67"  
01602 "8945A265D708586CFB2501952B920E1FB4EDDBCFF2019904F47E95E7B542C33D1B338B71659"  
01603 "18926A786FE8AF2A86089B1E0D06117742ACB1E8B2F6CD607C90A0355FA5128C6FD7FCFC06"  
01604 "E50D1D9440E56645771B49071C59E298E316E72981B96A53FB979129DEF897E6B7E75C2DE6"  
01605 "7A5961CB36BE8F360C8B7DE5092D6B3ADF79B1B143A5660E8BFBEB5A60A240A5E94E65DCF08"  
01606 "8FD25E00B7BABDAC300693B3111DA565DCFB8B4D97FBC4E23DBE163E0A42F2E3181612CD80B"  
01607 "6EF8277CEDBE43A59844A1FFD632B9CB7F8F5C0C60C926007C0C487B5E15E2D3FFC672A48E"  
01608 "6A2D88A521F095109DCB055197DEC67E6F6740FBD63FA5AD6B2A94B646BD30B981493E39057"  
01609 "8C8FEAA2250E7F6DB02FC09C119FB7ACD45A4C26CAFFAF820D6801C3C9F30878C0F6C3C6"  
01610 "EA9D565CD122E5BE2D21E444BEB5ACB64ABBA1E93DE184293E704C2356CC8695948D2BB76B"  
01611 "90251E944062D163C39035C73A5E94F9DAFFB10DC9F89B553B488AACDF2F4D0B517BC4B6D"  
01612 "1CA603D9DC1453B8609FF4C336B296FDDAF8F0376D01D973671A93E7AA744F1A4EA1834C"  
01613 "8743EFAC70137F4B6873008734FB864B9C5232DE56415B15AA42C786ABF2DE1F6007B9563F"  
01614 "9166118B3F1DD7DC40BCD5E0107A56FF3602B4FCE4446AECCA6F360273A111315E86778C"  
01615 "58331DBD5206B9A8568470402F5D8092A42F171BC10A0DDDB1532A7FD4AF79C5D103B861D"  
01616 "07A18C9DA515E51C8176F79D674E5D4032AC04E5AD6384044B280DF9199F92DB83C8779634"  
01617 "BA064601A5EE0EB381E946765F989CA9DE204766BA9005684BF6DA5BA8C2F42DF2A6579A39"  
01618 "020685A375D9A1BBDFE02B90C2977D0486AE3D45D75D2F27836AEAF4AC298AA6D4401C7219"  
01619 "0167D64A3E35DE0529128CACB2367F803D05F98788D9D024E0D88F8A8E02A8566E46FDCE44"

01620 "DEF16EAC95A0944E67F1FA9A7DDF0EE1C89EDF63E44A6810000006FB4916FA7D5C993175F"  
01621 "DB8E27A4D2C405242AA4CC3257FD3B0FFFE430E6C574C7AFB448975A37D6C12A399B7F75278"  
01622 "53FCAF1020D514E724637EC59152FD4BE8A232AA7AE574DE8C104F90FA4661E278A531CEFB"  
01623 "77E6B63EEED1DF950627EA67323D0A9FDF002F2E93DB0C6AB0C2B8F1D9BA0F2BB05FAF1089"  
01624 "C403DD0493A028CE7F792B6C85C1226398374FEE446C6F1FCE1EF617B75C7E3C398FF204C1"  
01625 "783824619F49A4CA9D7EEEA9377B619A9CC8F5D20C7EF89D0026D219B591632E5CA909986"  
01626 "FEBE7C518A0E7C92D3136058D6768529C67BD84664E4FE4D34DB662F0AD738B626E65C86A7"  
01627 "F9090042EB11816C4E74D12E07222B696C8CA460689CCF97AC81CFF14167CEC153C1112700"  
01628 "48B5046DFF64DDE18DA9E6EA2F6C7AEB0DD7DFCD9FE37D823A53B9F109C5F600FA9FAD1C21"  
01629 "DE694335AB817951B2352DF7B67A19F9BA7C684306DA14B2395CDB3F1C1BACF920FEEAD158"  
01630 "E87D88EF665A9C3D3A43025BB536CB285B5012D3A3D81A827B1AE89B4977FAD4F5E8249C5D"  
01631 "6229A78C031415A9F3D1D5A55F2914126F7E53986C05B5600C526A31732B3EEB6EFABEF5A7"  
01632 "C707F057ACCCADAC4FD9447BCA3D5CFA20352AF8EC", // PRN 45  
01633 "03928AF2A832A21DB2E2101BB6EEB4D135C6311C6CA3E15608F112373ADFAD87630BB67809"  
01634 "0A41506810F32576B24789B67C9FF8DD654956934E1290941F7C55BFA8821794EBFF46B5F6"  
01635 "2B28555EB2D34FB60ACA9BEF8149D43227A4A83329ADF75FE42CC48B154077A3946175E799"  
01636 "B698300238F7A5C3389E414462B987FD9587C2FB66E873C12BCDCF897423FD6DA964BC84FC"  
01637 "F1A437441C2D0E62039EBC23424842E0E63CC872BDF00F0B1CDF0B597B0F1D102EEF68358E"  
01638 "D8918E7E1727CA46E49D2504421D45DE9E744063B80C8C1932E626CAA99504514AB5527EB3"  
01639 "87F4936B335421B2678F737EB17EC133A3DA1AC354FA5D885A4BCC5904D858515A5EF2EEC3"  
01640 "A9D6C8A5EBFE721B9831F511A52FBA7E626E4966CEFF6337ED45F2B3BD26B306A69700843B0"  
01641 "B8DE0E0F70A3D480ED7E1186CC0917B53D461557524E245F51CEB64E20BA5D8A26877E6528"  
01642 "365964F0F729393BA48CC5FB3C5A1D46F497BEF2E824BE03CE7CF68D8D91D67D72FBD6D67EC"  
01643 "0902153AE09351FA3D3F24E93FD55F75DF561EDE2AB8CD57ABA3C745CCC6B8AA463B00EA72"  
01644 "9971D665919154D149889B21D64A19267C043493C72CCCEB25D8A5EC4A0957E417D8BCE0DFF"  
01645 "84D786C1EF4C727F72C7CA7952F4E1584AF38150EAD2664249F8C17F93A53A099FA179AEA9"  
01646 "CDE4E5936BC3CE42084AF8851757857EB38F661AC4A2376503DB5614C8CA4C9C9564607B4"  
01647 "41B57905CC3438E56ED0E90427728892EDFE685D857FD196CCD2B62F87D0CE8FCE5B7461EA"  
01648 "51239FF6A1854387ABD9B248F576E1BE9AC34C0F2CBC2B618B9952FA804CDB7048956BA184"  
01649 "7440EBA3068BE3CE42084AF8851757857EB38F661AC4A2376503DB5614C8CA4C9C9564607B4"  
01650 "21781B5503C2D71CDE7F7272675B8B6D8A633F24E2BF4CEAF625692A695E7E7CD38378D266"  
01651 "5AA2466B90733098F754E96E37E00B549A2CDE3C634A5A64BEF38D4510A5C586E9426C3EEC"  
01652 "F0247A5D2737121B84B2CE9243B068ABDD5214A3F7814AFE3BED1623AF9B222011F48783B"  
01653 "356395412F7E575948E6358F53859C9B47745A3AC08498FE9AB2E70E304106081215C75CA4"  
01654 "1C7F108544798F56EC23A0A34B6F7120162BBD9B0E2CC605B42CB3C92CE87285407E6187A8"  
01655 "247DB60A1E207E47D7B7EF39FFA73D4BD87E62B36E7051F744CE074199CD04CF7D7709A91D"  
01656 "04068178DDA51DDCA5B90C9552FE2378101EA6D7BDF9A93040BF2F114BAC01F7B5EA7E89C6"  
01657 "AF9219B12B79B7A7D1FC53AE793B21DECBA9855E5958F9E17ED2F54F1B8B2208185CFA968"  
01658 "A183B6FC9E536FC943BF0F70186B5CF7F74FEC1BF07E80BD8650DB2745BB368EBB4EBF07E"  
01659 "A7873187EFBF36BDD8EF65957458950441C5FEA3F666C5F296359CA500F21AC776C458038"  
01660 "3B66BF2DDEAA4B88C366C52D66DF26735A27E03D92ACCF01D74FE0812FA63863396EE1DD95"  
01661 "E852894D4B4BCA87899086E6AAE96A03ECAD7F77C78A477C2694F8FAF221D5B1391B7E8204"  
01662 "92C039F1B003C4128ACE0EDA2131B2E4D9F879CA6B7229FE21CDBAB2D8BD7B3F9F23DBC185"  
01663 "0445C6A50FE4D5B998BB45B54623076F507584366B29AFA7006DA4761E60C116FAAB0D6AEF"  
01664 "0126A41094CF6C77B11C8A6784196164AF92BA8AFD58BEBA47A3518FC5DF65360222870E8"  
01665 "85E86E65BAB3BDC85F728055584D541A60B81C3438566A908A35036468B63F93B9398600E7"  
01666 "6F55AA4E7A63C49D34382D3B5E6071ED3B028368E4032A528BE474C7817DCD889E1C1DC3E"  
01667 "17521FD9671F6A25678637648CDE6EFF885C419534", // PRN 46  
01668 "4FB5B9E9C2DB1949B53FDCC8AC765E5606735AE1F566749B0268F3B2AB3EAD97D321933EB7"  
01669 "D8DFAB60ABB792815BD2DFC08E3C3D34A0338BF4E7BD07982C2CA71E2FC8BBEA7F915FAD6"  
01670 "4FEBA464990EEA25FDCFCEFF77CB6C153942D3BAD6C14721C84F7EF4600F9A5E07EAB16E38"  
01671 "6C6F0E664F27B80C0F82A16180E65003D6044875FF4B36F85FD7247325EA993E6925982A11"  
01672 "243A115D65B1F93FC715B3A90B90181AF0450C1573C6F1582F7FA0866698BA9F3CBFD6D3B0"  
01673 "5557EA751C9D55F1521DE8358F788C555E42A28EA250724124BBF0F9193D03550C6B7"  
01674 "D77ED523704D0A5E1262EBB026119FC1AF766492DF5EA6DD019428CF3D64935AFC1DA3E596"  
01675 "0B08B2CF68019A1B551F437DC25A71332FBEC6C0BF061A521B8605E1E9B45FB17AFED028D51"  
01676 "B75EA7C66870DBD8257D7001E5EC472FCB247308C770CF49C1BBB89689A1C48212A289FADD"  
01677 "4C48D5386179C51DDDED4832C848AEA9B7FD85C867181EF081F2935F5E3A6FB01122DACE51B"  
01678 "E5E7B9D18063FA722022922EE48D82B69C556509383AF26AAE533EBD536B52280586CA0367"  
01679 "CC622938D09B0A7135A6552C187607640B8D866DD6DFC68C986C8318C3A5C3ED446AE59D0A"  
01680 "A656A4FB0C4037B0975AF167CD94AA32F4485F467FD7BD3BF46FC8F195D7157DB1EAF531BF"  
01681 "E65000ED01844E7F0835929049C4534BC2CFEFFF0554D0639255DFBAF7C9995F03AAC74230D"  
01682 "73DCE1899BF3F30D9936ECB98E3B61186987DA5A9512EA71522CCE9C6273B5F6DDCD69CD7D"  
01683 "B11C2D921114A7F7B3C88CE1234E0177B9202C7A06651C49CBCA7E7A1855BDDC8C0ED401E8"  
01684 "EAA4C32B8472E06EA18D152E5594D3364791563992D8C5219D164CD48390386CE4107D547B"  
01685 "F0445568543ED7B535F8AB942F177F78405974E37F123B15723EC4FB9018A4029742D0AD692"  
01686 "31E16E90026916C34E608B2908217F32F1199619AA95C892591483FA1BB847E144D46EB248"  
01687 "D660E35A830B8E94795575734F043AA9D44AF566151478126D4B0B89F5A0ED6E933BAFAC"  
01688 "360D827129135B1EA0D692735103D8AC41C44BFF0F4952F869F63049290ECB411CF8143E7"  
01689 "8F45B439A18264BCD0C0E1B635FCCB75F345670AE0BCCF56A39C44FDA88EE0B19668710322"  
01690 "77F5D79767E48B286261BF9CD93455CDD87972F6D7867587829423ADE21341CA240139D9"  
01691 "F6F9CC89FDDFEFB1E59E9FA53D5A4D6FE7039FC8C3A8E9CFB89289D5E7B78ED23C97F71B2AE"  
01692 "502C091E216F376294E22A9FAC057D30A061DAD81DB1E342336D32CCC62E434DAAD27B7256"  
01693 "ACBA48BCEFFBBD6318C17AE8E07B905D1D10BE64E2D73FC58B04465EBE84972DF6D0ADF01D"  
01694 "077BEE10695070A7ED156DE96730F489F92557701B85D13BC1622E483D42A6ECD2D142C8C5"  
01695 "9373548F1FFDDAE5F9E08F00EED398B082BE242D705FA6892D2759EA4E4E29B971375301D"  
01696 "96EFC3550795F802D0A066AA085B204330E0BF87C2034E4301C1336465198BC924FFC9BDEC"  
01697 "1346E217DCFE80C6DDDB44E9EDA88C5E44CA81477073D6771690CB73214B8C75D6EAA10A7"  
01698 "69A640F188C840E1A359AB439F846E5CAD3EC0789D5E09DA11017D2296DC4DAFDCDC521660"  
01699 "5E26AB808D2C51E20096DC9FED53608E1D94CD17357BBA29A8B943A07A56D76B7A2CAD1"  
01700 "5063885E477D78D78330D0BCBEF9C894CDBB2FC39303664FE4B27211ABDEBCAD668F60BBEB"  
01701 "F8126402A1873FBD651168F067560C5BC14477D5D0757FF0E9C0C1360F76E0305A17988A0E"  
01702 "AF8E2E99580F81D4F3D56C5BAA3C3CDBABB130C", // PRN 47  
01703 "101EC76558BD3A5C44CD26F555F7313C54093840EB0AF55F19260A5514A0816EBBFF4D7A52"  
01704 "5FB48E98A45B0BAD7FAB31FA0CAC69A6C43B0AF7B6A3CD33348F0061F637AB98A1A0BACBF0"  
01705 "A006BF432004C8054EF8ED573ED868D2AC47E3C6467C6413204E7AB54A07C893B98C3FAD5"  
01706 "E2BBFD920074CC1C0469B281FD949C265D1B72160257723CB8E84B0657A33DE45C2A0D43FA"

01707 "B8BB2C47B57BC7DB419C2547468889D50018158FB94F82A589B1D3F9F5B155DFBDF4F38D5"  
01708 "D5F4C37CD6D35D745AF47AC9EC65A13BB71E71B9267205F35C6427B7CBE90BFF1331E31D54"  
01709 "CB8B30E3ABCB8C65F89213A4A2A386C01EE76953735907911D975C88360591F248670CE560"  
01710 "9A8F621BC5B08E8C2F0ECD645D77566D5F02F972ABD7015A1C620ADB77783636B6498FB8D4"  
01711 "337F8552A68BC8652F08D5581917723463A76C9EEF63F2FE84E58265A851C0F19676A5709"  
01712 "8E6CE404DEAAB955C02D339390B75B3AA9653D1E785779E9C2284707864A68CD271699FA3E"  
01713 "A62963EE6FE7D1B0A582A3509BF66BE479E6AE5A0EEA5114A74809B4736C3425E03A9149B2"  
01714 "2154E6BB9D546A6D5F92FFBAC7EB619A09BEACD7410B594832035D1B96829CC6CBBFF4DA53"  
01715 "B7D6BC133377D2512C8B77EB2B72463EE917292931959BB2CC8091CB958AF5E6E28B2518D0"  
01716 "A7823F8704FEA6CF5E901B3777F72411CE4811E4D64D99E84F10A1FE1BFCDD212EA1ACA0892"  
01717 "16E7839634A1751EAEBA4DEFC2037230C5154A6D8662D670306E8155B701525313DAF7CA04"  
01718 "9CCA57B270A25E6C99B9F4E61C8D849FEEA4356B516BCED68F582E31783CB6267E2A317B"  
01719 "B6D08BB0FC93754FAA84320AD2F6E802E4B1AC6C009D69C00FFD390D8A66D4ECB335395E6"  
01720 "279A755283E9F552745DC2E93E2499E4A8C1A9E2DFB40DC29160EC04FFCC6D6821D1C0BB7"  
01721 "534FE0E56090983C3DD1CCDE18B5454B5C3EB5B2EA5935788C57B421BA641A919EF4A105A5"  
01722 "6FCE207F5084AF286C39A858E7F58CCE05D69BB1DB02ECADF9F3E1E7BD6543E70B6DC019B51"  
01723 "A0580E5F1E1F081167CFFB5F8C7B5A999CD0F66F114A34A1AB6D097818BFE41A955AF2F840"  
01724 "6917091AF4F32373017579B28E5743539E0D60AED6487C74411E82C1334DC0121160F407F2"  
01725 "0CEA14F504D1D52D79BC1B0A2C1C34AC9EE1D079DFF59716CECFFFA2B4E89F606D31B20B8B"  
01726 "5E1AB6ACCE60A391EA284974472EB1D6676A5A1DEAE5F22468CEAC43A4FFDBE6AA23FCFBAB"  
01727 "1B14C7274F25E7FFA8CBDE5EF3E3539ED4C2097361DF9C3FBE45ED0A7B1A6CE116FBBBC7EE5D"  
01728 "80E6922EB23A4664339C2E219C5FBE419105B9079F3A228A4D412D5F73DF3139A751E91A0A"  
01729 "75BD2B8A34430A7BFD3506BF8FE9E2C1ED3CF68D26A141967370FE0782F70A50B2ABF80811"  
01730 "86A6B629C990ADF48E0834CCD478B44C9A685B7D19484B9E82F05EA853F77E7BCAD705F35"  
01731 "043D34EEE2B3022AE9038D9268635A2B71EF912F1F390CDE49CF7AFEFF4B08AA1BD27C0130"  
01732 "AD679D63A5B586488339F86FB6BDB381E2FB70D726F3322FC6E0801F24E34AC97A23F029"  
01733 "FAE5148AD59CB8A179637EC7E9A670AFFD90966C35DED303DF28A3426307CD11F5E605A60A"  
01734 "D43D0358C7666DBE824FC037E87E84D214C92D4ED2B6319727729A53924FAE54E48CE4C522"  
01735 "E97C6A5C70186835D997265E29FFD68B5CA9CE5C52D90130F36BC5C8B206B157FAE7EC152D"  
01736 "ADC94AFBFC9FF9721F5D39A3A2672054381D7975FB5B3BEC6F35D7175B82DC73F8FD425920"  
01737 "97126775A28FD7275F733E65EBEC299749A236B6C8", // PRN 48  
01738 "C91D4F53C54EF5237894D424E84BB8F75ADF73B0F4D3C70751D432591C3FD585A78F29EBA3"  
01739 "EF8B1BF43FCF2B84E9EB7DF6BFF20BEB1EBE3CB8C4B7897252E1452172EAF89AD1750A1DA4"  
01740 "702AA0A532F2E90B2220AD9708AFE71B1EE042ED8DA8C3337985D75B0E250D8AFB24B6F2F"  
01741 "DA32745C7EAA26F8972006EBE2BE4FF164190823A3C30FF5B5FB66DACD921E7BBC903E7660"  
01742 "1CA9C8A87C00B428C6C3A0C987BB5F1E015E479C9CD87BF846914C0FFA96E6C584690782CD"  
01743 "602B6FCA56B8A209D97C0A34FE50216FD8EF5BA3891799DF06AE38443D3BAF5FE5013F89E0C"  
01744 "C4C9A8DBAD2048494B246B22DF66612C2DE8C89AEE0B59E3E062A9F8C7A51B4D01BA2617D7"  
01745 "F7C79553C119055454948F57146D3D665740BC5B6FE7CFE6F379F073BF82562FAC75843A87"  
01746 "3E9F490161A9DD201B4197420A562C8AC27214B09150C7BFD5BD395872FE7F4CB41DA9671A"  
01747 "234127BEDC4F5D9E190AABF9E6DD65702EF88961A75F2E6CC96825FB78BDBA68EC710F5C53"  
01748 "076F9FFC5989C3F0AB4EB89BE340FE4FA38B315C9574912C73215C99CC56F9D9E85CC43C9D"  
01749 "886B6FEF42E86CC7112CA8A56DC9EF282ABE9D073F64928B45A050025B4F44DEF1011804CF"  
01750 "339BAE62EDCA5651974A414DE8D214E06AB357FD3B604AD261CB8285E9EC1B52FB84D931D6"  
01751 "A6BD9D539E4A0401941FDE961A8B732F043ED5DB88E895CE2009E352838CF443B33FEA66B4"  
01752 "F69F8152B03EE6B44B750673ED3F7CC29EB9BE2D450F2546ECCB7E3DA44BB55E47E754DE"  
01753 "F4255988FAEEF63ED79B4CA007E0ECA2270635039A92418D40C5F433C00F47DAF5CAB3CFDE"  
01754 "14F708B2FADFDF70602077EEB72EDBB81F55D69400E1E90F16119481D8D117A5505FAC65F"  
01755 "96796347776C6DC9CA4823C2F3CFF3D48FD10674E8085D22CE74812E6ED4B225D8535B5D36"  
01756 "CAB4354FE6FF27F7856B2D6A592A2A902F6333F94D46665B9CD64E7A93110B78C6F1038CDF"  
01757 "403F8F65F62E4F972F0493E87981E535297D83EDB51A4CB0110CCC226CA53A1BOCF5643A92"  
01758 "DD1B1204FACBF10539DEA656A69C2DD650758C29A008B3B99E43FC45B9AE1882B27C29B02D"  
01759 "80714CDFFCADF5E079116DC26F47E3C9916E98C6A1017474BBE2A8326F05323F29513A611A0"  
01760 "48C8D279E06D71F6CA45844077A5009CF272433FC892609A94C99ED91FC51304396FF282A"  
01761 "FEC7096F2002982EE445C3B2FE3423DCF11BC66F36CB531E141383C11B33879D72A16727AB"  
01762 "8B55B48928E13110BB9604D4209A6B1B1484A4B2A67A35F5B2E4793349EE92287791C928A6"  
01763 "SADC0D238EC4F5484D006829ED66A6F74883233251CD7326D01FBF2A26D1DA98F62AAE070"  
01764 "3C9C70EE11E39504966C7E7F7A9C6369C69733C08A61A123D05C879E05A26FCF31B137E21C"  
01765 "A45B14DC337045B73922CC6BB14D5515EC036AC01BD970E99B68D39A454F83373EB32D0E8F"  
01766 "70B80E7FEF20371E4C054BEC3308EF920DAD6FC7EDAF01C2F40A03F249656246D08EDAC870"  
01767 "EEFB37DEFA6DE812E38EE3479FAE7CAEA76453866D2166DE450B4B88F360DEAC68268FD978"  
01768 "FB50A71686774BFCD213E06D59D08CA7AE9A7C7DCA14AABADA9A9A1269B0DCD9CDB7ED4F0D"  
01769 "5EF15A531745E2EE1259B3139FE29B3566F6E94ABC7C1053923D7796BC43926BC6A927FAA1"  
01770 "5DB29473C377DA06698DC366F8E68B07F1F932668E88CAE07F3676F3864E9480855F1557B"  
01771 "F1413B256B3BDBF725BEE677D286084925EAAC0E428AA1783401043B0EF734FC0C5912A9"  
01772 "780CEDD5CB613BAEC37D623F533AF8C634974A3374", // PRN 49  
01773 "AFC22B3A545791F52DA66F3DBF2702032EBC688D1DE2824845108266F95BEF050B363999B8"  
01774 "3BD69EA7EF3E250F4280314E5D832C58F1D85E22EAB4255493F913AF7B08CE0DC2AB3708E4"  
01775 "A3F779196B339625C38FA87B8AE0077ACFC9F5CD95EEFD10BEF5FE9A4EE310716C4E98085"  
01776 "01BD584CA7C0EE23976D565A9499210A95A3D2A78D7182F49009892716972323ACF295881D"  
01777 "DF44CAC78983A4073D8CAB2D0442C15158CA46D55ED2BA8950F2D85353CFBF6525560B5743"  
01778 "F61833101F37FFED90F1AE204811952000E608356105E43EED3CF09499A8593120BFB88FE0"  
01779 "ECC4AB465B9F3FD4065A2A1951B872C92092A5ABE984E5B24AA606E711075AE6CE179B951E"  
01780 "5B45B54EF728787A82DC82585A82D2A0CF368286AF117B399766A8A4DFDEA87072F18FC0A"  
01781 "847B8D485FD28AC3C0F58B3DFEDE3E076D56F27DFD19D244C7564688EAB0197A61CEDFCF15"  
01782 "98D8818BED9A53F973DD50CA6D6FF651D3B8242655F0F0C9D22A886DD804C799AF860B380"  
01783 "2C270B1A41FFDFAE3F92E09B31C12574DEB4ADC8684CC5B47F87C2D905DB1BADBFE78F1B73"  
01784 "82C35E14BCC2AFA0020B942E1EC369E99CE9072C4CC0155ED1665839EEA138161EA1239AA1"  
01785 "0A5BE5A6526A5AD6905F5C7EAE5866085F64A35EA9A8ED52E382DF5E42D0576008BD378A5F"  
01786 "B0A289C02EFFF02D6797110A7777D54845696BE0A7A78BEA81742F2F98003C55129A47D1D6"  
01787 "2E712A6B8C04B499295E0957A5F304A91AD0BC007024A0734EC99B769CB0C725D875B08C35"  
01788 "2F14C4A6245312278F67EFB74E16461598CF1970E30456C1C667E29885B3B291340514924D"  
01789 "9F726B8A3EFF64CA0248DD03C45E5363FF1F17FB1D98E51DCB74848A028C42E864A3E4777CF"  
01790 "90BA6D14BDCD58250E849CE88E19DEF46476B726DF3A4D4A859BBEC49AF7D1B4F4F2A3101A"  
01791 "FB0D0BCD9082C712C20C7C8DCD344361E3D9698C3A7FCA20BD554B954ADA52AE0710549CE5"  
01792 "7B5200D3795C91129F33DCD7DF9F3EC88E72D2EECF48376A991E08C682CA18EE1D97A6A"  
01793 "8087099882F775E62ABB5A75408C505D9DCE34EFC173EBCA5C14DDAFFCAC5CF77464A22FE6A"

```
01794 "BA0771E13C398B7AE6AAF10CFFC8CFE238FA9066F449AF172933BD0AA73AE90E406431BCCF"
01795 "4BDA4598818775D4D6233CF1E696810552890BD793B104F17BB1C219F52F6028EC1968F9C5"
01796 "32CC723DB3D180261721CD922443ECE10DEAA1A4703DA1393828DAEEDB4CE3541382AFC79B"
01797 "A928E7255920A3577D56EE8192F43B9BB2DDA58294B74C7C859B39EA7426C3397208703DA4"
01798 "49A941ABFBA8C0DD790B6FFBBIAC5F5237B59AD1C259D68E1775C6D001786090F3DF27F73"
01799 "D70A11C2491A461B7E8E5858436718C254DA1511221D0A6C8C88E76D3243E3815857B556E"
01800 "CC9F9AA3B198896153B287F80E68B8DA221E14D620DDBFB2B76DCAD209C6B13876CB044A4E"
01801 "E78FC01F00F915F62ED2355EE9094071AA99BCA1F58A9F1DECEDACDC4FB94C26DC54C447A0"
01802 "8FB4CFDBF6E5861F83B3E9277FB5DE96FC6C46C3CEF41ED9F187750FD51597D5A3D1580CBB"
01803 "3DD28DEBA103E036FC8B2508BD1A6C558849AED46C6EC216F6BB9306CEA01FE33E0B82A5AE"
01804 "7D18DA377FB5CCE75E6E3022D15D9EB3FFD02D22E2530D3D342C561F59BD0989865EEA7F59"
01805 "FC5F197D769CD6C62738646810ED1C94D87B0D47C13124572DED11BEC718C8161C78F0DE88"
01806 "0A9E39257B2DE14241A0A5163F0B3E41491EDAC323A050F7D75DB77821608C341F5DFC461F"
01807 "84E7DA25EAF86E7A7DD99C1A004408DC9DADB36F08" // PRN 50
01808 };
01809
01810 // Galileo E5b-Q primary codes
01811 constexpr size_t GALILEO_E5B_Q_PRIMARY_CODE_STR_LENGTH = 2558;
01812 constexpr char GALILEO_E5B_Q_PRIMARY_CODE[GALILEO_E5B_NUMBER_OF_CODES][2559] = {
01813 "E49AF0472DC849AFD1529E7227F001FC9FCDD82A502640F54EF88A2B201F911A1F90562966"
01814 "D0E4101A8846F50D25B0ABAF49171E3A0C8384465B39458FEC7D8644CD6ADDA0DE47617495"
01815 "DDD186FFD3A960A255D58DAF9333F73AE833D840ABD6DD1D749C20B8EEC0C29E63661B0E0"
01816 "BC50E9EBF4A4CDD0FB5B48E141C16E196ADF48314E28E87986FA609DC9F11307B68EEDFE07"
01817 "7AD6A58F8E7A333DBDFB74C126FDEE4E13E4C879CB19A381067DF74A075176104A3C60FA1C"
01818 "934F01CD20986A3844CC3395C2F14DBF38B4932660051732A813D7D224D6A9A46347702B34"
01819 "2A8A1F214550A3E30DE0D007029BF701A32DE08148C67E341C95FDAB5D31FC4AC790DEF152"
01820 "1BED12156D12E68349B60431261D99176797A9CD9F855D3746D03C575D6CED7E62C401E24A"
01821 "A204A3571AD6316A9F288E02E99C4B29AF9C9AAEDBE83306694FF697B2A894E350A3C2DA4"
01822 "0EF7CB537E8C046368E47ADD96BFE35755871927A94AD9B7C5C70773FA1B707FBAC5483946"
01823 "B33887C82C27F7D86045749BA8717002FCAA21DCDDCFCA7F90CDC9AAA43825374066C254C6"
01824 "5D7EADCD1BF4B2F32E0ED7C4C93A62F00B652938FAC680D874461D9F8624F23E26F7C40F52"
01825 "AD7668AF00CEBB7AAD8A47F144377BB81D9831DFA53DA0AC1D6C1B909FF9708B05A248EDFF"
01826 "820A848B7F94DEC6B1308C2DC6A4473CAA01A4CD4124CD99130899FC85F7D2C7FB1A12E5A"
01827 "93F75DC461F00290627EF320E3228338FAC94318B0D767956BA42A38B58571BEFDADD77921"
01828 "326E5C98F8983BFC70B0CDDFBD8C628A1A5CD6BD015767691E85A3D6FC1702C5FE9ED26DAB"
01829 "3B8148E28DA4A64AF460EAF272E262BCCB5F47209438ACFC0743BE66C9285404C1703331A"
01830 "6420CF3BFEEA100F555C633F91FED3C382E7359ACC381A1E1044148A2BFAFC132833E74811"
01831 "9EE745A927D24B72C1D9296AB2A28E6ED23CA85905AD923757E60528B78804E37CFCFEDDC5"
01832 "5F799946622E9F20BEE58193FB78BF63F53116ADA238C24F44ED12EE018AF62CC20CA45B5E"
01833 "E8D6344D35574F395B065C4BB76074704334B317A512FB60E24CB5B3719C471E0DB67010F6"
01834 "22D4B65400728CB7DFB821D63A79DC10090EB8C5D7E56F939F9E5339C29EE1673ED994F19F"
01835 "72FED83A247FCA65E12F3417B30EDA9C9B917BCBD372D1CE32520998E60E54F1D54CEBD87F"
01836 "4A5598600746D30011FCA3764686972C7FCD3CA7E5DCF79E226DC61C66AE6C70F2ADD41D34"
01837 "407F847582110810BF5F92978A50D5AACBE93F9A3632BEFA257B972D0996A6CAE12F78B8BD"
01838 "787016093BF64887367B84559C215B2C148403102DFE1D143F6724DF7078CA2EB855FCA5BF"
01839 "334010810EAA5C03094F2CC9EAB888B8C88886D7B1D83AB723D32FD7B6F80F2565FD7E087A"
01840 "4F6C48354C0FC63500DD7DCABAC255BD92AF6A6918B3159EB776EC62E6B4AF9A7A397C75E6"
01841 "963B92515FC29F1123577B069B07CFE5F01E986F28A6E0883793F79F41822944AE6CEBACAB"
01842 "4CF4E2534CC70EF682CD066F4EE142A55844C104E1239F51D69D3C6BA1ECF2720243C340B4"
01843 "E04DDF68FBE8D5A07B26F4C97583422B3C3143C2316BEA8C96EA365055AFB39B719DFC45EB"
01844 "B2A52F52CC67341DE493D5A788E89671B705A6DE2F92FB0ACC6C31E200EB38E689F64D24E9"
01845 "02C8AE95CC75B22BF9F44DC95130F9CE91B4F1C7EFA96D32001DCACC26D9D3F55B0C653D8B"
01846 "09AFC1F7DAF71C721D55F82F89E3DDE413080E8334898A5BECBB4B6FB2CFAA0D2390753F08"
01847 "62BC626CE011783C6240712939C93DEC36235A8", // PRN 01
01848 "CE701F7B81A1EE81DEF8354D759C27FF6E4C6EBED2153250472ACB70A82E3BE39DF9DF9BEF"
01849 "543B6645CC6C559565710BEFBDCAB8654CC62A8B60B6A90492EAF1216B47227F0B088A43D"
01850 "4E3B5E81946A7F49A14BF917765460C80543EE8867C116970B2957EDD5F7110654AF69375D"
01851 "673C72580640146C7A4F640F6524D3CD827342BFB29E5A211FC373C94E11F16771D58D008E"
01852 "DF116230ABDE98165797E8B0CFC589FEA57788516502A664C04AB9F15310C55758CADCD63FB"
01853 "E743F9DDE95AD39DF253A669B1B5286C32F379AC7CB91973AAB3FA7759255DB068D225FD8"
01854 "5DD5D8AB38484F4D09ED5AB3CDE5924851848C695EBEE8717B03919256AF0F0463CCF030ED"
01855 "39C5B38A5F4E5AC680389FD80A7FB83868D09D18EA4BA07B19312BFBCD33715B8CCF23800"
01856 "60BDC7850176AEDB634F21BC15AE9BB76A879C76F10AD8A09D4AA8AD2E4AA1186D744E2A15"
01857 "B0A107C3921C10B007A0D41FC89025A9CCA8E62BE52E136611F6F0B3742E115F29118E09CA"
01858 "13B078752CB2D6A3C2693BBEA490F3BD7882D397DC69F50999EA74BC3C276F6951B0E2281F"
01859 "A3CEDACFE57293030BA83D0055D492C310BD8E3B74582BD98CEC33847266015E207DE50CF"
01860 "DC1CFD5130B76185991C238E63CC4AE2E14DA55851C405EF1F38B7449C1C626555403069B"
01861 "DBF7D5CA36355EFC82F85B3CAD343BCF5B386E6974DF401E130BBD16A44F282F7E354048DB"
01862 "322D9CE029718DD62574CFE8614CC891AFF3905968FF8BBF0D6FFCAD3F738FA3B6F4871400"
01863 "7024DA010FF6430FE2B4C9BE4EA2CDC2357AC71529F4362D9A792D4472E14682EE124892B8F"
01864 "186F3978104964EE7115E14D991D51E8B4FCD92792E53CEF163726760741FCAEF963EA169"
01865 "B20F37D5B9627F62A503CFDA123AE58A780875171854D4D2C299E7B0E8377CF040313090AD"
01866 "9C53B4C79C7E49BA110A592673122871CADF826AB15B3E97275AC1CFE3948EA832E2D0EBBD"
01867 "317F8547CF916453D57A7A18D39CD3295B25977FDD73949B1E752CBCE0C285DDCF38FC515"
01868 "E509FFA2721880892BC6B911289C3D437BE6F58E36FFFEF9557A520C772E620597C691E85C6"
01869 "A63212CE7D1641A41055681A83CBD1F61024C6C475DEA3598819C7BC33F798D0E0AF481568"
01870 "52F9E262835607D496AE6A4C1261ED97D3BDC8D63A93EDB484089E228358AE37CB5EED7EF4"
01871 "BC2825F6DE89F79E4F8E5C8483479A3A63B0B0A5D54F5F79AF2A8657E167D08DC73CF1D912"
01872 "85DDA0B1277477B4E4B348F4167443B295730F4BAFB61DADC7F45D4EE9938D195F76CF1FE8"
01873 "947CAA68A6D59F2469DF27FA3DD5068E7271C55B42F882B54F5840E5A4FB09DAEA73ADE905"
01874 "604018C722CF9E1751C6998275BB40E8BB254A8948C13997BF83DE5E4640E8D40300FE6F70"
01875 "8BE7F1CCC62063E24B4DFA3550CDE01EC28FFE7FC55F2732080E3FD4F9D06D0D826D49A130"
01876 "2AEFF95D8867ACEE2E13DA40F29175030639DEEE713F5A3BBE9488349E594A5B2A27730B18"
01877 "C33943A89552D17F5CE967B05C55CFEA5E23230F49862375B8C6196650BB9B1B39C57E1878"
01878 "A67EB2C59D486CCB2D0AED2ADC102B5D0FD88C62CDF3C45221A76C09FA43F2924497B328E3"
01879 "AE725D108EA1907FDA8C8F75C1F53225DFE87C3E136BED8DD491D0BE615598C6D6DC0C6D3"
01880 "3E844E3DDBFDA299B1E12C11819FA83F9BE494A081862FAE5F687485E2264D9D7F0117CA3F"
```



01881 "E00B0EF59E03253BB2C5989B770951F16DD8B15E9D485E163AC60EBA10974F158D0F121C92"  
01882 "74B15FFBF3C4B639F04D32F3A03CC2E6E32A7B4AA8", // PRN 02  
01883 "54B709AE2BBF27FCB2730AE75AC1B3CF690A17A43038405FB0B8F87E2C9A2242050998FD7E"  
01884 "79B96CE5AC5CC9B986632BE6DF55BBB7BC0483EF2F1B27593C0408487E8F773A75FC51DEEF"  
01885 "FF6581423D0155440563B04767549F4965089102A0ABE280DE00E9F92804D386E2B84BB9AF"  
01886 "5D7C1F0565875E04BECB19C156776E75C73A4855A5E18DDF796B8E87D7FFCDDC51F3AFE9BE"  
01887 "FE8DB161FFA8D6CB37044EF3E1906667BAC68EE3CC9A1ECACD94C2FD20B6D8EC553796DD9C"  
01888 "BF3A4F81B395F7C2C8A56CF8B6CC27069A352DF4CE53A783F94823C233E4CB0E928335C744"  
01889 "B75649E2518B8AB8C6E09A704DD025B73FE9FF962FDBF4FCF25653026CE0E66EB947A33727"  
01890 "643568447C91B9DFA9623204FD4F127A708311CD25F7458E97F408F76FE14AB7267A4FE4AE"  
01891 "E0D28B6DEF4D4309C55FBD6886E1372EC8FE61DFC89DEEE121898DE2EE7EE661F436BED885"  
01892 "0E8C28F569D2CB2C9944D4F88384C0907A8076002E5A3E80F09FF52517932D78A647455B3D"  
01893 "5F1AA72F9F70697617ED745DFBD402331BE6F72238C7E8B83C6A7B0DBC617D1C4D414AC016"  
01894 "D337ED5043C0B7BBD46237949DC9E583C580A578FB36062F5BC5107E5D15508188FB8FA983"  
01895 "A01C556F472C489173A1A219EFFFFEE9815265EC1F0E87ED545BF55CC80AC26F5A17A04EA1C"  
01896 "8167ADB4E97194D515B23B3DA5F1B77719B37F5FDDACA0671965C8AF2F900B0F1A26CA56EA"  
01897 "0C3A801E05AE3DDAC42C5E975E7D8D65151AE2499702CA49B905F8FBDA83225E363680E76"  
01898 "B3C8442006EB644B19B01698BB63DEF284DD9C6B78E500EFB4C10AFFE3F5B19B755A9C47A"  
01899 "6394551BE0C01F56DD60A7A0E6EAE7A57A74096142CC055F5883BE9B21010B7D6B1BF525C8"  
01900 "0C25DAF89E12A8C9BC8B09B7A796319D5562B0942421E6BFECF59A891C28EFF59BE7D93BE6"  
01901 "A9AE3CE6BAD0F057B009C3C83B5A70A546984181EB121D23ADD7DC74BF93118AE200344E2"  
01902 "0DA0496A930215EF923DF384C69E222B4D6C95D3E6698171205466B042E1EDB58F786172DC"  
01903 "DD960CF676992ED7DD5B8D88C8D4A3C452D30782206032C4D50AFAD089531863F5D3D12D7C"  
01904 "12790E6D68C5AD406317A97CFA2970A0F5EAB7A6AB48F6B246BE79AF1C17BCE38286190CAD"  
01905 "21C0C0A2EDEA69CD82C0B50BD9CC9BBB321D1AD292632EB0D3793D6DABA3CA1217B25C7296"  
01906 "4464DA5E6A6731F1CD9868E4678122B0FD38DC51418BC199BDF6185FC9BC6C4596A6383A540"  
01907 "9D9C442006EB644B19B01698BB63DEF284DD9C6B78E500EFB4C10AFFE3F5B19B755A9C47A"  
01908 "D34702497837AD5E095A4DF9ED4C018DF813D5D89E04BFE1E62424AB17E9640CF7503DC9C"  
01909 "03A85BC094B3D3077F263E11CA128EB8C14A4BD4F26DB8FFB2D0315B38EC76B46121E95AD0"  
01910 "3ACACB851DD792F877056DF5736F2B200FB02B9C26148EB88EFA66514C03ECD57801277D"  
01911 "77744D6D77C2AE630DB858A29DD219DA84A814276F7A03BD0C4873CBD12C883353130DA044"  
01912 "B887DCB397750C94AC5E047683D457E25A3E25E34D7C6E8D5905E2B04153ABB905CBAFBD341"  
01913 "3ECC4C022754C05C0C0814EEE8284FB399EF2BA34020851E1B233C1F4095302F2FCB07BDA"  
01914 "74F1483B213E74FC9D1A1B0FB1FC00ED8B59CA2EF39EC3099713C704D7D4CFF0D83E95B1B5"  
01915 "2BCB0A6B5AD81A9EC988E8165C8C1BF36FE3CDD1F416FAEC8E8255ACC758194064F4F75BFA"  
01916 "AD22F2D000DEF719C09F10297D2D5307ADD524E23D44C36F1672D1482D15399450F46EF33A"  
01917 "36466F34B15795388CEF2FE5C060F32F337AABDBE8", // PRN 03  
01918 "641AB1974308060B0EDC824D9A911F7F95EE18C74C95D97E6998D1D8F2137FC6B95615004C"  
01919 "2A87747D65C96A11E58CAAFAB597FA901588E2F66E0AD827DD2F3CBD3322E9D9B700C67406"  
01920 "F154C3749C349958CD199D7759C0236AC94868B15DCB0C85230CAA19D56C7BAF7A761C61DB"  
01921 "809FA99E63B394DDAF76564353339FC270EF76F4734BCAB1938397014D334221E05DC5C47"  
01922 "55BAA521BC6933120825875F90FD3CC65651582B7F60BC3F14D7C289472776CE8217694BF"  
01923 "8FD204C0298C78C6BEC7E209666210593F7E0F78DE625CB684B3D4EBDF0B29F965FA295EE"  
01924 "C2F0CA520053DE46514DB165A02C1F218D4925D4AB4EF5ECFDAE4506DB184A181606DE65BC"  
01925 "857AB778549AD8AD878D11A0FB9AC44504359D85622BA6BBAC2F8CA3FF4FEBBC0DB0A5A9E21"  
01926 "14A1B58D34FF416BEECCBF714BD42269CA84853864C21F3DAE1153390EC1BFAAF3BED24B70"  
01927 "D00D335DB0A4F528922A2A17F188506407F2FC7782B33DC5D87C7F0B4B635CDE1252609650C"  
01928 "994BC046722929976E64BAF5C2C094AF27BA9BD2E4CDB0BB67158DEB6B5E8A0A1C8992554A"  
01929 "FA3449AED56D1F0EE41E924B907507BE3BD7996EE08991F1D55A8F1EB9C949A34B7031DCFF2"  
01930 "049166489D05A3FAAE6B4A1F1B31607BABFB74A0F92537E1BF76CFEB1F76B56CD21E1F262E"  
01931 "D59BAEF09E800FA1A9B30DF7702D8CBF40AB6B19DA2D4D6573653CDC092B7B858F8A4407A"  
01932 "555CBBCDC7D6CB8671032A5DF38801593557AF5E7BF9B1640A9BB183014DCDC068E172224E"  
01933 "48D04023273F795E3F1B966618CFC7840A06518B3CCA1987B186EFE749E6EFDE8F1C85F4C7"  
01934 "57F5F628E986B093C2F46060ECBD40D9C8E547CA9AD57D9C62C411DD9E484C2228F34E6105"  
01935 "E7A94C26932DEA29C2437C1EA011D93919C029E08221F633B4941EA1AD3B8562A5E0499D01"  
01936 "F236811488D9416F11EAD0294A95D268BD988F946205EED0B98897DA28B21E75361F5F0880"  
01937 "C48864D14B674C91639A9FBA0F1B0D8BE5073C7E9FC157688E48A14E5868AF395C5514BF30"  
01938 "931B0EDC45EC49C5D3A4DA377E129CCCC5E284253FFB7133662145C3351BDBDD0BACBB370"  
01939 "DDBD193C23BDE59CD55DB29E54FBA2D7F890063E272E2DE88BFF0849B04F8F24D1A0CBE0CA"  
01940 "F25F815BDD87DA4318740654014E89D12683C0C2ABCDCA3456F475A9D79D7FB6858914DA47"  
01941 "9FD11A2F545E113293DF5D8B84D0FEC6E49601606025D8CBF071A4980BEC07E8CB3D204754"  
01942 "3D56FC32DA90FB77459E4E9C7048526B31386FC907AE118ACCC050403F9AAF5D530CFD27B9"  
01943 "7238B91AFBF936F2F3D1E6BD5C8630D0C6DD92ECB212D87FCC314AF7F0FF49E50A2BBF36E7"  
01944 "D19AA5C2D785F1926EBA6922583E885307ABDA921F99C44565B2FF4135CC3C3B6D11AC2DCA"  
01945 "0E3844335F8A5C3387B76795B98C468A48E87994B110A3B3BE39DA16586171F33E1162623CD"  
01946 "4DC3A77C3A887B942A7D792684F7E3EC9FEFC08276241A2D0051682139D23F864A31358D83"  
01947 "415F4E2DA5AA16660D74C00539564A0C791983EB1BDC5384C975DA16605B9676C61EE54487"  
01948 "7A27F5E6FF1D1C6F2329917706AB2FF9CFD4FF0543E7B33AB7D573467041C88A83AEA30AEE4"  
01949 "84A6A2F2C73856DEEE40101D66589D52F44867ED29506376AFE0992CD3318A674B055EB9B9"  
01950 "70E183ADB46D8CA41EC3DCFE7CF14033FCB1FEB4C38DC18B84FCD85DBF488A4316C9C7635"  
01951 "12D6058C2659645E2A2AD79294DCE325D00B9BE94B78AE970FF25109FD7BF9DBF42BE0EF05"  
01952 "8B2771B7BB7455D5905CB66EE700A8DCE17645074C", // PRN 04  
01953 "FBD0AE56FBE835762FE2970D976200CE1798C3E9C416B2974967BF06D9C937FF318888EA7"  
01954 "D2763BDE785CA9054DA3315793EC4C53815E0397CC274D510A81DCEED4B3D5232D8F19043"  
01955 "E567253F0B717D6E7CA009A01592BF499D4BEA95612D126D46D1B62E2E2100EE640D55398C"  
01956 "AC05EAEA7045AB515A8189C50940E15F2DD574F9CC4A3D0D3C0B3DC11D9FF2DD0453B6B8D"  
01957 "8C2B8B3BEF35EFA9DFDCC9A6822EFE365B0D1A7E63A86B92A2FD34FEB0364B41AD17830AE4"  
01958 "9025EAD29AC32C1C6722420F5F1C8F9570E3C39D9EBAE407EFE9EE4D54BB70A5525039AA38"  
01959 "CC109582C46DD190DBCF28E94557986C53724C64CB02487DE67ABD26895D3286154D2DCB91"  
01960 "D66DF17D6FC53E3F4B02F55F9DB19BDFB744D9815E35FB285E331664E99A41DE1278C051E"  
01961 "BE8EE81904F80712F12E92034423B05AC4003121C55AEF7AC8807EBB0461DF9A6F92027F4B"  
01962 "8829BA4D81A79A68653BB132D1F02CE72547FC7D36B9084FD1541EA69F6648A0F2BAD4D45"  
01963 "C95FC0B9F34962E8E4A3D001D515BD1133476810647DC79989E7B7AB0603E9FD57E15D137"  
01964 "34071A136D2856040E09ED9C6E13AF417CE477635BA9B7D33C6AAE826FC5CE4490AD53B537"  
01965 "FB8CAEB110E81DF15BF7FA679DF85F2643EACE2D60FCADBA01119792AF6AD86C88E508A678"  
01966 "4BB679851EB1AE071492A8E963C13BD7914CC02F4836DAC1CFAAB94E51DBDE7B9D3E6B4DBA"  
01967 "96DC8E7A38B8A5C58E724F417E01549BA56DE0B711233A370D51EC73DA83EE1F8D92B343F5"

01968 "B1ADA8CADEC9774623B0C59991E9A09E42D563F69E54AF3A6D286F88050E01E562CE23E45B"  
01969 "8366B9847024C96E6BCDF756AD0183FC8725738DE34C3F6C9054AFF0B17EB7ECF36FD08505"  
01970 "00E258C3AA9BC4D9E988083346F0B03C8FA394DCA5AEA0F8E663520B726346A0A13498E3CE"  
01971 "45845A2D6C4A8AA14E1488916B3751E70C18C98617191F785181E7231F2DB8C712A7DBD034"  
01972 "0810F07DE493C7ECF8D6EC3E9DD80608A31D0F4E0BABAB397704DD4EFD12D8F27541FE02B1"  
01973 "Q4973D783F18223147D12E09F5A6D0127D2BE9C5EC0F5F348AA61F534D6551991A935CF910"  
01974 "DCB5DCB09BFF6E8DE3F4C17E2DF1CCA2411BD91D074A3E6295CE365EACB1A3FB61AC2AB85B"  
01975 "D460B01E290D3DD1088BFB15F7FAE3D534011B7F2D985DD84C82B6454FCF80BC03338E0DDD"  
01976 "2726AC8CF34FD8176FAA5650D29EDB15A3663CF28BE7AD5D0B0A92F429BF8CAA807F12A910"  
01977 "896987006161C0B3C9A5ACCAE1E1B2A7CD791ADDB02A9C41A861EE6AD2B7882D84A7E4AF7F"  
01978 "1F8D15CEECB25C3770ED938F31A8F6ADFF298D7FB7C90E96677A7F0A78DE6F4443C61E357D"  
01979 "3FD2E9A0309B65185E02001ECB4054EBCF7DD0D3D4926A1C02FC39EF9E160A81CADEB6A43"  
01980 "922D46C25D9C15312B8D634E72FA49EE822B748EF959B0D5545499FA75DA1BB24F48D0BEF9"  
01981 "D423AFE2BDA14E66266BE6FBD5754C36881475BA4CADA0E2A8EBDC6A25712A55A6D546CF42"  
01982 "2726AC8CF34FD8176FAA5650D29EDB15A3663CF28BE7AD5D0B0A92F429BF8CAA807F12A910"  
01983 "3DEC5890D3FCBCBD2731E7B36EE58E3FB2670B301BA7DF1A7135CC2BFBBCF052D268A6A636"  
01984 "465DF743F2886B985F8DE35B8428D6515F32C6F9FDFD3C04D3B2816D416E2906FF7814E8E2C"  
01985 "9178836C2573C545AA976C5DC44DC87CA242360861C97332AAE38ABE73D556460316FC335F"  
01986 "11069CB25D3064EA9F62BAFC0759F419EB889849DFDA8393A6B32E1BE7C4FA7B93BCE0FA2"  
01987 "1898D879EB0E3AD4CDE61AEF8F0DFB33E408E9992C", // PRN 05  
01988 "0D8BC96A69914CBDD029FE74ED2E51FAD1714E703FDF5E34758864D69E68453B640813982F"  
01989 "4164AE4B86CDC5233D544387215FBEADFB53862BF19DFEEA39279DDA4205C4371045A62071"  
01990 "14E151B055FC03FCF21E8E71631B249E8466568F1FA8D290D63AE42121F04877213FBC758C"  
01991 "8D1D5B7D7814C9778FAFF5CA280D2FE81CAE909DC0C4D29315E7713F8E1C64DBD1D91079A2"  
01992 "6BCE900D29AE70B8820264A4BEAFAF6135EBA6172C887B5A64F7AEDCE15C85E4CD8222C3FF"  
01993 "3C16B6EB153F190EE23BE455B48757086AE6A5FB7474C01A11469120D8C5618B315777950E"  
01994 "4BFFF35BA4AA015CB3623B706B81F3DB5CEEF23B93F111724D787B2CF9DC9CE02E1CD35C"  
01995 "60D86BFF8A68668F2C8C53123869547889C72BE1E54F261D3755A9D5EDBE95F72FCCE57643"  
01996 "7787E3039570FF6F9A67B675D57213EDAF80C2C59F46B3EE7B27E54281CA4DF4581AEA8E78"  
01997 "0A89E46DC3EBA3DE258F38F963405A6B0DA44C03CC7ED52043252141056F4BFC4F57EEBD6"  
01998 "F556D369166FDCA966702B8BE5404653CC31CF17354A9E6C9EB13EC892615380CB6F3E891B"  
01999 "FB3B4E39B191982855DB272117C758E5191873E537EC9DF004C4C9A637C819C9538FDE6EDF"  
02000 "02F900748CDC07B1CEE9E2071716B4FA6F5A2A9DB613B9B8D4E9E0D5B65C73198B7C9BC20"  
02001 "14DCCFA3A8B6B4564D2CC0352DC85BA2FE18E2B5DD9D348E45E9B17C1A2FB24D1297217F06"  
02002 "A5FC96480FD6D7438E1187D050E5ADB681448DCD2AF16DF7558D90B06DEB83259C2DEA3EEB5"  
02003 "CB8E882AAF853556582F423304C62C1794B72E4C4EF47AFCE04128065F8CE6CC0CBFF0A9"  
02004 "F493804E7F692945DC98677BACABFE1C28F0E041EF081D8A24428A5FF845C912D034573098"  
02005 "401C292516BCBD1362A72E44D0E0A88ADAB019008F4C09E82FC8E17DF47CA600F090ED9392"  
02006 "086C37B65C18B3F86DDC23A1BD6CCE98E5F671CB183F90AD3A0E90F0FC4DFC3CAFD9CD3C4C"  
02007 "D3E0C37CE41067317849A58181E079726120095A4C2A71001037CB003DA53F566C9C47473D"  
02008 "82C5688CF9D0EDEA58EF17898DA47BEB7CE2BDF7FAC23F5EED9D4F31E810A2658AFBAE2B9F"  
02009 "92B4FAE99169584FFFEA64AEA47A97E3A723EBD291E054CC790C9FB62661874C61432780DB"  
02010 "76C2C667B78DD6FF2639E1BDADC7A7F010AED1E48EE6E1C9742B137ACB7C5B06477DC55CCB"  
02011 "53F400904020422D93F60316761ABBFC4BF664836CC676EA8C818576E3A505D80A8E750F"  
02012 "456201395B6BDA0A1E180513EE17AD80EE3A06F8B36CD3F9A6E041DEA6362B11FC89193454"  
02013 "BAFA0A01E1918BABD7616D9366B6A33E890F9CD237E43A551766DCEC80FAF344810AABB9651"  
02014 "3EFB30A72C436C22959C7D9D9A9F26F5B24D79CA7921B75A9DD8DA4B068089EC5AA09619C0"  
02015 "C2F591AF17B1A16D796D61A7BA98AEAA12F9F444CF695C7AD6C826B7E82BB599925193390"  
02016 "08E065F485153FBA0C6326F58C2092AFDDE5BEF1D11FE65AE008EE249EA4CDFBAE7BF084"  
02017 "A59E38D81FC0C4AB7216C84C9104316439FF387909CA66A7D88520C0F2137B5465598B9386"  
02018 "9DEF41DE46347B20146A1C274E6E908D2CAD97517EC7732EEF179F69C299F227B2B8B385D"  
02019 "F5DCB6103C95AA3C9BCE35C46330AEF530541953B77C2BCE29C1A2974C6DE07E0F3DB4754"  
02020 "3EEF21C307274567532B150A93342D1FEF7B4675349089A1CBD395327E61F2579883D4BDCC"  
02021 "1BDAD06721F1502E46267893BC76560CBBC4A1BA02525AF67C6D023C14B38CE284333253B"  
02022 "522E068C97374D6B83635E70D9BA88796342F5B998", // PRN 06  
02023 "805FA5A4A543914EEAEA33F40E46532D09CD4CC4CEB1E5987770B405EE24848BC2170B7B05"  
02024 "72D058E726BD79E7DA005F0D6A11A083CD2BE82DED9C5355C254F645B98EECD37C0F0834F0"  
02025 "C69E1AE681A69BC2A0F108AF1205DBB51A6D5B5D044795DF198A94E1C4AAD78B680C84A7CA"  
02026 "148EB6698DA12792FBAC8FAD677CF086899E147B645EAF7D8F4BED2C7D0DD69DC3F0953DC"  
02027 "6A6CABC1E3B9F72C07F3D0DD0971D9BDD11C861016D0DC73A946D95DBF07248BFA896AEF4C"  
02028 "21FF57D76F894399585EB2AF7299472ECC6D5386C9E219D535D8CF522737A44A3C416BE882"  
02029 "49CF607D5DC7EE411734AB878D5E10D61F7A2B501F5754740DE92397F69A40B8763642496C"  
02030 "B141EF5CDD8AE9454F6C5639EB70D302E92C7ED592EBB3EE0F6F6714E113403726E838385"  
02031 "3CE8E59C3400C0BA0B335B6212035BCE33A91C597009F8BB933BDCB3481D2F790499923B4"  
02032 "5470FD6EE3D2EAC6BA84B686752F826D7D0804BC53E677DA1E5198497A91EB071562ECB42"  
02033 "E38B3F5BBAD0A47AD06F4F6972B8126618E773C35E464962FD4E338C33273A167189C8AE5D"  
02034 "7A62D78302189F57FE04FC420843137DD8CBFC2896174465DB4D178256170B7DDEC0D8093D"  
02035 "0951A5C8A8B25C94C318B56E7E2DED2FE39E49AB6CC5D3B3425C82142866EA2F1C1FA55EF"  
02036 "B420578D7601F90C14832C7424FD51ED9310FE62AFDF7648E18B8F6340FDFB50B6C34992DC"  
02037 "45C06F5E887A511408FE836464BE40E5F672E09E4FBEC8FD486B4313BF60E2F6932FA9BED0"  
02038 "ACD0A76833DA40E811EBBDAD6CE221656303C7E881F63E08E74DBD10054C30864CDE812016"  
02039 "293A419A32BF5400093A3F9EB405617B0B6AF64ADCC70880936CF9D4EEB17BBD8E416AD981"  
02040 "3F1D08BE078145DD493EE752D3954BFA12136FFAB111625948C5E7EDD13B2018D226B089EB"  
02041 "4FB71698BB7D3B9B7A1E445CBEAB19DCF146375D0FE53D24978F7B5D7222660BBC5B952117"  
02042 "06E865D98EEBF4D3E88BA9137CFB74D18B7285F540E2770C9DC2BF7F667FB3C7708AA4B3D1"  
02043 "CB8A71E374A9BD7F3AD6CAC883536A78AFE696A7E446F766F1CE5FB5F6F7438F2FB04D09EA"  
02044 "0A27945DB95C29E334F526B699C9758B2654EEEOC1D1ABEFADFE7BCBB4E3472129B8F8F0"  
02045 "83A63564A6856F5223B08A6FE3181878C64F3C83DED97DB0DB3F110869B7929BA2F9C3715E"  
02046 "1717779AF88A061FF91ADEB7F1E293E822EB38947FB6DD1026980FD33E118FC9ED9C69AAF5"  
02047 "3ECC127A92CD2FFB025DA5C639BE6E1BC25D4950031335A2E24F5DD034563700F9F8C9BFE"  
02048 "35B1250FF96E34E4BAD34A864693DAEEE7979E405FD7302C47DFFD3EC860BA280EAA116811"  
02049 "4CAB3C229F1B76BB5B486534C7E3D9D1A871CD69AAA1FECDF7D57FB8187209878C240C9B9"  
02050 "4CF9224ABD20AD649C71B850E91836A293ABA64F12E0859CACCB8A0F5E23DF38FAA483AD9"  
02051 "231CA0B797FE57758AAF7D8E7434E7BBEE4C8FAA8D403FF1ECE06443C19EC02632CBDD0F15"  
02052 "498B8CD6F01B23D6300171C8CA10A87C321A2891BAC887299607A644781A0429657F5139CD"  
02053 "2BB2147EFFFCD8DD18D16FC0B3B20DC1E21306DD17981E5AD27D26CB43272089E40802D484"  
02054 "01CCA1225E3F55678C5093C19A65C9FB882C295E3B3DB5F3A94EACB5B0D9C960FD7913A997"

02055 "904610121D033BB02EED60754B4E7C0E6C241A95E0A6BC7A9958F8256AD9F2EDEA9B6D6CE9"  
02056 "2F7AEF8BD5D651BCAA6483CBEDEF5F9550CE2941D7164FC01ABAD744A0E04DEB0B2EE26226"  
02057 "20A593D8F8188C5701967F70A2592E6E3D8DBA4838", // PRN 07  
02058 "D86BA0EC5F2F711EBE19B487C0A3DF6D083DCA33E132E879AC4116BA4028CF2D131A6146C5"  
02059 "3532AB0CF8969C0F71C8CA2D62AFCF9D98FE3654B54F1D09DD01A03B1D56A0D49F0C651DF4"  
02060 "A07B50F55242CD2C3A4FBAABC116BEA613D3B7EE5C12D4EFFE45E2C2A13DBC3D893497A10"  
02061 "70D089232BF38A2546F381F080AF51D7AE4BB9A7B6FF7D70D0D9FD49D91E621E8BBEAF4B06"  
02062 "929A267922923B9A31C952B83B60D6C667C5B177F3196F0AAB6E8081F6CC5F222BD17FF2B5"  
02063 "4C9BE6E51DAFEA96999A7321C458BA3D87E2DC1651159CA8784B416F003B525CEE296EFD41"  
02064 "2973D8BBB41084C9215C49D95ADB95D8F763D7F1E34CDA773A3B3071454E7ECB81010B3023"  
02065 "2C4CA479AC5A03E7C694A8F27F73E4E8C02D48F9A31F6262E8EDE318D0EDFE90973C5A53A9"  
02066 "CDC46F5318D8B21C9D55A8FE997DB6C2A76BD2FBF3B9EBD5FA58A65279F517DE527CD91B96"  
02067 "17BE9C8065AF3B0F141C9CDC76B54CB352A249EFF27F581E592A46F96FCCBF8DFB6683737B"  
02068 "8B3C1EAC0E772CD0103FE3D98E5C1D05644AEAE2876660C423C76E088F78B7EB43DAF6B6C5"  
02069 "B3F7742B61F71E4B94C6BD65A1CF726499DA0356773C709096E0896194EB9CD737CEE94B0B"  
02070 "A86469D460C871A4FD8C094DD26D144103590A9F4BC01EDBC94D6008DA48391A851C19E374"  
02071 "B2249C5467BCB68E11A77BFDC4C9EED771C5CF16CF8CC3B688A7479D0F0B99E54F687E869C"  
02072 "60618AE87B9B87993D9C58829962D4588426B96FDC8892FFAED3168401D8FB01F41F591"  
02073 "36C9286C45B352303141BBAE6520EC6620B645A010F520846959C4810190A7D4E6AB27A3A6"  
02074 "78E21557A5127F15E4C6050806A5BB398FFFF75AD33F1E84249455A241F006EF15A02506BD"  
02075 "171669CC2968E6794FEB316DA723C25C5BF0F068873EAAEA4BF39018026F1A8A3A73F9B1A73994EAD14"  
02076 "940514FA95EE3982B72B511BBCD1FD3B5D19C681807DE3FF7730229675AA0A984985143ABF"  
02077 "DF75FC7C350D8154738A6FC37954C5A98E1B672837BA729D3C645FE6E51EFFFF9414CF0D4B8"  
02078 "26D331020D2703D703C25C5BF0F068873EAAEA4BF39018026F1A8A3A73F9B1A73994EAD14"  
02079 "9C8A3CE889208C3C3949C48B679E499D3DEB7930828285DE1B5E2D09F44C7121508D35CBD7"  
02080 "598FDE0F7B4F8754D77E048C5DC53219CF8275B72410E79744E582911AD3895790AD37F21A"  
02081 "608EE385F7A30DFD2BE9B10B1C5B5A0662E54C2531E54672C99AB3647CCFE1DF7DC50F9126"  
02082 "FCB4E71627E75B31AFA829FA1926ABF650DEAFB8B3AF7A6692C10A3888C7795C3764949F9F"  
02083 "12E0EA1577EEA3EB29AAB0ED64D852A36CD3D7C9151DCF64F00649F9F0299618B39E70CB22"  
02084 "45E893C7808FB17B511FC2487572977A7068B5AA90E6518A603410A4C26F9BDCB529CC5F67"  
02085 "41C660AAC3E37F7F4E40F32142331A1E8F3AAAE8A9E58C717BDFD237CB29AD5568F1D0872B"  
02086 "111AE18F361D4494FB609F438F3429AE675B7C59991F745F7BC16BBC1581AB60928AF522AE"  
02087 "91ED5A801BC4FE8F9845E38627B01D14BB2FE8814D440D78C8EE4C8AB3771E48931DC3B7E"  
02088 "18C7736DF123DA6E3BD0C303AC115FB93EAB2E2AA4B3A1496E35E575664E3A17E8C7C6C3D5"  
02089 "4AB4E9FEE8754E85228241B6971F7863044AF204F67CC72505169F72E3FF1B8C9AAA8CE86E"  
02090 "A01BAD7C87B78BEFF6F721F5FDOA61E558AFE11652BF4BEE5664D81F8E2BA453B40D8AC90"  
02091 "DD57E1A59F8F827156A65602DAF49BE12857E811F8066EA48A9014F501BEE091F8E28ACFA"  
02092 "7E20AB6613641FDEFEAAAF38B8BCC107EEC4244CB0", // PRN 08  
02093 "A7E921AAEFAB06F89B802EB01EB3A09871F66F0B85C95860BFBF78E7CB2F078BD27BE0948"  
02094 "B42145F900CD2DEC1E08062960A14CEB9A8310A8C5AE0E2D5F5B86A0351C83103C03C26F6"  
02095 "7FE5C09253C11AE0DAEDAC78DA3581CA6F2B51EFBC187B0BF960B20AFA083D5A447E08F085"  
02096 "A71F7B628838C0187EF71894AAF78ADFDE312EE590D210774839367783B7C8A538B408E9D"  
02097 "D208FA7A9EA6D999DF408B5E41637A7EDD0DDBD69BEC7F9E39D1AAEF7B36AF6AC977EB0CA7"  
02098 "83F50FF42482AECF9C88725AA45E70E3D7E492AAC74FF351668A7A2FEEC6EC2B3A08156504"  
02099 "983A493B0A418A421172513DB2F3FD9057F42021E12170D3815EE52B5ED0F1A49896C532E4"  
02100 "78AB6D497585AB2153627FB1A9D87EC0DBC0621CC81477757ACE26BF6C7518CD3629D5A74"  
02101 "08A85E1EAA4E3EAC99F2DC0A2456DF0EE69AA48A2B156539C0B2354F6857503325A0BE4C6B"  
02102 "901B8D98505AF50577FC1695E039DC219224248D3F4C50C1A6045D1B5B02BC61F86ED3756"  
02103 "E9BFB1F5B15919776309452E599C6B737EFD8B8B0B1BB74B5F4EFD75DC98A4BF88BF5A4A6C2"  
02104 "0DCD155E8D03E1519E24EC3613F30094EF287B27794F03C519C5E7B9113BED1F1718CEF5C9"  
02105 "EA6A12B1192296DE7D806C77854OA35CE36E4CFC9EC6411B267118D4FCA89C8FD9554EB90"  
02106 "262D49B727E83928DFA27EB2FCBED2392901CF09C7E3B14EB8795A2E152063058BD213964A"  
02107 "A5DD8EF99865255AC7CD5379017B96A6A379A736CB1A5D041B1F2CC43D27F7C234DF3FC5A7"  
02108 "E4596E02111268781A30BD9B3A10A15716E7753BFE3EC8013D4ACAFDF6F2918D901CACAA5A"  
02109 "A53FB3124666D570CE1D7EDDF64065C4B5EAE52789E2D3AE39BC252095C9B703FE57507EA7"  
02110 "769BC64341B5EA6BF320520E02D59493453837A219BC9851435CD6B8A9D902EEEA658EA28A"  
02111 "C5DA822CDEFD1EA0DDBAD07F0C9165742CF2397884EA700F67F64F4D49D8E886E4791C3F"  
02112 "8E29CA77FE81EFC90522ECA5F77128D1D64B89A9A8706D5C84846C9BC42094C402ABCF7ACC"  
02113 "B147C8D299224CA0D6874CB30E3BFC23D77204F47C84E233B646822166902F5BF3070F839"  
02114 "0188F3A2418C7EF1BE68B7550E4AC21E55DC9D7596B163958C1BC93F45F5E5F61CDDFA9E883"  
02115 "5A0DFA339D516F310A62337710BE7307B4CB2DE0088AA4E35713BD1DEF2456D1CABEE7D110"  
02116 "BB0D753E85BC7637D3631D0E169055C822137BEFE8775BE0CB00DE8CE917AA1A851ADC8415"  
02117 "1D3D6B8F7DDDD0F57B9EF563D34D16E96CF3BA9AC589474218E5EE251294573FE1E3A0F7C"  
02118 "06C380062B2F6F6C8B79B3042F1A68676348E81F1BFD90B43438462F7AAB0F9456E5F786FC"  
02119 "7E812387C260685E7E21A300A09793B9B1B87C3F1178F5B6BF6360EE07F23D30293C5D4"  
02120 "30822496DDBEECC8D5DA3437979904D7F25621D703CAD1E54BD8944ED1CB802767062B425"  
02121 "95AEF82EAEED03F26A4D1D492F5419324B8B4862C13BFD4F4A29A35D5218BB2D5BFFDF85F4"  
02122 "F9D3D0B5E4F096922B3500E7A1D26AFB4616DC144C540280DEEAFB721F01B109EE4F5856A"  
02123 "E534B1ED4734F880C9EA836EF172A8F2566FAA576A80F2F4D670AC7F8C5945C016A16E3C92"  
02124 "795AE445C836C6C12A3DD65C64B0844AA0488C109610FDC88FB21791663ABF41C63B3B586"  
02125 "B1A120E55E2DD35191370BAF43BDA644AE9D7E47FB0623CB653F579B96EB22732F833BB0F5"  
02126 "9B217F700B8195E9CA4CD59393D5824EE98E5D13B865B1375FE7986FECF8B1592077D6E6A"  
02127 "446F420CC9E4CD934F75FE3769F1EE29D0545F748", // PRN 09  
02128 "067E55DA111785828DC1EFC60B788608ACF2928466F425975E29FC11E3EDB16B37D93C135"  
02129 "5B0AA7D3B0D906C0213105F1BF8A2AA3ECCB1AC1720CB783FAE25B154FC507B725F2594E1C"  
02130 "671E20D549718AA9308EB438A71FE15D7619CD2EF7731CB309DF97E0404BB10AC9D645838B"  
02131 "9501F0E2879DEF1095E726F855AFB15EA1723E1DAF883A7D2E0AA3999E80B0489E7589DBD"  
02132 "9667A99A1FEFCC7BA74DA490F9D63623C51936A901A74DA1AA2C3242055C62AFD9671B75D1"  
02133 "CC3D7D9F38D9F867E7AD87E888A0D85EA1B452B7DEAB98D9D34245F1A60AD639EA6E9306C5"  
02134 "85A00D8D026B0BC2D302E27D621E9D0273BAE782C2A46EB7C069C0FFF44D98047482B327D7"  
02135 "1D43278C0EFAA584FFCA492108FD3E84D83E679DE5EA5C6AFC7CBD3F503F027B1C689F8228"  
02136 "83139677ABF4B2344DF7F1835A1E7C3A86C459F0F3ECD643FEDCCD5CBAA4CFC9D9652CCF3"  
02137 "1BBF4D1E2317B05397E17850B51461B53AC475FE2BDB137EDB825EF548C78B80F5B08A513F"  
02138 "87462C94579968C36262A97058C613C4232BB9D4D9D2CC939437845245FBDEE1518EF2F6FA"  
02139 "53AF66F483464F69133F70A24F0E9439F7869ECC80FE008DB7D19C4631ADF628BD659FE4"  
02140 "C6753B96DA4DAE510215A37203A1EFC62D6910317F18C16BCFD993514D3A3AA843463D914A"  
02141 "D15FF3D69ADB9A903015342ABECD807417B3A980201515A8E4A05CDD19B55C055ACFF5F341"

02142 "6617CC012CD9FEB5FCA3210291780B8AF0ED41EE70DAF229CBA8B7419931AE4CC02FA1F8FC"  
02143 "C912B330BA6F0B771223EEDEC12E7F9C7A00E774D107B7F85FAA4C76F8CC578C885578D99C"  
02144 "069F96D1A238ABEA068F9CCE949177A6E7585AF5A414BE9FF7AE6E2CBA84AF6B329501D556"  
02145 "317EF7B3432003B4DF8401A4F4D3CAA38BA9DC73B66EC98693A6F8699AA3E1CB5DE6C17232"  
02146 "DA689F469F1CE489C930B5AD949127454FA133354C030D5C0E61266F49F3EC7FA68F88C71"  
02147 "A6510E8FA2F2177A7BE2DF209534D42D135C5B420DB82EA37183B64A6C30DECFC65DD6F193"  
02148 "0D2E13DD6BF36F3627729EA06E898A4456F011B0C16A1E800FAF35516D7AA1B513FCADA1FD"  
02149 "82372615AC3DF1339BA4D7BCC9AC12E5091672B179A3537A05418F98C6C614F3C88D77FAB4"  
02150 "779F7CCB36E5DBACC93C9F1133A7C5D1079B7155A39CA81283DA09A6F6D0B0101B4F2F0BE4B"  
02151 "1DA059E9739CF2AD5644AB8A18FBECE9037029DEF778F689EF8E0CA6DD61476A9B6657F79E"  
02152 "8FAB1F7C02ED687D1A7F1E4B2312A13A9D0B33F08852249849D9F61AD40177C049099B4153"  
02153 "BC1CA4E454E05B7E8DC0C8DE5BC80FE64132F4BBCC35A0EE4B4D0025C5C3507A82051DA017"  
02154 "4023C956E040AC37CA5473D73B0F909EB11E4010FA349CF5023C590C9014F39C53925E4D71"  
02155 "74E5715A3935ED37A36250662CE41BF9478E92518082C34688D485131746BC848A96D5C797"  
02156 "519F73CF9D13821181EF17110C1F12DF1D4F9EA234ED3A7887322A9C7E1616C7DFAC8D3089"  
02157 "302298D482B9E81FADE4F5CEE785EC9F4510903BFEFAEDC54BA621BC500E6D3439684243D6"  
02158 "4D7E00F1A9947F7FB41236AECE4DAA7B03273A47EE2C2FD872685FEDA7DC87F0C5F16342888"  
02159 "C18C81CA96572B5E3979DDE44D546BA0D3DC928B56479363AFE814C59DFBE0EAE1A323D02"  
02160 "ED77A221EBC4F43E4EF360DFB50E0E5B60A1A9CDFA3F72AE81F1A55995FDE8D15E38BEECA0"  
02161 "B9384A0665515615DE46E11D3D96D199E32B83DB7CAD7F6A6098634D8D207628450EDC8ABB"  
02162 "BEC233E23932D01AB80F021AB10059C9B6565E73C", // PRN 10  
02163 "3E4B585767C32759C804C1CE4E2B0E4EFE8624004ACC2BAB24CD0C40A72E6E9CBDC0F51DAD"  
02164 "8D976B092B0659C1F8E96F2CF6C2F1D7BD0C46639759AC480AAD7BA80785194FCC778EFDCC"  
02165 "9B760B5C0C3F4258FB4F6FF76BB5C7D81A37A04754110BF2CEFF14B8DEB0B982F03A906429"  
02166 "08EBF01F1374B39CEC341347ECEDB9EF4A5BD2B4699E6C77ED4341BCAAA32218B92384DA89"  
02167 "A8FC91E0AD32DA754F4C780F9AA179E0EF9F5D1776B250E41AF20ABB11F3820A976E3C92B"  
02168 "3BB784E5122BDD2FE5821C2F1DD5AFCCOF7808120D97ECD467A7456F05ECA0D73AB772A5441"  
02169 "6C80A0491ABE7703E7C63B39A071262DD1A1B7CEFB57B33F190C1A21D86A6B73021BA4A805"  
02170 "B4D78CE1E1AD6613843EA175410855F3EC96133C27E10C8C3BCF12E285DF6BFD2AD6A22146"  
02171 "B35965DA2675DCA085B3CC6D817AA1A511E29414FB4DB132D9F123FBFFC49BBE3CD0BEOBF"  
02172 "8736E75DD36680303A51071467F782511CF994A178BFF0F192416370FC5D8AC58568A57F0E"  
02173 "3D33F6365A035054D783C833E2C06CEC5C2D5A541426486A38C28BCEB8360CBF0269B1A433"  
02174 "3BB784E5122BDD2FE5821C2F1DD5AFCCOF7808120D97ECD467A7456F05ECA0D73AB772A5441"  
02175 "0D432713153409032C02C3AD92B5BF0DF4BF5E006156D05AAEC4045EA32EFA8A3CF32D0931"  
02176 "70C887A64536AF7D346CD90FDF91321D1A63E658DE26C2779AB934927B39776B5495D83B3C"  
02177 "893C2E2B9CCDD599C39A22DA4E536DC286A54B87C778A548E4BF543ED5A561CAE61DC49016"  
02178 "D1E95377A0DF9B3EF840F698916A0043C2A4A2A49826CD040AF15699C428FDBDC05FD35AA9B"  
02179 "34811D9DAD9D84E2ED9FA2A26BC94C27981B4A0D9B4E9495919DFBC2A7B3D786945480E7A0"  
02180 "7511A453C2691656A7892E6A4C6686FE93BD310C2C25FBB9005B60C4E4C92C0E4DF6310B44"  
02181 "95A0ED2D8347B221B5B6FEE8F5B7BAC922AAD32E4D245B5D752E2DB8BC5304D52A0E7EEFB0"  
02182 "BBCB4A39AAD896FAB9926C6045D9B4BF371C9CE12E3D30C730B4939EC6ADB76F8049A0E4A0"  
02183 "68C55918108C61938BDA830CFACCA00550E1431EC94359DA47F6A05CB82F242DCFA6F79A1"  
02184 "098CEEFCAD8884BC468974AED85B2A2847E093A45D88FB27FA1E7602653B083C70BB9B7A2C"  
02185 "FC3BBC4D45A89EFD09A22628D8B039A6247D46E35BB37E60423812820BA711475EC497D822"  
02186 "7359C4E8955C8319572F47D419A1945E5EA628BA9F114EAF8F565FA95110D273411D01A7A6"  
02187 "D1C96CD60CA84DDA156805771411AF3A4BD4CE493D9DD95DDABC03F830B372976AC302B7E"  
02188 "F0038D93978F63F1F25507DBDD27EA11E2951363F455D24304B324BC59AA73364BF397DACD"  
02189 "730F85D730F8F25E3394F797CC2D38FAC26991DC238135EE55FFF9C81CC9C828FB8287A"  
02190 "B72F33A28D8936D8980C221DBCE786D5141A74893B426C6B58631E96B350E50A7E664BFACC"  
02191 "7CD4AEFD366EE11E89FC4454F721CC1471A80CED1BD44581E5D2AC177028D0C6C7E559434F"  
02192 "38B2BD1766679685657617DD8554FEFD17A22FE9D30C80F485B46AB07F43755CF5AA13E827"  
02193 "0A9B3101AB0196246D87394612DC980FCE24FAF4D10AC50F67FD00B4EB29C29E8957BF830"  
02194 "DE3E9CA028BE889CF0BF3B836D5542991B0F28229920A393F57F6565E0095821D296B3AD4C"  
02195 "C50F66D9B7870C8D303353AD74BF2A7A5B67B265FF7315174F67F883DEDA2DA30CEA7BAFF"  
02196 "05519109D7301DE9F45C0B7663FCE41F88ACB224FA0F2FC954533E3EF633D568E9C201D37F"  
02197 "3D7DE60DDE22AF138E4C90C86CF6D56177AF8A23D0", // PRN 11  
02198 "7D82FBA232DFAAD30DCD0F0A1748BF5E0B0D80C8A8C0B4313AD2B92C8E9608363707A5C28"  
02199 "9C771D8772CA9C5248C496C5F0B59F84DE2DD2F6EA681E7BFC6AB426679B19E53237FCA8C3"  
02200 "1EC5534695F19640521213084FAFFB6CB9E4200EC253AE7F7E781311AA92DB5420431E689A"  
02201 "ED416AB4A84419E6CE801A6A2CD67AA16A653A29F7F9CC4D26C8ED3A00B5AD49B06AA282F"  
02202 "40233EBA84F944E8B5D8316E68B9854A4B69A4F160DD2562EC4A20CF5AD65D8656E70FDA9F"  
02203 "35585F026F69CA9661D908B605DE4AD01F3383C157304FC0FEFE5F8F23CA2BA9013C9C0914"  
02204 "CD3948BFBE270BE3CB37575337CBBD75DC8E8A38F6AD986BB8DA4F3763CDBC2534050B812"  
02205 "86B5261025E88A6319E2F7B655D37C9B2796594B68AADE3AA70B0DF5E2C2E135C4150C8877"  
02206 "62431A194F1081A8F77834697C994F546B1CA1F7494E6507E3A8BAA3F058E510232E93F408D"  
02207 "0A912E196D2886D478872F2E926CF032468EA7CB17C28CF6F520C805D97EE714C275D7F49B"  
02208 "695CD9F76681A575D7F82B95F59EA3A3AD8DF471011D6BAD17C5E4E93F26EB88373143F2DE"  
02209 "87865B40A335993F9838ED9BAC0873FF70D3F00F9D23496B0D4435C65A6D39458F269C1A4F"  
02210 "541B0C1E32BC9DD3A24B8330CA81BA450ED3BE5035A557296021881EA2DE7C7D137020E8EA"  
02211 "416A3808497D09AC3906E4CF9873C1F6C44803B075EE575792075B8C2EECA9120C46966723"  
02212 "DC6A508F4481D27AC0F70BDE28B627D12227FC97A42D564C37AC312DDC1B8D746E488A50F1"  
02213 "1E9F701237B4E38E678D2B42F5343F09641887FE3F33DE1D40C6C8CD655A7600D75CE72D93"  
02214 "19C286836322F3B56C39BA78933FC19F593086BBAD74CBF0A38CF4B9116AE0E1719D339AE2"  
02215 "5D48781A0912BF87F9C3393210237CCE19B9803EDE13A37B69A9DD7AF46A39ED501B47609"  
02216 "062ED00A8FB1A4F3DDB3EB2B950B020FCB7DB9A67B4CB903A2ABA8B6544AF874D615F0EC87"  
02217 "69D8AC8A7EE6E50DEDB152FD84FF9A46CC7643A7C5CE0CA4107A462812387B7A7879B8AB"  
02218 "73D03086EC46F276D903781C3E8040CF79D6662D172A8D51C22E97B8FD3B17CB84FDB27E2"  
02219 "629777B1D94DAD7E73C94C8C9B954BC6C1BD4267223C2C93EBF9C297E049C19847FAA419D4"  
02220 "3364C1B5200A9377390F9E11B007185198D2CE499BE3D2D0BEDC1DE1770A886687E4BE7EFC"  
02221 "FA04403CAD17E2B04115F4CD03437F63D49A9A6F5D30F8826DA3EB010012D0DA0363552EB"  
02222 "7C578F14F03FE7331B594741B524632683AEE3790309657E71EB398C7E6BD92C8BA073CB83"  
02223 "98ABA40B59DB8852FB6B140DB2321C8F3285CE1F17185CAE4879B86BB7F42422BA02224A"  
02224 "9B998D18AB11BA768ED42D0F3ABD270AB08DEF356CAF62B2B27289372F7B6E695BCC85C1D"  
02225 "26811175AD764FFBA8E4695C26FE09D1627366511BE86A6851A1D395F0E56742EF65557642"  
02226 "68AD3FF9EAF9B3B4FC5AC7E4A6F998D9E60527F8F6287F1757727655F4BDD1D6C81D4F47F8"  
02227 "6B2ECBB8271AC1E8439C8E1AD5F9F8F394D0C4DD5D956CBF102E6F8389F9669CE7197BCB7"  
02228 "A43A46FCD4E7FCAB99F407A6C96EFBE2D2714DB68025B917F40534ECE2ED6DF5670DA523D0"



02229 "616757A7A0D21B692E3E202BB4C8DB3CA0E95DA502216543F0D0D41DE560BAFCA5CEB63CED"  
02230 "63EFFFF5AD6FB32BCF66734333A12B246DC3FB58E66E3A7CD48AA1378E1E19AD96BB1515C28"  
02231 "3AA687581F888B5DD5FAF88F5514CAAB8D017FBF7B3CDF4AA040228ED8C42B6675D9EB39E"  
02232 "BCB7533226B0C9622AC1A3EFB6DF3E0AAA65622524", // PRN 12  
02233 "E33BC2492ADC8D8ED6EB60688EA2F93402ABFCF1FD2C4DA925CA4DFBA610FDE285B00BE829"  
02234 "F79B438628B22B0F9EC5F2AD990DB9E4067EDD7A9BFD07A1873CE41E39F96A4F3C72D90FCC"  
02235 "09DE6F7196FF96968DEFAABD80523EF8CBD20559C8134740DBD3793CE68DF223E32C26E7E5"  
02236 "788E6F3247A1EE456967C2524CA4D7EE9441A3B59EDA8084D04297E0F7C717480D590C90C0"  
02237 "F769A6C8F7AB09D607BEB8899E1D6316926F57DF5E225B3F1AC45C4991C1F7EE25BD435A4"  
02238 "5B2A24926CF4E6C5F53FD00DC5716700A8A806654169C1340E6062D40E81F69BAA41FF061C"  
02239 "DE6A0D44D5F9C08F2C213A5A0A0342B5BD4E63ADB1ACD89EF259D735D59CFE01B79E2E22DA"  
02240 "14573572F4FAE831AB5D259BA3BE4D5535DD556425B2B6F6655AA604A527D0C5C680E5575B"  
02241 "C610EB52F7562A41E66897381BC522964BE4CE47EE0575C9B9C47A871A3C4C687308C5699E"  
02242 "22377330ED2CA5C24A662502D6B655FDC76A8E72C124A38A79A1584BF10623EC9E062F47E5"  
02243 "A96F5C9537D59B0C02AC0303DD5C56AA5725D9115760C3E1DCB2B6834F30878550080EF4"  
02244 "9D6C301EC9001F541F09121D0E58D0CBBB8E83748E8009C7C1996C02D9D2ECAA629713D86A"  
02245 "987BE59D46C603F68A447DFD8B155CC549519D3065452FD32FF9A298F4CFA85B8C4947529"  
02246 "1FC65E2B50CBDF1F695A51648E7F30220DFCB8A981098729DE1034D7884185C5FA6612FDA6"  
02247 "FBDCD4812F38F35BA3A53AC1271B4EF5ACF5D6A3D2C026524B2772A64E8DF987117830D2BC"  
02248 "D86E6DOC11C4B785A3857266FF84D36133B07F7B17016DAA576786E2DF8552E2A893D33F95"  
02249 "A0B95CF7B145729716251EAF5B0F9BE02C768C04C52B25244366D2522FC80A305DEFBFC9E"  
02250 "EED9B5110C8B580D7AC7A4AAE4894D95B634A29D196D85B726571572EB21B652318AB5CCF8"  
02251 "BE4A81B1F2AE8A4605006DA8DD19A1CCC76CE460F6701AEE323CDB04907B7395E5EB710F6C"  
02252 "CBD2F4A935ED085B0A0DD5DA0E13C8575D45B7FA9682488F742CF3BE1E3701A7C310735E"  
02253 "A22519C0BBC09675DF4582A6012DAD8931586D572F048C7290952FC72799006056EF5C5C6A"  
02254 "AB7EF5A47142EF1A6FC76504D99A383634EFE796B306E0DA5CA3D68D589FA417E6F6BAC025"  
02255 "07247A37BE475DD0004E77ADB571F2E39D72FB0627F62862C75D95652BDC0AA2320D991123"  
02256 "3E6F30C2797961E538C6D54DE9655212F3BB6BA5E319ADF99D5F62E53218187BCEE6F6AEFF"  
02257 "766301BAFBB65A3EA83F4D069E001A33E5B4280AECF458A9784ADD5D84C6A8C271DC1FC31C"  
02258 "EF0466D6D6C3A5834F96A134C0933221C4CDBDEE1E29C26A260F6BBD5C3A420E27834D850"  
02259 "18A23A311A81703C304CF416C1DEC0CD8082EF60138F5B983430509B22378903579816E262"  
02260 "B48C0F2FFC9F8FDF63C53B9AFEBD8A05D862573A73D4AD6E085306D8A538FF17D2740F72D4"  
02261 "88B0DE5EFAE6C7FA336917F9C33C4E9AB30E6050728CEAE0406198276FAC4E26FEDD93C8F"  
02262 "23A702D205D2EC8E8FE2C8D7FBAD1FBD4CA9BE9EAA1A0C22B059115CB21376212B3BA36FB"  
02263 "283FB474A0BE7DDB5ED346E58B2BFF5D6CF15EDBD50062395ED5E8189995A14B171B9F84F6"  
02264 "28E1995AE6E6B4D9F420957765880455F7C205FDA8C9F4BFB49E5489D82980A9D8058178C"  
02265 "E9826BEEB37B178751C2DD687470BDCD6FB170EA11827099D0429E6ADE8A243A510EBD6100"  
02266 "B26728F66F8293AF30F6EA3E7EDB10AAE5E91BA71F21D7A4B342D1CC4FD365B5157CF5A783"  
02267 "C67AEA038B850D71D7F1DEC056FEBB038F4D4CD84", // PRN 13  
02268 "31372C958C1ACEA7FC11056CE94F76AB1E6BC0F0A8D0CD4D8FD3469D1625CE7F2D815952D0"  
02269 "09D377FF41F58C0F0DE977E6826CB7EB549699BF365DF4BD9E5B2F4B0CAE1C2E68869C67DE"  
02270 "525E187F0ABFE847B6D365FDF5C529A1C68628476A01D5173E9EA883B884EE7731B3E4BE08"  
02271 "A5AC4817EF2ACEE72948AE5295FF1DEE5D96317ED325A35CF72A8A3F4009EDCC222A0426E9"  
02272 "C559B102D67A6FD3C4A2258C931477046C9557B040B6B2DF4B0706B748F1FC4C1B9B8CDF38"  
02273 "C20C3F999BC47B2FD7AAD8B634647D0DF4AB0423CB76DC2E4E06115B627E1072A0C4BEB1"  
02274 "5843BB16046FC01BC2BFAFFF673ACA8B938058B820715FC3A03A75019D6150FA3B74D1922E"  
02275 "B67433CC4D3B405DC0EB808D7990642534A9B1555610FE9F253FBD4B37AEAC59CB4860FB0B"  
02276 "B3FBDD631E2C42732B66DC0150BE1C0A1AE889C4C8B4F2CAEEED25945BE356D994AC59E59"  
02277 "6E19CEAE72CFE49E004C534558E85B238864AFBCAF4E126ED47A48F1FF3F27F65AF1F703A4"  
02278 "6609F3A87F9209B14C8D8AD45E13C5FB579D359C9E0E2FA6AFEC0006D5557F96ACF1149C75"  
02279 "A7DC581458F4C694BFA0F513E9C51BD33B8D627960DF0C022F0F47A68EF43D2F35090C11"  
02280 "18F4203E03D037F4B9F3D6F598A02912683EB26FF4E5F4618AC1308572054A1DB70292AF7B"  
02281 "170B94BF7782DA71499CFA8287D916E82BEFC570FC192DABBFDFAA3BCE8AAB130B072D661D"  
02282 "4F1468FFBA745600CCDF84DD9CC4E9B70EBA7DF86B6565F74537282178B064F9FFD7339E8D"  
02283 "6D9F80176046E01CCFE906D835E0F17FC4DDCD15D8CAFECCA563D6516716C415FB316315AE"  
02284 "52639595D5A13FC509A6972C4662181A0AE1931BE41ACA2EB9DFD54E558613DC704B6FC3C6"  
02285 "6818C608158FD68AAB756E5926469FABE491645A95B2AB2038B15526461E2ED664DB5E0A"  
02286 "21709B27B84D2B9340139C554A30E81F76B19B7EDF84795C8A2CED2E43468A00C221D4BCCC"  
02287 "9230ADBCABB4B8BF647618E09F637351A1CD15E9D5A156604BC4190CD37999902FB56DB55E"  
02288 "E6A2850366CC2D659BAEB09690E62E01A5288C4676A0E0C5764B179DC69CB222DFB3D071A"  
02289 "D9AF9B779E3BB00B3A47C45DB863F028F216BCA4F0B4A089111EC43651E71B0C1C921CBC14"  
02290 "36B75170EE0802D2389BF32FFA22EA21AFF0D8F8DA347E0B2DC6CE58EC59F38D67DF8573A7"  
02291 "532D5B5A40E995BF58BBFC35AE3DD3E19441AA79E8E3B57E5CA5B3B5612984D4BE0F4D6992"  
02292 "406F082D499520A1AC3505A98DC0734D907182680434FFC2849884CC785944071A2F49AD34"  
02293 "6F83E51D64BDE61017747B81834A1D8C55AC2EABF59B446ACEC048E1B9FDE5DE103E5508C6"  
02294 "9B96092F126DA219800DA01AC7CD45578605846482AD5DA52D2C7868003E302DF2CCFADF65"  
02295 "D42EFB8667B0D589481EC3EFD3D8CBB15D6B7D3853CB5D42013CF78048667DED7743CDED6A"  
02296 "F0E0C89C1EBB19BD7B64BC55BDD523822381D99618FB5DE6AD2744C099003B2EBAF231DAF2"  
02297 "E21066D6C18FCA8ABC78A2FD75A9EA7483F1017D130E917F9CC4EFA65D2E27BB35580734CC"  
02298 "798C8F47EBC9B13051E545682B2A0A3F9CFB8D1E6FEED2327D96A5D9AA9363D3B678132C38"  
02299 "FB3B2BBD2415A742C452BC5A5DF0BC8C167A1D8DAC3A60A636E79AA6B12109BBB7380A10C5"  
02300 "14AF30BFD91BEECCFEFD9A4AC0F8C13B543860FC5AC9F4FA711819DFFB4F5D51E9DEDF47B66"  
02301 "FDD37B20B839067F3D1F7723B9A6AAFE27FA813576D44490FC7D64EB7D35AD828D185C5F85"  
02302 "4E95D92BD59502C0D8DAFB337CB49823E474C82E4", // PRN 14  
02303 "46676F28487E3A66F6C502CDD84171BFDA4AC74950AA641F771BDBFB32FE637520DD326017"  
02304 "D9557893B96FAC52CCA65099D1F0171F8A9422CA6F5F24A7398525B65592DCEA2AA8F726BC"  
02305 "08AA8B40E55DEFFCD8AB1F212DB4264C61A80F59866AC763F15CDD280592CD91B22E824B3"  
02306 "92C429E9E52FD4A07E4D6A8C1326AEEB7E03A3425996AE394EDD74A1AB8D3D3E4AD7529E1E"  
02307 "B867870699C1B832AA65A8339024670C56988FB9B42D431CCC0AA949890B83406381AAC09"  
02308 "A6E034D7E3ADA9A0BB0F62BD936B281181091AF7C053715C04F1960DCE215F7CA4A662CD04"  
02309 "5E69EC493BA6C694EF97591F755F5A22F0EC7979CA2C575B158E6B621E4DA9909DDE3F5E04"  
02310 "28FC9ECE064B4EAD13638F375EF7FDCD3A1B53BE77BA7B3CF17DEF01CB07166E2089437191"  
02311 "E003968B8A4680C387A709D45D0E0C40BA85F1DC55B1BD75C7513622415336D66FB31ACEDC"  
02312 "8078D86B5489E66744767F85E08BC88329C58D3EF091526762D0DA1C686A4F9820CC4B65FB"  
02313 "5E02EEFBECB6CAD6322E9083854ECA1F9F595F32AA5BA0A83B856FBD150B30E064D658D0E"  
02314 "BBFB2F01E831F05EE59235017416C927962A537EA506C8BEB014BFAC0CC2807BA62ABCD360"  
02315 "85ABD620979457236ECBE3F815D6FDB0B37B7741F4CB68E7A4B579885A09E9744553BD3CFA"

02316 "AE1B2135793FBCADD14A674E5D42C9B72A361D1A21506570CB8E6D676C6FFE08C0F76246A0"  
02317 "0754760753F181B31F23ECF61581A77FFF71BC23A284A1CE88FA7E1DED0B06EEBE10131DD5"  
02318 "F364889F59A35FB6474ECD80474182348A41E206773BCC41488FFBDCED0D243F1F22AF78B"  
02319 "664007C5F2F732C45B980C14527724FBD6122A78F41AB60F0436158D2DF0DFB4DFA0ECD12F"  
02320 "DC5B96EFF1490E64905CFF97230CB4BAE3E6BEFF9D00C7DCBE6E8586B141CB7B49CDED4D0B"  
02321 "D9B0406DF95492C1C4FABCF88E283D63B7855FA79B4616E7A58EF87E0F7F79DD4AF8B16612"  
02322 "BE175ECBDD2DAF30B6DF8C4BF28C249236B6E10181E8A30B3CF0B17872BB92C0C972DE1A83"  
02323 "3342EC8550721075378AA1FA4483EC77C31E0C87B987B8418E9DC315022941345AE94FD9F0"  
02324 "ADF99EA0E060E92776DB4DF7FB93811891BB0CDBB466436AF538F9D70EB47FEF93DBDF8191"  
02325 "822D50359E58ADFC092186F5582F701938453999998E787405B9C8BB6C9407099E73707C3B"  
02326 "AEE08904130ED6ED91E7880F069FADDE26A483439ED81A9D999B3FC3362B8211E468F25B36"  
02327 "B93BBE5942A96B831FFE33B5080F0AEAC25F9508DBA5EB924B01AA916FC99EC1FC7C149C7E"  
02328 "1F0EB5136C2D15886CD5ECF5F395F0B9B8A85746D0FA18D8A7CD98AE6103D609CF404C6089"  
02329 "F91A22411C378FD1FDE28BF8ED7D517CAA3ABD675A7276C3655A30626989927A93E3EA4C1A"  
02330 "98B70004182B0E1A206380F94C27B2114811E20159B193DDE84BD4D7535168ACFB372C880"  
02331 "4AA344CC931354CE7C56A2DEF1DA86AD2B6AC9A8740405DD2C5807D0613F6104B094024F1C"  
02332 "FF271E3B674E4BB84A3CCDAE9528A36161890C52DD62433C1DCFB4B22B46C890ACBFAAC61"  
02333 "07E64282FBD9414927AA8D1A403CFB693480A8EBDDC22C9AC0E0F519B78A87B8E3CDD1B3E"  
02334 "B3F711C9F4732121054F261EEBF8960D96B86D3EECEA8DC6A4C3DA6112028CA5D3793D3511"  
02335 "BE32591F350F98D9DCBC4FE6CAF76CBBE62AA7F33C3E6AE7558263FA35D82B358046242432"  
02336 "5B2B2867773304A76194CCEB50A71F8BA827AC9664C5B3A83133920A6811862AD0E703A61F"  
02337 "ED68C5E46DA3EE4040515FF3FAC7D566B5B4569DDC", // PRN 15  
02338 "D2613E9CB156ADAF36198A971E462F0CB406AC07B6C729F0D89B37170790C7D14D671A83B9"  
02339 "8827D1E5C9788B2C92027096AD84D5F4537DDCFE7A9B86005DFDC056B8762C08364E118DA2"  
02340 "42773A4B45BE4EBE61C5CC7656C7B40E35CF4E00987E66029153F13B1A86F0781F7A1DB045"  
02341 "C627222CE04E70516CD77B453B3505EDFEAAB0CF5C9CA4A73E330D7D1A053D005A09B76E7F"  
02342 "045CE132CE3479BBDEBF899ED79C6FF008E5F47CC8905665DE54AB835573728EBDF718B660"  
02343 "6D63DE7D2C7932A93BF302D0364BA4CD284B6E6796B17D36BA910C6FA2833E4B7931B3C7AC"  
02344 "787515A7FFCA747980814AE6CEE416A937052DB39F9AA0F0F445D6986A678D6A2865A50A3E"  
02345 "781D9B667528977AFD897F039DECA2D3EBAC4B9B0E2CDBFBB9A14943EAE860B50665BFC8A7"  
02346 "8F23D380B1CFB9ACF29D22D6FEB121EC2CA3BF988090454D5F60E0FA7EF15FA79E4109D02C"  
02347 "2FD16A1A932D07F986AC32854233A494BEDD50041DC054AA7E9B00ECD8CB929B1F5874E68A"  
02348 "9E6414B99060B051FC50766AA67390050913BC937F1EB359C61246F4C6BAED9F7AFFACB35"  
02349 "907A2C6345022FC7B7EE9F5A4E1357020773B6E5AB442F0263BDA8A7BCAD48F18E4D70444E"  
02350 "0E63BB43E4BDF9BBBCF3A55012D2B5C97F46A78A4EF307113E43FBB4AC9B59DA1C673BD4DD"  
02351 "426438DE9680B019F6FB215BEA45E4A1CE08EA8FD0453D7ADE1872FBED83919D73D35962A5"  
02352 "338FA72CE3660A42A085A691D692BE4D4B04B05E52A72C96C0DE8B2C61282ECC81FFCF38B3"  
02353 "6DCD3D2C85D26A699129CD1E2C32CC05F127872F22D5BB16830C8A35D1E6ADEFECC32955603"  
02354 "1A5FAAD0F7C42B0ACE9697A824A443B4F69084F5DEAF6AB8EC7298BB1783EE798E60DB59"  
02355 "6396B376EE80F0A8C6E3CAF1C184BCC2DF5D991BA43549F1153AB8144B98DD041D9356656B"  
02356 "D73CB2583D0E76DE2BDEDAE587400E208665D16AB91D12F2E54E994656DF11ABFAB1A34671"  
02357 "F6869C44EFD0D3A425C8AC2FF22FA2EA93C2F53F4248A1EB4BBA2672C69F3C2526950DFE581"  
02358 "F3D41390DEA7F85A9DD9DD3BF646976AF30495A5DF3389B345A2D83D3A049CFC68FE8A280"  
02359 "276B8B74CE9F46DBBB8AB3799C4F9A6ED1F74909D4257ACC5790199852288A646DE04E728A"  
02360 "595147DEA7385CF19B5F3D257B5A6B2A8B718EE5573E8B143D63C80E9ABB07A4A228DE973B"  
02361 "1CD4BDD8003EBAC41621B0C58A7290750EB8AC72A70D1DBF0B48B848BB5999063689934C8"  
02362 "65E159A7A06352980D55B6957C90D83F774933074F1E0A25D746155B884B5DB60E87C54F8D"  
02363 "F312852A3A8AEA9A201BD86455F10BFC85C25CFF8DFAC6417611EF75C44FF7306666D339BA"  
02364 "BDE9D9136F19E3528464C1676D15DF8F98BF5C827B0E2C9D08ADADD550B45A7B5D75C38669"  
02365 "F4E69057767AEB2C60D4332353B8036E0A2C8885FB024C836F033944DC97EB15A0D6C4BB49"  
02366 "6D3AD8736DA248D39D1BDBDF156A2471289987866E5F7B2131563991036D06C39A52B1643"  
02367 "1631E50DFFA9429DCC2B6B22A14EC249B471AFA0454E257935336D190B61178A25D39DE494"  
02368 "EF31ABAA10313C5E4F33306BD2E73ACEDCDCA732D3770C6A3DF47C8DE4A41378E1DC500A96"  
02369 "36E133A46A9B1D31193CD6DFC46D04D5E9F79028E96A3FF589A65010BC03D872D075FBEF8"  
02370 "CBE0BD3ECAF51B3B46847C3CF4017B74D0579C7D1B9B3D7F07C76F740ED84009A8511E77AA"  
02371 "A4E879FF26A96109D829453AF0F179E504C5ED498E324570F6D280EC586565D9B899F5C368"  
02372 "0F1F49FC1A6743631390C60B782DFC09BCAFC0549C", // PRN 16  
02373 "EB443CE9D186B88CDFAA0BD64D8832C38F909D07B105322C669700906B564D2FF5563418FD"  
02374 "BD8F251F431C446DEAADE73402A68174718460872DCCFFE53EB0F926C52375F2A52F21922E"  
02375 "4FC39E5ADE2B66F0F3CF1037DB90781D7982DA4CBE85D4F8B006F5F885BB28F58927B05287"  
02376 "B91055AB71368D7BF59DCD9BCE493CA41E4394B5ACA13D474B447613AB0336C21334AB321"  
02377 "02C482592DEE4F58DA10BB460F751264C0A1C99213F48BFA497266675DD215FF9ADADF1125"  
02378 "0D37A99899BE660F369A6EF3158A2F7F34C9A2AB88F4374D94F199A94CD06BFE5026BCE5"  
02379 "D86142AA7D38976227DCE0C617479FD2691339883FFD5277042BEE94B52FC2F0E7355CC5E2"  
02380 "A5387E58011DBBEF3ACAD589BA0F5F9BDB93FECF8B4D281A852562C33CA8435137D96958732"  
02381 "C4449333757D92941FCEA8A09B2B54CECD5B26191D549B4CD853183735D31C2AB921D7724"  
02382 "FAD18A939C1480439FDB7588BCE0FA44F9438252080B723AA39FFD633BF6EF6CFECEFF010A2"  
02383 "5CD2F45C32CB378F0EE54C1AB258BC3A3010644C5478FB2F8E182F4364ED8CB06D1CD33FFC"  
02384 "DC824E00635A16638509B2BE51A539CA1423280AF03D2F9E7117B728A1B337E85BF2EEFEA"  
02385 "36A6B399606AC9C8EA1B1190E8804547CE454145BB70BF06CA72F4375939F97C7F5E65350E"  
02386 "35EAE2B28E939F7DB00E1F53FED4BA864EDA16645B11F8AB5F56A5024930B2644C0AD3B543"  
02387 "9F53F62DEC619D265C0280201327D4CAF95BEB7C9D4DA741A5D28AFDFF6E9647DD7C32DC3F"  
02388 "AE6C102AEDB45678488D867EAE9AD409D62CB3222C03B234C700F007E9B74BABFF40C81CAA"  
02389 "CCCSA411B0A57838F2223BFF9354AE6185EA61454FBEF2EBB127719965F53234C48B959136"  
02390 "E6A9CF864EAD75607D5F4461DAD3E43F0D87A83ABB46C525FA54A946092258DBE86CCFCF54"  
02391 "06F049B53C67F569192D282B3945B3C8F8E42F40862D580DB2DE229D4227687E4901109A6"  
02392 "15827F7897B5234B31314E14EFD35E92A919614312D95C166C2716AEF68403464D5EFD146"  
02393 "EA858E376BC2FBD7BFEE7735B3633065E41FA992DC3D89659A5F6188F5F959681AC8AF7F0"  
02394 "8365188E9429E40F7A782405AABC129928ABB951BADB25B66DEE641BD42FA0B0920207EB4C"  
02395 "CB658B41CF9FF15EEC749D3EA86050A470052761700BAFC7D645A090E5D149C16FB8B1A7F2"  
02396 "4F24B0C9DE58EA782A859F4EC1BF2E86B85F8BCBA734CF7A3206D5C307BF6B4D6811216E89"  
02397 "0079B6E8A980872AD9E865BFCB4E8CFF67C0758FCF3E693193F31A97BDCA220208FADF6E"  
02398 "1F7294AEA7CDD6F01A19845A1816E2500CEB0DC9A0EBE0645D3A12571B29242C874263FB8E"  
02399 "6C6F55DF4A035C46FA37DE67444B427D52FA662D1B6BCF91E1B310CFAD91470652D72E955E"  
02400 "5AD4CB2FE397E0AD9BF5E5874DA7A1FF38A6ED5763020EB20C2D29E618DA414958C88620D4"  
02401 "03562FBB87BB206FBE5DF23836992970FE8AA02E42BA387E42684D234CB2EF9369D8860CC2"  
02402 "129153E29E3C27660AB6DBD9875A7D8B15DF0915A3ED7D1F8D4072E161441DB969664AD325"

02403 "D505A166DDE4A01674CE6E9509FFC6EC801AA9AEAD1A03B4F15A80C315871D8074783A72A5"  
02404 "09F16E18B75C2A1C8BBA2E7470D99916B75BF60BF0DCE946D9FAE5BBAA0F29A619C140F67"  
02405 "859DBADA0B83EE8EDEBA8ABCAD485E1F17E1187370221C5E64955E6B0175292D268344ED"  
02406 "CA3F7818EE1D59B0F14E0A0D03FA2EC1D8AAFC986C5540304A92057E18CE6B1F3DE37A4B3A"  
02407 "8C3477BD6073A198BE62882C84DEBD29D7387DF3D0", // PRN 17  
02408 3FDD115E949ACD46F1FF5A6AF28CC0C85B149CE83F9D6F575BA8C612D3C219C37A4A880541"  
02409 "239366839FC927A4B11BC6401ED48941F86BD00FAE5232BB44092786C63A3B676DE8BCA175"  
02410 "B457BED1C099F83A6FAE755777A4E63F2AC0E369BCF0B428A5251C4B04FA7B3B94E88BC4B6"  
02411 "B91797B0AD88814C72F10B1130F10DB4DAA00CE3ADCB0C5FBB1B3FCEAD129C4E75B3684A9D"  
02412 "7B93D520C88D0261AA6DEE1FF5E67354DF2DC426B7E5106F613CF46D819398A856C9935CB1"  
02413 "4111522A63CAA36430F69B37BCCEFF5A118EE51DCFE6D20C246580D02213F3C3DC75D4D260"  
02414 "9CFF141D0E7210283BFC55C87B3CA52FB44D3F105C96B1AD32C8EA3B87F71A199AA19481DF"  
02415 "4BE31A859E8A9D22DEC274EECB2874131E7A8CACC337343A969FE4A96EBB32007BAC4E80C6"  
02416 "896AAD0D371E4C85B4CAA4B2FDB76716969C1275C6D8CB462E9E96DED97DE86122E9436130"  
02417 "C171ED611BE4EE7B93043DC7262FAABD369F5F77ED35060F8FOA3A20917CC88F05E0A28305"  
02418 "3F680A606154F0EA7ECE38940D5E78E9EC944D2385F44B00D3DD3195FE122F5173F32C49F1"  
02419 "25930B0C9AC644B7415D3745802F8088064AE7804F8ADDC060AB6205CD16CDADD6BF697138"  
02420 "60B378BFF0B7CD3365F82CED1F61D3C3386D83F35B951DDA4DAC06897B92F83CED9D96A533"  
02421 "B81854D609F71FDB2B743A25EC25C4B1BA135A9C3972DEFDD1F72ABC4D41196D48947584C7"  
02422 "FA57B1D017E531F58934365B9C2C269452FFD3B99597239695A1AB154BAE6C152DF283DFD9"  
02423 "0BF69FD27182732CA3AA8364F7872961C527946E06A392E1881BBC9D7875C1770ECADBACF"  
02424 "274EF0D4612436EF0B8688DC6A6B77EB520CCD991EBFFDEC26438D1E4344F781AE0C27CE6D"  
02425 "C466250D99C87961ACC102D9518F67C2E90B6B9B9640AD43C64570E24242DA73F060E8C460"  
02426 "0BF69FD27182732CA3AA8364F7872961C527946E06A392E1881BBC9D7875C1770ECADBACF"  
02427 "1F90A6E3C944B26CA1CCF6509440301C60FE132B4DDD35ECE30ACAD661F85B633C283DC777"  
02428 "CB9AFDBBBF2D95CE29BC5DF684C3CC9FF47E2B55D640B65E502D4C548738A142D4B0831C63"  
02429 "1DD46749E617877415FAD96286A2332B677CC355D47843C00B788B820231CCB78942115589"  
02430 "2B702CCABA429148AAF206BCCE6DD59488C23E30EDAC1AA2D5C046B7DF71AD9335C840D641"  
02431 "8C548029D33515A3B1D9609298F80DE4ED2F2FC534CE95CC90D2143190C35DBEE0E6D83279"  
02432 "581748B7D3BAF9575A6E62951FBB90A866FBA956C306C8E34B4904682DBC7711A1B01568B6"  
02433 "9B55A18A7426F0988466EE147DDB723F3C225E39ADFA3E020D862EF334A26FE139B1D51C69"  
02434 "D56F6C0D8F3A122B718BD49599403BC508BB03B5F424C6723401C4727EBBF0D06227AFA06A"  
02435 "056726C3559998969067278DB89A99788450137728693F5304713E444E40CAEB1640F0EF5"  
02436 "357B3E9D82900B929C8294EB2A880DC03C7E71A8CE761D64FD7C337053B2C6CDF42A4C49F8"  
02437 "BDF1360192D539DB104BFEC4FA55DF6606E8A72F789549E286C189C0B7E003A6CBF7785940"  
02438 "079DA124EC878E06D60A233F5507D8AE1B87C7CC5AD2CBADF5B8074F5A02DABFFF4FD6719D"  
02439 "5FD5312E4724BF8F72E726D8B85DE43C71F4BBC009E2AF01086468C60A5BD48EDE5EB5A1C1"  
02440 "307B19826D3B9D9126D0AFB69E04EBC828CF2BB33DCEEC107A016E1BEEE9898A2E4A6F82E3"  
02441 "5641157130D1F74CBFFB2903FB6F8AABA099C38831106DD169E0DC567744D054743F4FA1FC"  
02442 "EE05ECE00B46A453E4D3E8E39992266266A7338ED4", // PRN 18  
02443 "FCB7CFDD8ECD161BFEA5F40252E37463CEB8B0BC69606B5297306CAEA2D362DF5D3D9DB584"  
02444 "15593CB15FDBBA7763EF6BE72FD34C983F0962341EB998F7304FAE9973E54AFA321EA23531"  
02445 "D48069E9F594754CBB0CAE12B28AE611B646CD1DF5FD9C24064CEE1A2B69FA82845D01CA0B"  
02446 "DFCB0F6630445BD7ADAE0AC8E89BD0FD927B78B26A97AEBAAE00F43EE232210635AD90A318"  
02447 "AC2006DCA0A68B24E4A51FD4343DBAB51B1409EBE1D985D4032BB5AAA110C036ADC11B655D"  
02448 "94F6F9B789CA92F0D9AA174E6FFB40E3538654F6E60A7BCA32338A136BF1D6F7FFE784F536"  
02449 "91FF4623954A043D071D893BC25FB52FC86C82DBBC55C9DBFEA045D004E5F3ADD9FB1BF1CE"  
02450 "B35FE07214737E082CE73CECE4B3B3866B0D12FAF7EA433D43CA7B0058A833F8BB7704C1DF"  
02451 "303829E417B3889D61B66586396E23D4F9273DD7D388877FDD431E07BCA7364D051A3417A9"  
02452 "D6FE3FC88F4F8394C4C7AABCD2C7FEF4C3F428434541A7CD845ADCE722D3155AE58924B5BB"  
02453 "02CB64502BA7DD67F059E1D8891DA6045AD994DAF798954059CB46878CCCC52525F66B37AA"  
02454 "7CBF4790136C0747306D54E83050C45AE5A903A6726B82930448ED4C0E2B4819416E2C01E0"  
02455 "4F9D4905856D6F38FFC0BB58FE0090793E12AB8C5D769D17B928FF258E9D64C3004FC8F6DA"  
02456 "FB4A9D87DCDB779554B88C0C51FB28A15F09FA2B50489FFD73270DBF273BD941C22018FFB"  
02457 "B175021B21F9B43AF09364758B4CF90546AF1C20BBCA34064F674D46169F2665A0A28DD749"  
02458 "9201E88336D3417A8264DC9DA7114C8366B4DADA1262072D9DBB3698CCED710582D4C26C00C"  
02459 "0A868D3B4AC61C5F1FADA4E35368AF34796A85EF56B5945B8414EC3154A788CD3AC3A1E85F"  
02460 "A10ECF1562283E7B1C514B3332972192AD74B423DC8D9010AD22C812FD6730C9B3ECB849C8"  
02461 "468D98E07F093F2306858B8FE7B9459E9D1C151145B36ADED231B6FB4BE02F6A39B2362C0E"  
02462 "9EA250A761752EDE8D91428996FEA4D5B24E3359304A76A838F5459C60F61D56E5F80767"  
02463 "3BE38BF4775FFED272EAF58DA1367EB30AD696238680AB47B446C4E14F077F0A188A91CC05"  
02464 "9BB7A6547C53317C3F7B59A0F61173FB8A1C517194BFE0E38E66EFC33514181D142BB1B99D"  
02465 "5D651FAF51085433D5FCA310665B4535AD827A3313A99B8F2BEB5B1C0F2EF935CD9BB7194B"  
02466 "8E8CDCA268FDCA204ACF4FB55B1FF51A17E28BC22F80E8BB7A49CECF3AC71D1DFA178A4F"  
02467 "B497D03C777A7F24172B33558100336D7846E24FD65B38CC9B46344339DC55E76CA11061F1"  
02468 "FC33C8E48DB917E44175A1516372C3E665AE9647BF5A6A643E0893ECF1606D4824CC612"  
02469 "833F5453ED49B82534AE739A8E5B07F20BB503632BD97FC7FBFC01E60C8DD5876867469FCF"  
02470 "EA0D0B8F4A70EE4A063EFD074E46AC1167A0F694D5614BCD5035EABABAC09532D84743C703"  
02471 "BF01E46AF02A0A2FAF212C049DC295824F2C3C967C1094830BD8D52C8AB435C161154C98E5"  
02472 "59943DC25897C16FAC9011FF9427A4F605758B746C739159466EA410007B8602DDAEF6ADB0"  
02473 "8BA6F78DE7017C6BAC99BC102ABFF75EA0E6D992BB5D7D2CFE0959081810D8A19346DD6D03"  
02475 "61D0E8C5880A6D93B339709A3A57413E2E911B7DAC253DBFA850B9CB14C758B954A94A3560"  
02476 "1991AD0882C9D0D5E450010D3C9D782209DE09E03F167FD2D3A7999F6A0F9EDA2D960DD70"  
02477 "6F95A2C873825C0F972B5B9D59A1A300235BCF4BC0", // PRN 19  
02478 "B83815123603B9E0F1EFD6323908E879487E0BD6380DC9B000307BE0E1C458FDBDA99BE430"  
02479 "EC71AB789F07322DA6E83554419F27D69642FDBB139741B05210A3D1E5F62199C4FD09FBC"  
02480 "31784E79AB305543C7786CF96B631825A3826FBA918579C4EBA81EF8378F1C299A17DC7B00"  
02481 "7F289F279C0B2653E9B2FCA21B4E1E089263B2C6C4D1925374DF289FE66AD492C3F973A833"  
02482 "A941A2DB23C9AEDCCB6B4D2B49A5F8871E03759137FC08B2A3B594B1CE25C6CC7D200369"  
02483 "C303383EFD10B35D9E121FB231863B262F3B418FF4AE7ADDFD040A9D03EFDDED962C071F9C"  
02484 "5E2BDD2A15B66450A23D05E76B1DC1BF9E8E8EB1023553B992022AE7D97222A771DD070820"  
02485 "7733529F7B65FAF1975C7067235AE74C3EBBE551162F1EC737E05FFAED13A9AC8127E5C477"  
02486 "435BF033F95978A9A2AD3A98E8D93BF180B1FB109864213DFCAC68DC9A4BCAF433751452A2"  
02487 "ED7706A7FA75A8966778871044E1B0B01D5D07BDD707115BBF8B596D321064415268D07CA"  
02488 "C56F85CE97D906752E18DCF1E057B3EB7889217EBD34FB0E96A9B84B358FAA654360DCDE8"  
02489 "00693D17DD12E1AE1F44A74999346A60F8FD52996DB2F0079973F09BFFC9D75F7A45AD0282"

02490 "E2AE969CD711C563C5C5257D6F5975E230504C308EDF8A1C094737AFA6FAA8542EC4009BB7"  
02491 "86633312D8F5FD1524FC0B909226241AFC8F0542FF20BC321A0C075AB473B7C3BCE7E9610"  
02492 "5ECA0EAF0648F22F479449636F720E90B7BA442AAE5C2B9ACE8B2D23A5BB1D41A596560662"  
02493 "229663C2D2925DCFE55ED0E8C561C44B9B6E20F96E0487E8579041AEF138CC7D4BD28F75BF"  
02494 "D6FED2A9BF9B39937EB70C5EAE8B363B3CCC823DC131E1D39B22939D4967376CB2BFBE454A"  
02495 "A43DD6DF9526A5C76A822BF3ABEC03A50484F487567CAC163AC594485B49BCC3C553756C51"  
02496 "5944662C39B60A45E40D83F7DB02627AA4299B3CEEE7574C28E8A0841FBE944259BCBADE3B"  
02497 "5AF8B4B8493334F0E1B1276BA88F7069F7E94A2619F02B4EF1989D332C29E43D35346915E8"  
02498 "09C31069D0AB06CA29CFFB6ABAFCC46AFB42B9C01D27D13AF0E9473EF257EF51A4A073310"  
02499 "15624E7BD6004D55C49CDCA19A3F36266E68D1AB2E780E755FEFE738A972BF6664A123AC49"  
02500 "739AB0539C59AABDB220B866261700E7CDC14395B1017BC07777AE97F11568C6D640CB0EA0"  
02501 "B7EC2DAFFA83477227D9F47C066FA889D4A711EBFAF5805D4D554963F36169F61D4823BE93"  
02502 "A7A7BC15F0BC47EE9BC60EFA91C1C5C4193E56D4488BF4716D82FC2BEDC3E88927DCA860F"  
02503 "1EF9F1FF9F10145E308888C0A3218A31F2E108D3E7BCE5A083BCCBE41E0AB1BEBBC8D0F62B"  
02504 "D598AB41A1878CB492CA88DFB3367C4B56B5DDFAFA05081C5EB020DFA5AB4ADE5B23B5649D"  
02505 "DD23670C33F6A4D5865673F47CC34DF446DCACAB896D67A690876AA7456BCF8AB2B1E2754"  
02506 "29E9BFA732F80103685B544B836F0327A1B56928CD97108AA27A28B3E5D183B2A832DB045A"  
02507 "A725D811D10D0ECAC1C3515ADC4F5261794791B533EF51037671578894AAF5B8247E05686"  
02508 "0147F32C214D1A219F0F4D1F93AC3EA7BDA10DEA49AF12D0E512852E959C36F934610080FC"  
02509 "604E8A9A8F98C11A3E427DB7C2E3F887AE53F9624649F4B8AD27CDC81A0153B8E3C3C4B4F6E"  
02510 "333728379CD686E0A48736DAF27055024CED3544507C82DFB19208E9EAB93A32BDDC98D141"  
02511 "E66B2EE5EA1895285BF27B1C78E609B6CD1A39AEA60F72F0580FD6ACEC63CC2E3A26A38C58"  
02512 "3E0FC766CA3FD5DE377D3DF3F543E2DFC9AC399304", // PRN 20  
02513 "48224ABF4769215AC1C60CCD0D9E8C066F853A8CC4D8523A20119B120D5BC81F0D912D4960"  
02514 "1A3A401CC6FA63574A49B56E7257F0C7B2F4506720A1003EA5EC06FA9C645690F05F859CBD"  
02515 "B59928F80AEA94678FF9E3BF2C6924ED21EC2413C4BFAA853380F2D25144CABC10E122664B"  
02516 "2678F206DE41A6DBEE4EF7A040B24833C5C30C6F892135BCEACB2D2F5F7DA30311FA19432C"  
02517 "131AE40991D2EB77515B76F8FCC3F54379A1F040D6CDC597C2CE6C9580D96C47C09BBA89CA"  
02518 "423E4A04D2B3F4BB309DE70DFC1D997C3ACFF77C60E413A502BBC8E6502E30ED755E187FCF4"  
02519 "5CFC05961702953E1991072FBBCEB1F35F3E28AE1AD6798A26F48847A9CEE64025C18D436E"  
02520 "691D64E71563E8F03AE38B1448C51CC7165F891BB839F1548AB092EB78B97FE6EA13BB569"  
02521 "BC706E38F486B346E5E2F5289819DEAB1F4DB223A0B667A46F417C292FA52F80300F7315D3"  
02522 "321993977E930C0C1A1C4A0B80CF97CAF67DD9DC45A0F1419C3795CC4A692C1E44FF4B014"  
02523 "B3B36139C97AA3FF5C8407E698425651E7A51CA3FFFCAD008BA7FE05026D1C3160F70BF6"  
02524 "104F9B9C2B6F5C38C68B83D7C74695923D3C5F8F71CF7C05A9E0587C728E9714FD8C11FFB7"  
02525 "738B52BA12B3E1A760A05AD934B5AF81AC74DA28D045BACD90C394F844756237CEC12B4809"  
02526 "D12B944734F94DB0CF6C171A5686C1EE0E3CADF64907CD884EA65FD4DB785F46EF4DBD6242"  
02527 "789B42155548FA696BF18B7737FBBBDB28B98C79DDF1E7C4379221F339EBECC8552EB1F704"  
02528 "DF16048B1619A4366F71753F8EF143B47161955A3E2413FE8A7C73024E401FCB54B64DE247"  
02529 "82CB067F034F5695E0E5133AE7AF977D555A38DB3716A752C2BB292EC06048C76D64DD5ACE"  
02530 "EFA5170C1AEB043CB317F47A75C8627A163079C8501B4E1D9260286C351299CE34041C056A"  
02531 "DFF93E07F8567E882AD3A9C8D0E7B990DBAF24ADD0E98ECD4708E5AA8F3CE1BF650FBD70E"  
02532 "50B7D9B055E446B2A377830996902ABBBEF4EE72711DF1AEC6D18BB43B0CC8D4D3CA087F0C"  
02533 "924EF25CA14650CEB514EF04E3BAAD9E7F1B7A6D0B821874C121D6EFB050C0BCDEB5995B34"  
02534 "1810B9CF3FA482578068E5E835AF569290825DBE50A8B1A8F62E33757F0FFAB3C6F9C6B4E7"  
02535 "ECDF93C98A529972A72A0F48559C16137F6E1DE08CE862BF76563586BB9DA6ED4F89DDDD5"  
02536 "549CE50C8C79792F4C68937100FF8DCE52BF335077EFF91A9674678496E90AE1DF2FDEBD54"  
02537 "70E3A8A5B6A2A5034364FA37F42B19233DBA4F6EF1C95E4921E7E8AA8454FDC185071587B14"  
02538 "135A1BAD13BF189636840535D75ED1EB7E7014C5044009E9CCA5A469675A24DC821591F8DC"  
02539 "C40B78458AD792305AF2948CA3B262ACB75F78B119277D84CDDE94F790DAE6154461AD592E"  
02540 "3F48BE67B36CA3928C24F02704C9A02122D7ECA027EEFD8F0DFB75BB7CE11FD2240CC698B"  
02541 "96616946DF9F513CF422FB5F811DA1EC68288CACCBBAA85CE410D220B4F95B1D844EF7A83A2"  
02542 "8F31249C4107CD76B77AF9012392BDD43D584C8DB4E10F6B997456B12ABFF5B85FB3188E97"  
02543 "3FA8AC9A9A532CDD93CF0D3F7B9B5E806FCECAE45671A25FA49794CEE3126F35F1F2B0AD05"  
02544 "8948FDF4DCFBEO8EAB3F4C05F1153F482033E762ECB7B4CAF82B487B421BCA7637DB621212"  
02545 "CC13320B2875560ED11747F7C0D205E599172686ACFB51BD21ED437012810217A6BA6548"  
02546 "CBAB1526A438ABC066C1CBC422D8214AB002F21F7063BDBAAA93E7343674204536942A0AE"  
02547 "D09ED4F41E7A458F176A8BF399EE07DFAD47F0F6A0", // PRN 21  
02548 "0FEE25E2B7368DBF42023A84F771CC8BBBE9F4401605A100E3DC1BF165591FC7B19399B80E"  
02549 "A19642D41629131B8BF913716AB4DAF40602FCC7501DC4D9F1F08CB1D27A78A688318380BC"  
02550 "1826D38B61756FECA71D8FF5827DA53F5EA772C79E2794D7DC062CA34FBBE1221BECFDADA4"  
02551 "61373DB6AE81438171B2BE9378600EAA562E189A3CCFD0C8E9B14A682EA2CD96217EFFE7B6"  
02552 "6E7E291A5E3FA2BBE8E97B1EE5E98B42DB9A3C8AC14A511233A715225EAFB0A337BBBD392"  
02553 "E51787FC5CBE15E878B147D489F524DE451E4BBCE81BADEA4EEA74A31CF32871DE17FA4162"  
02554 "8533210849706ACDC500A39A7B33DF16FAD8049979139C1A13D34BA9C5FC38403042F67AA"  
02555 "CC781B434E8F3293A6C71D0C165F2C168FDB2A4C5AB45576A8AE22FD09D2DF593B4D370908"  
02556 "2153B99BAB2210E94416A88AFEEC18E0704B2D82D089039EBBC2BFDD7D837ECC8A5C731D95"  
02557 "1E7C183261AB947851F8250F8A13522567EDEBD580AFE9DE97B9EADA25B12A6EE8B7F49ED"  
02558 "399CC4EE828DC01FC7F5DD39A4B173CB0B97B01C872F1B1FEF6569FD5658C8E7DC2360FA"  
02559 "C741F863262BDF7EAF1D39313161D31364A5E53CF8C803AE763B3B63F6C50FD52E03DC5E4C"  
02560 "AA1E8D33CC9780787214D79632D24D8A04D1660CBE1E9FC06194FD9177F3DF6F02871825A4"  
02561 "B78B2D2C62D5545366FF7E6C656F0E4EF2A48328FE4A97D1E14A1EDDDE72102BE7516A1000"  
02562 "3A5DE6776B57A0BB22EC2F235F16613B14409AFECAD4CB2DD16C4099DD70612A6F12FD17AC"  
02563 "C89E147A5EOCC5EEBFDE24CE84A126D31F4ED7C66151ADB74E83ED6A6A4D6D3CE05FE92D45"  
02564 "C63F3F36ACDF36211E0F9F2F997F28A0FC9BEC96E16BE28760E3CC366EFF2FE4F772D66901"  
02565 "FAAFA022696012C801B8AA0F48217B051711E2D91A9380360BA93E6F40F1E9069772E63B37"  
02566 "B44859ED791542A04B51827003402DE69BDCADB886FBFB167173596F248A0F46166DC0D4"  
02567 "FD3E7431BE49C8DD3940908BCE8D716EB6524B188A8E133E0F6D194920016B3E8146A773C4"  
02568 "94427FA9D5390B143985F312004641D7300215E072D28D16F8FC97E14741536FDA9EC945B0"  
02569 "D028A59C2F209CB0618240A3D68D230A03ECE99665D91D63E96C2B48CCED9232B1FCFA291A"  
02570 "A28F101AF25874A7C3214F22CAE43DE61B9D1EF18451CFBAFF3DE7C864222E3BAE64BA5DEA"  
02571 "C8E54A188E0BDBE485F576F506D2188AB1C839E8DDABCB707AA9C6036EC11E88C273C9AE2E"  
02572 "6B8564B1387C18C61C8396E9A56E684875B7B526BF60BE71C38A727054713347D5CA464DE8"  
02573 "1F9A5B8DFB8878B1E4EBC23BC88061643FF5052572CE5025093614449EB322ABEC46F2D948"  
02574 "517649F7B92155FF4928A8C094C22DEE253FD02413DE44A19C63DAAE25C20F78D50E8AAE"  
02575 "76DDA4DEBB7ABE2F6150D245D219D9AF2E64090ABD4546BFA781570F75C2D1C6E10EECS763"  
02576 "0BFF0EE6E554DA907288EDD0D014E8C79A8E647364BE0BE42842C327B97B20CBD63B3F13F3"



02577 "AF26929C2DAA2876D3913064856D2C382EEC42626F0672298ECDD2EA3E26EDB3BAA3F03D95"  
02578 "DE26E22B1CD7A6D4ECB20D19A1311536329B6B5DBE8753C6D877B8BBAD5EFC386C083CC45A"  
02579 "E43CC09E6559613079911AB5404464BC3024D5004646708F145BD5A714D8DEAD1EB3A91258"  
02580 "BB5C79D99EE9917FFA7669F5B72E11AB4CA8C63CA2D22C2C7B54929B0A43908148FA5AB17E"  
02581 "FE7047ADC92FBF78C6D66EA4FBFCA62EEB82A2863E428595AC4D7CA566B5DB79F210A18D79D"  
02582 "BAF1A988191D2424624FA89E48A91AAD26D2A4998C", // PRN 22  
02583 "38D33B3C841911BACCBABF31696C273DA36C219EE91A8F634B3EBC1A4AF8F09ED9AED1BBAE8"  
02584 "EDC31CEC9E949D083073CB13F1D1F7ED660BB22EB967920A69736E5D31CB21BB504456A686"  
02585 "3CB625C518ECD1A435C12C5D7B2164EA34421B47CF476F9753E912CB401F8F6183F6A69ABF5"  
02586 "D5CE152167419F092B77B9CDCB0E42C118C75D7D3605024058424CC8ACF77BEC2D0AE951C"  
02587 "E39BD13DCE61069BE7950FDC0E3062BCEFF8FF828662E29F7D34AF1D2BCB356459CA5F76E23"  
02588 "EEB7646C1D125A95FCF55B22E306F110B01E5376CCB5EBD6A770AB7FBD770844AE1AEA269"  
02589 "AE3640ACB862F0DD15781992A13394EE22318A4309E1D7B87D8CBCBAA24F1D61F365935CCC"  
02590 "F82FEE82C7FBEB820B80A504EA01516D4AA7975923A6F177BF63C137B36910722124FC8AE2"  
02591 "B89C6D00A2A2EC8929B3880622455A94E01DCF8EBA376304D9AE1DBC4A5EE91D80F455016C"  
02592 "F0B83D31CB75AF547CA4391C7366DC769C2B684AF3EB3520130FB48958EEA93CBAB12FA8E8"  
02593 "91EFA7375504C7302B04EE822775066BEE720176DBCCC3BD5DCE8A1F2670B4345EEA4C40BF"  
02594 "CC7A715A495EC755ADBE861C9DF7F55D2070C4C4C0503278AE04990A8734FDC630F3737FE"  
02595 "DBD929D252555A3572B66FFA254555573E3C370A7302D41A91245C4C7CCC4B5081F6D8435E"  
02596 "F5F7FB56E51B2AB58FD0579FC4C66AE0C8B9D779C1554EA375D725C03983F2B4A16A419103"  
02597 "58FD05A8F205DC1179D62ABD7A8738DE8F4ED8197607623ADADC2F51575806E6F83C76DFC1"  
02598 "ABED3DD4C6A9FDA817D92086FD6D1B03D4F4EFA997CB5B3572757FDAC41459BA087805525C"  
02599 "2DD0803A18F887670BB898028F9B1657696AF9CEEB7E7498FC622059FB4C0BDFD377E2B2D0"  
02600 "D11F37A998DFB482667BB9167A950E096AE1C15442DBA4C33A251D4762CE06DAB214789282"  
02601 "079421B2EBB01F8EC29D5EF3268D8147D9DE7AAA6A117F8E9E084526C3394AB0E79E11DF79"  
02602 "E83B472874BEC646853643EB0E140A4C81FBB582E04E20209FAFE157AA745B28B3018BEE1"  
02603 "0D5DB657196FB4215F677B1A78AFAF0D43B206E5F79D92B6A064EF738150DD4F2FEC7C942"  
02604 "E5ECFF7830CFB8D67C8B289095BCC384CECEB2927099DD02D1358B20BA5DDB20919F2BB8D8"  
02605 "2DBD308160B221E977F13928F1EEA454EE387C4C3BD69D6CAB26AF98F6820C3A51667ACB15"  
02606 "D68FD9F0A1A18898AC0E917007BE36CF6C2C4D6807110C9EEC784F0B955D782FB903433C3B"  
02607 "BFAB9B1EC4B1493892E44AF2D32E240DA7D7F61F19C216918CD6EE85DF8F08D626D19D31FF"  
02608 "F4A33B7CF61A060A34C8B720FB3355D3433BB5D612F68A25BCC03B44338F7905074640ED7C"  
02609 "8D65799B90A693C71E1E67B1A2643EF1D36A280563AF37CD924BA3B16DA2C264172BD25049"  
02610 "883F6EE7A06815AA14244D9707ACCF0921F0B2EB0AD901006D66D5CC02302A1732FD004B53"  
02611 "32A9A919F71814A4E32E892B952985BC23D364ADFD6FD793F3E78F0680D206548F087BE008"  
02612 "FD55BCF0AED3AE3C4C1158EA765FA9D9CBCD3E38316B85B439F4EF223FE87B9A70D09C6075"  
02613 "89B00A70D11574DFC0757761EE2FB8A09775F2080036D442D1DE1A463231998E1FC11B53C6"  
02614 "5BC071EE261CE62F1A919691AA7D87EDE03634F229793DEB5D2ADC167C51EF8447EC4004A3"  
02615 "8DC49DE0BB6FD6D976420F949FFBE908BE303E826C3BA297D47E0096C88E6000FB26697"  
02616 "D69BAFD0DE4726C5B744B5655C14FA21328C17A6CEBF911C381E5231AB5FA614387A9DC1BF"  
02617 "D7A34E48EB09F8CF115816D808613DB91AB0D8B7B4", // PRN 23  
02618 "C135B97C53366AF456E70332D981E740CFCA12A17F0037635EE03C7E33924D30FD3946E0CD"  
02619 "639296056886A4A950C0F7E2A17A850A5EBADD6593927559B7E2F9AD23D290AFF8CB41F1A"  
02620 "AA026B0FB9C12B7433B1200109003E732D23E00495F4D2CDEF36B8C9B3EA186E67EC58D7D3"  
02621 "AA1E295C1300D40D216C46713949ADFA210932E701168018451999C00A521117BEE87993C"  
02622 "2AB4008D145FA3B32D65C35CE6F8BDEBA87A4580AF3FE899EC8435464D25AE9209ACA14DBE"  
02623 "3AA78B1D042F6AD749F4B36E60873421B85FBD08D21FBF689557F2F68B1C32071DFB62139D"  
02624 "8FB59B929EAE7AD746B1C646461BA9D9A29F57154955D1118A0E03ECED7F2C7178FF43045A"  
02625 "450B03793F768ACDDF5A6FD2D9BA02D6D8DBBE5BA78A937DDBA95798419BBF88567DA242F"  
02626 "5184128223E47D5CE8C2BFA794299BDC057940AC16F43A6AA451BA06AB6AD587BFA1FB3CB3"  
02627 "BF59BC22580570FCD4756DCBF62DE781F5D6A37D51745C1EEE4BC7D40BFB73BD254B364B1"  
02628 "AAED78A9C563B1D5D136F7CB70A61D48D11E7D8E3DCD23B7F22A8D3A76B8EF01DF43D0ADFE"  
02629 "46FEA695B16F649EE783BE180493DF05FB78B4BE57121455C30301EA4C0AA76F9A48C0FD59"  
02630 "79F269408E56E60190B58173FCA32F5492848DC219A3BCEFF81B0CEF356A2C6AA71BE3AE5CA"  
02631 "0BE534380C0A1398CB89528DC5BD0AE60267877F8FE9D8F91A2E3B0102C31C491F1D95B078"  
02632 "C0B564E163856240775D1A936263D0F6E706788E81D760577078943482021D7877536F480F"  
02633 "A92A69300F992477BF1FB5702A2ECD84EDCF444BD2E30336760C09BC5677F90A764410C9B"  
02634 "240D455629B535F187089CA5FB6FAB7C00A6B83F9D0FAC34728E1AB368511A7D949A956EB7"  
02635 "86E2B00C89C78177A876FA75272026BFB908AD4497C8548A7375E0D61E6DCA6265F9C72EC"  
02636 "43B9433144B9A88F73A2DC5807A5687AFA8601D384349538A79920E7A6C0C421B8DD9F02B"  
02637 "F83CF91215CB2AB27A2C49095F47B870F30BCDAA5675EC717AAF7130147FC062FAF83868C6"  
02638 "B39F9459625ACDCBB81DFB161D40ADD3CA4FED01B1295B0710CC35DD49909856422CBF400F"  
02639 "ED086B71E54CF882F32DE69590F3FBF3F220EA567086D56DA3CDBBFEA7EEC10B6B7FEF0C2B5"  
02640 "3D6B935D64CC0ED7CA1387767B522D06EEB39ABDDE8BC08B3EAB9B7D3777595DBB2BF0EB6B"  
02641 "ED5D4D60E7CFDC399646BA01018B69FF211D7B39378AE25504FD1B189B218F4D8F5A358B00"  
02642 "B65746C362F4AAC38816029A8FBD4E8792ACCA0ABBF4C4C41FCF3ADAF0DDCEF973D76C68A"  
02643 "8CB95AF159956ED1B8DA0BEB9850F025101AA521A99B1953C29CA073A3ADFC98C3DAF757E1"  
02644 "98E42844607D8D4393C0D215E7872E9380637D45717CF349BC4DBE6B97BFA87C55B1D3124"  
02645 "AEBF3FBB9663DE262FB93B3F84CA412CCC88EABBE4589ECA72A76D2C314D0B2959C043AC6C"  
02646 "41DD3E8D0EC8C016B4338781BDB60A509A992A3073842F62BC4492934053B70216E7B74936"  
02647 "A87821F261C16690BEDCDFCB0581C00D4048E60FD52E9187DFBF7867DBDECA23E407D032E9"  
02648 "AAE409F2F52992C12810AA7DD9D4F5DBD66B89415861F8D892AA09B00DBD6C79C04741401E"  
02649 "F8CAAA6DB0979430615B6E1B06788D26EFF874041EB3E47940B8DE8C11B97A5C0C29B8DB70"  
02650 "A8F77F5937F6DCCE3F07CF785957CD1AC4A8F82C7C37DC4DC76FF4AF5DB651083C207B1C39"  
02651 "4678BBF3B4547A047BC317B65FFDB1E30848183DF1699FFC67CE960FA4892C96A1F678BA38"  
02652 "9A7D7521B69B4728043C220283378C184B3934418", // PRN 24  
02653 "71DE1330277DE3EBE33872AF8677F9D48BA4C57647353AEE28349EA576576330D3A5C0CEB9"  
02654 "2A9FBE09CB8F56F87F53765BB1CF0F04E3DC01FD450FB9832C3DA6ACFA0AFA72B1806F6B4B"  
02655 "6D58C9AEACDAE0847C919D49491EC69AA8F453D7699CF944AA787038782BC3A78628611A1E"  
02656 "FC22BCE7711795CA62959CF9E89C12E64EC57485EDFB683010B8C010797614AEC2AC8F28D7"  
02657 "0358A94B0209A9706487E6684D6AEDCCFE7744D291A4D08DF98B49ABE9CCD8209D080F075"  
02658 "DA64763BC847C53CAAE65ADA23DDBB51A9A701F225954612BFA5D7BE893A9C0888E158DBC"  
02659 "B660C954BBF780470A2C7BE86C0E398BB260591A0757CD592DB7F2DD9BBF535BA90F82BF8"  
02660 "505B4F40CCAB0E75B82F78D10A68D75BF49A343713EC025A32ED56EEFFED53ACCC92B1531B"  
02661 "F9F0B34028B4989B5B6D627274401DEABB44E3C226BEC94729DF2DDFE7EE96595217B6E54A"  
02662 "CD596262AF96201DBF99BCA8E9679906440467B499EEF821E958C1EED0FC87F00B7508097"  
02663 "970C27FAA1803E9CDE0B727FEAA75838093153BF87B8F915FFB0B5ADC3598E426545AF8581"

02664 "400C26584F9B9D32128C22E5C94181561C3BE6770B662EFD617D3E58A9239D19A1E7233867"  
02665 "F5843E428893BC662F87258AA8155B5FAECFAF71D295A06A4EC969A178B5A675E0AA6A2A53"  
02666 "25B823A8D6CCFC9B6B818FD48ED04EE92941F042E7EAC89C5C30EEC06BAF16C7E4127D975"  
02667 "10FABC1F5052BD3E6E3555F1EE48BEF8BD2BB0F36EFC827DC06CBF18BD3B1EA0A878090079"  
02668 "E1200087ADCE4558A8FCCA3CBE14B8E7CB99CAEA5D5D0797DD050D22F0467D6F7203ED323"  
02669 "6EC0780542D280D1382D761FB8E3FBF1A8EBCB38AF1127787D2B35B834124116BC2AEFFC8"  
02670 "C5F24833D3ED8B119BE314681058CB197A11B8220D13D6A8267E4BEF19EBF3689EEAD02FF4"  
02671 "57543CAB0F8B973B0AA280BBEAB79397BF58196B778B3B389622C4A60149F9F03773714D5D"  
02672 "A8FCE40BF1A0A207B2139741937DAAA0958DAE6B4BEFA8C77285F0B0D60F73222A1558FAE2"  
02673 "F358377AE744A98A8F382FBC80290D3FDE39F832D4C211C26BB33B78E538AEC6E309FF040C"  
02674 "130DEA5C1B67E5292372B23368A00C3CF822320AFB3D1D84A278CE717AC362197A88D7CFFB"  
02675 "CB8A86794F64532895EA1DD5864EE2BED85F1A25552F000D4DE6DE2CF7E9E064503C027C48"  
02676 "0B6AFB98C819AEF8745D9EFC762B2F15177A1D5059737F34E23D6AAC9A4A63138E004D3CED"  
02677 "605622AB0B4C8CACE0D26FA5D96ECD32027C9C3255DD3FE64FB9D60AA01AC7D3A03E78C65C"  
02678 "5344E3449668AB45CAEC2EBC9210191AB5B1CF0CE584302401F398FCDC38339D453512A042"  
02679 "3F03F07C6602759BC71894BD375B3E20CE8C4D0EA065753526F7785E56CCB01F87284F7C1A"  
02680 "1CD9B65C60C7387AF6BFC83311F541FDD16E8FB02792DC32C61BB6C30A237952C9C3DE803F"  
02681 "D53B636CE67EDC603A80225ABEF84A46A07176B5ECFF2501739F19C4840E447EDD969825A"  
02682 "4FEA6EDD58BE9FF25E5A54C53B47C115099C95C9768EB4443B6106FD134C283086D6C250BB"  
02683 "185BDBDF0344A30F3A5BA65294CC9D17143A22B01F84C928903FE06F21193FD336411F70D0"  
02684 "69217BC262D645E0B75BF287C9BA8916CB7A4BC0101F913015924A50291DC0125A9BC866"  
02685 "9A167575732C792C7198DE3C8FB1E26648296862544F18B59B1C0C1C2376628D5E6587E0FC"  
02686 "51AA50567C1785F17C19C61E4FB7BA8DFE514E624D0044F3D5AFDDF38E93C6FDDF9C4D3C53"  
02687 "1D4C8AA7F9391746B1EB15B3B90386D486ECA0898", // PRN 25  
02688 "7E8CFB74E2DD4318FF211564D342AEA2EAF031332F27616E5FB202E50B86580C1CE5B559AE"  
02689 "7392EA245D2D446A28B63B8A0E910618B70E9E3F1CE264A10B180E27DAA8BA5DD80B63081"  
02690 "6D971E7E827971FA7A4B6F55496A908602CDB20DFDFB5CB74BDB251E91E2E00BB1B2D6C130A"  
02691 "EEDC46E73CA2C15D0E2F37BA9B4DFF057C3CB4DD0767F1160B833008ABC8CE46B879FC6DD4"  
02692 "7EFD75E663B0B8C78CA7C80C883F0E4475AFA5E9290956A44FEA050CBE08231C9E75ED9C9FD"  
02693 "E44A3D50DD618BB97B43978179546FF522BEB0D9561E431B91B6E7F215FC2762D98CFA3E2"  
02694 "CB6778717ACDDA4C19E2A2A5C9526AE4272BCD140C25093EA43FB99937DD21EB4068F6E53B"  
02695 "06B105061709804F5D385279D4375522F1707637E42B4E748F73C90DBEB11986BAD53F0D15"  
02696 "27BFBDD61E6DAF12ABFC0B962CE4E24EEA180141A62FF062186E27130B7EF1AFE18F3D5B2D3"  
02697 "60AA4A872EA197952DAD78C1A59318349552DE767DB1FD532769F1DEA34CF68D6E7B9B71D5C"  
02698 "8F8C241BAC6E619AC5C8C41CAFD804209E7CF1A553C3799A3A57621194925F15F46FEF94FD"  
02699 "9E3B8DE21D5071BCADC7592AE3B207580C5C39C94A8F3205140F46BBF3A3A9613678BB05714"  
02700 "3FA22D3F51EEACCCADE6EA1444539071783ECE4E492E14A2F2B7FCEDDDDEC90F5EB6D1F5F4"  
02701 "671A6C038D75508833B608A7CFDB3973EC7AB007624F197828D87F6E51239D000AFFEDBD57"  
02702 "E0E0DD1AD3E9D76AD08C751A15DF02D6E4A78124344C770A7090EDBFCF069012321FB146DFA"  
02703 "1B069798B2EF74B7A206511BA29B05979094387EAF7FE56B8041DD0181D177BA26E4837C28"  
02704 "64DDA836BB917536242B19BB9F2E9C59D40C25563DE3B84267BD7F58229461029A3F39C681"  
02705 "1643954CCDD71E72295B045E3A8BF6E0EF367287BEB38F4A7F6BD5B6A90FDD1353D0415F4F"  
02706 "51AB8A0989F75E49E21E066DD85097E350B9D01F7FB7FBF0AE119AABAE7F0F5EDC196E92"  
02707 "C1D7EEEE62E8CC85CCD466DD2D865A6BD7F6BA0A109FDA161CC61FC9891E26A28B4C39032A"  
02708 "ED9CEB21D5071BCADC7592AE3B207580C5C39C94A8F3205140F46BBF3A3A9613678BB05714"  
02709 "D13F687FE59A277F4A982968556789C5D777E01699B970C6C9E0F7E18E99026453F090977A"  
02710 "38ED7CB8936FA930122BE704D61DD322772FF39E9242360144716C0F8D76E46E189311BC21"  
02711 "751F3E32AAF16E5F1FA14CA5E7D7B0F9D211E06375BFFB361F6D8D2D9D93BF154B50E5C2"  
02712 "CDC745E4EA73E2AE03AA086852B6373FC8EDA53854BBCD27CA7DEE6723F7792F4327ED5D9"  
02713 "C942642F92F98943F7F53294613210A39B149BE7A8835087B15A03D3866E1846BD403FC503"  
02714 "8B86F95664226A6BEB11FB4C6FB660A234297ACC3FDB0D6530359C03888BF8122C84319577"  
02715 "044939E5A6AD4FCDFE16E96E531ABB021DBB9058C1EF1D85A21F729E2C274AC8A3E95084CA"  
02716 "1CF4BE892DDEF2F7BE1FC35B18798BB4B320FCD65CF8A3E32C0ABAEB84ABEB71EE54C98905"  
02717 "2D1A640E9D04714255791788FA74C7D791E0F645DE40679E89E0C40E00D17842AFFBFA2803"  
02718 "C1BA63C7734215F0D796082D7AE03E86A5109AE218CD3405CB8E8A0FD85281C84F85648C7E"  
02719 "110D4C6BD47B5EDFA5B47C2BE9F1D760487B47F51B67B6A7528FF31E9741D1E8B8B47B85DC"  
02720 "CED4ABBEF1458CBF45D7E189A8304AE7FB10CCBAC49905AA83452027161A728B8BAFE3B8F"  
02721 "C276D04A957519564ADC181B36D93F31227C6ED08177E37724450499D07A53A61484357CC9"  
02722 "876353510F62707AD967C890C45B2F697C36BB76A8", // PRN 26  
02723 "B536C31B1447B59AC3D787D7D1B9A2402D7DD4C36CC8ED1530625544D93EF0B31F3516759"  
02724 "E663FF7CDF38D275B437D7E59B22223930BAAE4CDC9D43680E9B18190CF794EC4B202458D"  
02725 "10673DF57FDF3B52953AD67CB496E7BC09BDBE766A06170CE220A4B43FE828FDCF0BD59093"  
02726 "E199E0E0EE354528863001C537503FBADDB4C8C2AA72F4251249C7063074B895D8B96516F6"  
02727 "495405A6C7E1FAFB8A302F50D29920C87FD91A240C90496DB12653B38B6EC194960DD172DA"  
02728 "007308DEF074C86054EE06E68C118447A4BE5816754206B5D04815B4742CB1E1EDBFE436C"  
02729 "064523756F8C1A2D21BB02CDBFE223F607481C85CEAA3B1E3FAD53CC7969F19D4A53FC7C89"  
02730 "F31467BBA88ED29909231A29469777230E72B6A11B3CE0A1675C635EC298E2613FD4532350"  
02731 "FA5BC9E13C6036C42911306A35D24249DAF09E88772B28DE291EBC6149617D220385956796"  
02732 "B2C4EB4B4CBB1D1B2958B10EFD2CFB8D8A6363C1BCCF4EC5287928F948F1F4BA3FC8A0429EB"  
02733 "F34294E8CC56BB9A703EEB1C02E45F8E757C4391671F6AAA567EE5D7AC58D3F8C9C4C788D2"  
02734 "AF0958F4D1FBFF1305B98254E43BC88DEBCAFACAB36CAC4FA92F20BBA296DC94D08D49F8E7"  
02735 "303696DEE672EB73670168D733A16BA0ABCD2252791742050DA5B023E97F6F6BA5F3EC988F"  
02736 "DE476CDD14A48CFFE8134486F6721FECB549F4F4991D6A56445B06E81BBC1D68840CDF38B6"  
02737 "EB2CAD9760F08130BFFE6C695C3EBB47800B7E1EA490E64F52C3192EFB45C65959B40AC06C"  
02738 "BEE80ECD0C5EF333ABAFF3E63DF46021E35D1FE93341AA0974E298869EB27736020EBA9F79"  
02739 "B4CF49C4D4AE38526D5A100AF7A7AD708D53CC40C6D520A56E98F94F84185564FB17EC3E22"  
02740 "E4DB6D4D873090E47B72D12539A4EE87A7C39A50579B64A2027CBD2F147E4FC89277A90340"  
02741 "88A60129790A4AC0764D28F91D8718F52C7B1817D5962D58CA130D76519968B1189CD14986"  
02742 "FDB592C05B14953AACDDA46E0F7824095C1FCFD8F4BE25EC9F2B771D813B2315F040922CF4"  
02743 "D97349F8E5A2E5DFD100A7010E3AB6B5065B6357F9DB71CB255AD202B3EB364501E1CB555"  
02744 "0ED59F893D5DC66DACA85D8BA60FF733B200026110A1070264C3A4201761808E7A9F998CEC"  
02745 "F1ABE00ED4935E7D1BD7DD4155AC6B87BDF01A94B1944EEB20AB318377050C513E98A611F0B"  
02746 "23895C9BAC3231D8C7A0EA2D20B26963B8F689B08D4B21269FCD8E2AF83FE6AF0AF2002"  
02747 "86FD3CE1D6FD870C03422E09961349761E8AF6C18013CAE65773A2A4E6262E497FE2D9C204"  
02748 "A6D0151075638040AB48DFAE0801D032A61AC98C4C9B7DB80BF9068C3241EF2CCD352CB09C"  
02749 "DAF391C05D6441CC32D91DC9ED7A93677684B150DD1EB0499BE2014CF2CDA7D264E2F5C7D5"  
02750 "37C3DF97E32D92B52224C1B3ED952C58F1E2531965E64C642417DC7D6AB91B3E6BA2E95DA8"

02751 "04314E10D29E03026F1AEF4ABDFC7D1FD14B9CACB79B3805A1CE5A7D83FD1E33C7FD99337E1"  
02752 "D5B5E548DB9B37813D83832BD1B7F5D612C38B18C196913DADA7C1B6FFFEA6A1169A0197657"  
02753 "E4E2EDB02301A834B362C927AE79FB4AB178E2639C8ADFDB1C7D8F710CFE7EC94286E8122"  
02754 "7A3F8B6DCB7432273F9421EAA6D6A833D4DBB32B19DCC3FE505FE9B9494B2CB18474BFC7F2"  
02755 "A59504B515D63B50A7E259F2F1570B4961DAC9CB30E122A8CF4D3303FA04F610657850111E"  
02756 "CEEC6B7DFD14E75C4FC65F4C331DE1B7FBCF6BF314E5DFFD974861202784CD49054C71434"  
02757 "D1DBA42C3154743562B029FE03D6906A4C26B34CA4", // PRN 27  
02758 "B8E68CC3EA746245BDCEE9B1F97B14A0C8DF0844B23A53379B6317F61ACD608E92C7A0111"  
02759 "902BF515689A85149B20349AD7B6EDA5B539CF8C27D12FD6C7F30111D9251A5682E2463A52"  
02760 "A82DF0C82CB45CEA6585F3960ED56BEC8415F3056D340987AE2F019B370338C3A3F48DAB12"  
02761 "9EDA018D16F6F22E003942B2A674757B7549F05C369B76F2C88E13D20CB5BA1601D05AAB"  
02762 "5140E895DAAC8C419CDB7FDACF3ED0D6FC9EA5D7ADF811FDA3A08E52315E46124212FDA67"  
02763 "7623C8C3E999270EB086DFB0200D2256AA2D63E2A649BA2FEB0CABDAC780BBAD5E0B1323FE"  
02764 "2205E4D64C090449608C2719157F4A56EB5C6A90FDCFD054FAD62CC2DCC559ABDAF2D93CCD"  
02765 "F00EA623A8570F984A2F3457DE06F24D4C178B47947B57B698E5DD62B4D45CFC22DC5A41B9D"  
02766 "B4AD28F4184B9E99D64EC9F0962F5E8662C67208EFA08C1EAF6DF12418A1E39B3A6E829BE1"  
02767 "9122DCFD849407C054690BCB2FB8189E4C005B7DA59A15CFE6744FE53B1E204FA28B5E515C"  
02768 "AE94958CA43F03BC10E6B8138E95CA97AF0715C15F4108D9FC70A0F20B57C4E3F5F3C486D2"  
02769 "79D6A54D643402D691CBB471B3318864D46F57C11B86DBD3E5316482A6125C8FDD52706D10"  
02770 "93917106DE1299DE28FF4C4EAF848EC623BA828A3E030BF0C1883805190D8213DB19FCAC2"  
02771 "E0F50A54F3522819B4CA4275FAE720EC6454129192DEA014BCEFB3A33F487731AFF9CDDA59A"  
02772 "B8B84B2FDB302E1D6F49F7ECF828E2C974FBE023FFBDAD926068644DF44DD50A1C4F341EA7"  
02773 "D20F9C8223678CCA94F75D633CAA620FB14CB9AF5E20A2CB70646F8047AE5D4116C6B8D1B"  
02774 "E67DF616F9585563A3485B06D8EAB51841919DEB3792C5949E9A6CA65B34F13751E1EA6012"  
02775 "2A4A869C28E2C133AB88FFF40ADE0DA21040D580EF84D6BF683D5CD53D6D676ECF5E290763"  
02776 "9E94E023554E9065B9CCCCD0A451FEDA50F799879BBE559C324F7E8EE2E7A8829DFE94B57A"  
02777 "38A2402BFEDCD6C559CA7EBFA7686E0232E9D68EAB024C37909916567D1139FF9B1EF81972"  
02778 "B2F478076300B57CE5CBFA72752915F5AA97BE0E3D58CE1ECAB7638D71CEFA25E822FA65A8"  
02779 "E21F704DD24FF68225F21B2C253C0636652DC73278C95849C6810948AA2DE221CBC803A3"  
02780 "42EF50A54F3522819B4CA4275FAE720EC6454129192DEA014BCEFB3A33F487731AFF9CDDA59A"  
02781 "02EA07F8A5BC8E74A8F4DC88E67DCD197DC47DCA3B1FDD5D5E0FD7D90E2450F77EE89505F"  
02782 "8D3934B8F2F737E06B1586F1133B7BC5A99D578FAFA915C2B50C6D90F4209732A5960BA585"  
02783 "84AA09B4A2ABB01AD43D99A4B7698FDC72FA23AAA7A5DF52ADAD57400650E5CBD003E29297B"  
02784 "5076CA5764A1CAB66B8EB2F3738E1FD8F6147CD8785AD75C57068A4D483C1FB6355988D8FB"  
02785 "821B6C1160658016E128D0EE2B1810FB1C72C46AD7227DEFC52422FC8952B9343381A25645"  
02786 "52F598D02748C435A201668B91F2D0F55518CE19CA52C94DC0C813030E7BCF6EEB8C185"  
02787 "D6B8127703849A6C7BCB0D20A04983F2B54F3CDE7D3EDB8CB14961318462A3F8EBB5B17915"  
02788 "80FEE399AB4344743979B825F76A578E2A0D43223725B983E66D4E9C93ED3AC2FAAF0CFDC"  
02789 "DD06CA8477174CD8D9181DFE96ABDC966A232A39D80157C94EA009E2694E8DAF4820C6FED"  
02790 "6BA9D33F14EF6A35432CAB30F241C1EB10614C9765C33B9DB923E3C00610FC24EBD65B98A3"  
02791 "C35F0D77BA3FC3255AA0458740F21C8A2CC6CA17C51F728C79B21D4C2F1BBAE25289480F4A"  
02792 "D7BB843C454FC4268B7CA790C39C82DECE43BB9C", // PRN 28  
02793 "1E7272D1251A8B5484E3ABC21A85B0316C8B3A5A1B25EB8FEFEC5E867F6DED4807D1C476C5"  
02794 "B1506D50D9D3149425713A11E798F5415D11A7C48746D806C59EABB56240CF0C997CC9554B"  
02795 "3FE3F20F785095DABC5A5B509B23BCAF52BE692D76D92362C524148E884239B085DE22B756"  
02796 "C01DF9C514035E6F5295189B2EC033A3C65E203C1DC7ED072A6400FFE8E626A707A397B7DC"  
02797 "8D27BF24DA0FD9DC2C3879F0EF90B0B8477F3B2C315F8AB8FBB195F7605FE749FE5DBB9183"  
02798 "964D8CEE8B1A885707A0EA17533B84A17DED1594BCFEE87DDDD1BFAB10A802F818F9EF3036"  
02799 "0D57F4D1F233653C8C78D8031F9D1B05596018608B3DA825F4A55F9753E037735F22A8F4ED"  
02800 "26CDDCF8A77E82B1DB19A207606341EE965BAA0D8ADA95390C71ACE1366FBE47B84714C9E5"  
02801 "3DE020108C424B1A961A4656582A5BC95AB14EE6B7648C01A70283DC287FFE282045787B25"  
02802 "2B4FCC2B174803BE7B57B62991E69222C5525BF6436BF092BB93DA680FFBE1BAF7ECB41CD8"  
02803 "6EEAA5BA76AF45D24447825BA5E4CCA2F03758F06910384212BB48775F44CA9F0291F2DFA4"  
02804 "C5EADCB1B9E9A49372F3F1E1499EAF3B1248A893C2193C9AEBFECA9A8BB3A2C0D096BC181FB"  
02805 "64EA091CBF9F36514024C05C13032B8FC7CDFB09B3178BD29F6EFDCCAA74803B82C5B5675"  
02806 "5A681FD5A28DBECFFFD965579077DFC0284CF9FD8D0F94AC4416B3BC8CD0A7C3C77F0ED555"  
02807 "D9D56DAF8A6C181644C599718E8061AFFADE0D5BA7ED4313FFA27B8C4CADE6773D715F23EA"  
02808 "3530EFF6B4F3751A1DFD22703D608EBF74F872D0BB7F84C74FF7095B19EF2555A03AF262D7"  
02809 "4F43A33B645A9A716BE4A9CB5FCBC6C62845166EEFA521D18AA98AEDCC711501E0587EB75E"  
02810 "78D6A5C82B60E50F4D961791D7C23A595D3B949E94CD8EE844A71FA12E9B58E23AA9F68139"  
02811 "8D4C43298ECD06EECBDB371A5CDB52DF5A3C9649DB48625EBDD631CE07DC56A0826DF7082"  
02812 "E77165181F555A5C03F0F8815F79A8AE16FBAC166D13D8268F6EF844CE5E6D2542F0C436BB"  
02813 "C85AE8E47723DF6FB90A36F5D7FA16CE15584CA32C3F1C8C4573C5DB5F0F5F362A063E9AD"  
02814 "99983D93E92646E2694F8A37F1C42664232A42706C7F3BA9CF7F5691B1FD38B0D03364C55C"  
02815 "69A1C6EB937EF1C77E9F35C5FA898BD74E7A0ED3F8E68F85D724C93DD876D9EA074E0DADC"  
02816 "F578364F6210897D530049BEE77504F13DDA44D5617CFE96E8DC4746DA10A8DEE3077991D96"  
02817 "F81393E29ACC32FDEE595F8BC18A251A6A343CA5A20D3D8BE51D6D6F7540DC4D2F51615192"  
02818 "5183B7A05F58729A622ABB4A4553CFB12F99D4E1CF615B2F14E627BDD4B15B7EA2956FE85"  
02819 "36DE3510574FC70D4BA8901A43AF830BA7400CFFD614AD99A81B42B4C5D1A1BDC0C538A07D"  
02820 "4B2A31F971DF0D8F38B28B504C40694B001FC4302C35FDB95B09F589090BA32E5D883A6051"  
02821 "911684B5613E5AE0360F71130D1853641C710DDC0C9DC48D034826B4F115A2B511CAA1AD08"  
02822 "5FEDD0E739CB7306A1405E276A1473297B6CE33B0EFFBA37483489D87D94ADE0F71EE2869C"  
02823 "6699F9D17871DAE5DD21265EF1F6CA340A4EEFFDF1D44B04EAF59B45509CAA51869367F99D"  
02824 "A1C5EFCBB74503B414D3712D79D490E65059FC78011EF17C73A1147B1F22C62ADF4978420B"  
02825 "8AC0D6A977525612CC7A6062050D6B3448A28E93B11245520E9138F9E28337EAB428032082"  
02826 "619C82C63A79D1504651ED6F4982E5BDB36078337D9BAA383D2A89E080EF319FC9C53B8BD8"  
02827 "14B0577A57FCF4BE17B9B804045C8889693F09588", // PRN 29  
02828 "B75B69BFAF2260C0D84908BB648C5426C5833985061551799593D3830E45449119E8BC4DBC"  
02829 "8D68EA696079D306921D78F9DF7CE52A4ECEC6D5807B42BA30ED16F56472522708F3F333FC"  
02830 "C8CB3194535A84F84989191C34A80EBF43A1B677233E2C2EF63C7E98AC09E2CBE40559B34"  
02831 "C0127DF2AD7F46005C4C958ED3B051FEF0D64B6DF3C5D210B53AF1A0C7A378E3AEA3D244A4"  
02832 "7F8911D710C943AECC2D3431AB672D878672045797CBD92AB2CB1E2D8DCFDE6667B230AAB"  
02833 "E0070BB8758225E809E91A648173FFAB869EBFA522DCA8C27848D3E7D15E8006A1691AD3C"  
02834 "846959BF14A66BA8B439B21FC76B6EFEE3DC15504DEAGF91996356C9365186A1A40B387C96"  
02835 "FB7B15438850CF2A1308E0C9822D1A33D8D16AA56511FE1C85E23DD47E11B061B432A2C60D"  
02836 "A7B6C5C6C095F7317C76DA27B1123C79EC220ECE197F89E46135CD677C3E4D6F32B3C4570D"  
02837 "5C0F03CE18ADFCE62E926B6A47833D15AEBE1BD891718F8E2B854EF5AEFAE7D01B0113B96"

02838 "A99F59C2D17ECB18A4075AF0E9132AF66F1DCE3EE877B12644C22EB4CFFD25ABE8D43033BF"  
02839 "3C7F26A84ACC6F31802446C8538155986A9B233DE081D7DD0989B463521D1ABB4CB4EBC5F"  
02840 "C8CF9F519E235FAA5FE2BAA7FCC060862B9C7E6472DCC6B90D318A08FF14A86DDABBC760E"  
02841 "418D731CAC44BF82C92D2FBBB595209FC1DE600D49C9D8015955ACC08433F1D1CE4242B65D"  
02842 "13DD2547D6541B10EA8F58690978512AD967CD1B6584ABD9F44385D252C12D2DC6C3D2429"  
02843 "AADBE8869B1B17051A16673171A6FC02B75E9DC0DFA7BA4E1113F84B165FF0C771C75F3C72E"  
02844 "98550AB0C75807DE98ADCF8CADB475D387884FD64DA73FDEA46073E381129E6B35571A09E8"  
02845 "3D4970DF85AAF0DCEAA9AE0C17B3DA2942213DCCCEC15E243C84ACE9B85A5DB20AB1B89751"  
02846 "9751EF3BCA451E1AFA314D6D4FDB665CF6F3B7DC6CAEC944DB1DE75B8D2AA8BDC934A7650"  
02847 "F355CD109F2A9AEC264F3DE2EA03A3448224A66894F2987D84FF8F3DD37E5B8BF30B6182D9"  
02848 "8A640FFDDEFD034D1D8F43FE6811E92FB52A2753C5C52C58D74648C5A53E9C5BC898703E6A"  
02849 "A4FAC21D0D5A8014B64A70F9A9F86500BC84B678B139F745025289A21DC1E0BEE6B349B8D7"  
02850 "A98A89FD78DD2F30FAE4C45893B392DC8569924E04C102325A7980DDA8C6A53A14950239BB"  
02851 "73985536BD3686206BD6456C5DE09D77745054B303B5C03C86DF7888FF2E00921986BA476"  
02852 "48CE6F5C6EB8CE06E9551DE6875E18959B918CC3B3ECC4BE43833835777A39584C97B8C1"  
02853 "59CDDDE9F88E3E4B5ED46FD68EC8EF6F4E0B7301D542E7E3B59E5EF5825D822C97CE3B30CA"  
02854 "44DE46B5DD3D5BD65CD085FFF9B9707B13C2C7CE088ABE5E037EEBCF6384CE11A1243ACA15"  
02855 "F443EA201C43FFC6DD55A4AF601C799A61D81CF00F3CC73723DD818DB17738E18ED1283C13"  
02856 "FD4CA6F92B680D1A73B1BCB534DA1A059998AED115058EB87D60DA12CF15F0082A2F35277D"  
02857 "012D1B2120194B7C94BA141D900B36F35D03BF5FB80FD3CD5F377F9BDD0DC91DFB23C879257"  
02858 "03A9F9551AB786C498A9BD0A4806F813D74331A1E45DB36E330945DCF97242E90FCBFFED"  
02859 "CD8D50542FE7F9E9606A795287746BB6C55BF5396CC03256B85699407ED48A03A0CC3B1F46"  
02860 "E10D9019BA22B02D3CAE2A766394009A27C0FAFBA48BDEF936014F035F7E81DB4C6CBAAC9F"  
02861 "596058D987E08CA8BB3AB72B8A9AD6943060613C711F1FA7BCF3BB05FFA47095A7D505E54"  
02862 "D0D361A08916FF2454195A1E7ADCFE1FF5AD6C6F84", // PRN 30  
02863 "533F653DDE61A6E71AAA977EFF7A55BAEE911662E03DEFEEC8BD919CF1ED7A4794A83E1DE7"  
02864 "FAD820DB9D9A9DDEFF8E72A63E2C3279143D9779B3277B35CCD804707DD408CC932B6E29F9"  
02865 "AEAD59D7612856E0FBF51DEFD264E3ECBA052DE09D55B5BCC81B566EF68A4C198259CA131A"  
02866 "62848D6314F32379911302A3DAAA49DC1CE86DB465F50F5186241C749053546766C75A27"  
02867 "BDE9AAADCDFC2AC87373D767CE9F4F6C58147D8C7077A70323D4CC8991F88B7737D894612B"  
02868 "FDE4219723CC91F91B67140DDA7908725098D878F273EACEEEA4D8B91E575547AE3FD038A1"  
02869 "3CD453CE6970928FA7C7CBBF575EA520307A866C44F9BF74BFDD174BE5AA2C91F1F07A83E0"  
02870 "5C7C4474C1A5D87C9A6B74AB385086BDB089466A5945D4C03C2C118FACDB0D1860C506A3F1"  
02871 "A33A302922A6CF9F1B52822A669FFF5B61F9E6376CD14C78552F8DF262262DE5DC87ADB212"  
02872 "308BB8538E8953916B76D2B9BB2523839BF1B9E2E3A13E4EE516E2D6E487723308D06454C2"  
02873 "BB05CC693F31415BAC3511048525A55F252F720DC86F83119113D3DBA5CF61DBB35BC3829A"  
02874 "001A989DAF8E3AB9675142F38FC71AD47A366347F3A137ACFCDAB3DAAA67B3E9B47C76A3D"  
02875 "85754FF14BE38C8622EADA513349A483F7888C1387E50464DD33317DA61678777D65CD9DB"  
02876 "D2417F588EA20894EC826680CB603C4466BFD4DB7B83EBD095C664BAB158647DE5AF4634E"  
02877 "9E01C8B6D475881ABDDE73997037D4B6D8E6209C5E35D18DCD7D617EED87220487F342CDC7"  
02878 "1296E7A014CE7C8A6DC58451355D417CA1B2E516FF94B5E1929F5931FAB6B36B240A890513"  
02879 "9E414D311D6160ED8EE30F3B4A106C0E3EC5C3501FF2CB186CAFA8FCBF2B1E1337FCE30821"  
02880 "697B6612E5C09A37BEC83F953287ADCA66F09C92CFE7072EAAAB26EF5A6FF8D3B81EAF97169"  
02881 "A59E7991A0EBF2D8064B9E06F04AB68AC257BB8664BC950ACD5AF05B0AC802D1E1D76F52FF"  
02882 "39CE6E62338A245B8370B95131B9E70AC975B029D072C6A5854D38A9D2EDF4AE65909FC569"  
02883 "C4B43442B160F645436952923A7589002169859FCE431562DD5C5AED80C3685B53F2D910B3"  
02884 "D1AC44FAAE81467145CAC12C90CA2936CB6E87DDCC8C27B0A8B403A4E68B59E424760CD707"  
02885 "E9D64077D31A3E274B26B329C8BFC1B020CABE4F2D1C3F1D10A3F3CDD49B394C3FB6B93701"  
02886 "D75F115EAA2835951F689B372C9A4FDA0A36861BE8EC3149E5D9ABDA6FF009C1437F2F34A4"  
02887 "C80459C34BAA373EF2C45A86C89E37E490347110F69A51314E5E1EBE98A3EEB4EF098BCD0B"  
02888 "77988C55478AC408A506E390050B2147334B5836DD6B30F3A82E2E83EC39E6F0B12EE5303"  
02889 "1F6867C1C14F53E3D53F35777A27BD25697B196E0B93F0CBB976F387F5A312A3A8854DF"  
02890 "68228F40B07761E48E179D8A6269194A74FBF5438E7D94F93E12BA1B80773F23950D963024"  
02891 "7A206073D5D032C48E8F5891DC3DB058E5637D67E82A7219A29E0F8EB736CA75F53CA566A44"  
02892 "71504E78DA8A2F4CBB3619A9341AAD66754D2D55644E5E9A932433D64490B72400C6A6C753"  
02893 "C036C48D9721DB697A6BC23D3F6F67C9C05EAA50A89B2E4ADA72B134AD3BB78278FF578A7"  
02894 "D1ADE5A439CC314BD347FCAE41F886653B8F1965E18EABFC5C25104433A51677D53AC398B"  
02895 "0505317ECB059DC39EB4A221471B508E309326100B4D9EA303B9404AB33AFE5722ED9728A"  
02896 "AC5616E1737CBF7080FA0B66EFA42F9BA01DA14E6D312859F2B465A64BC1E76EA8FAA0306C"  
02897 "DC0C29EF13D9AFDD9BC5873B9FE28A2AD938AB2C2C", // PRN 31  
02898 "812B4195D681D927AA8F28BB9CCF2FB10405C9689EC40EAE93B600EB88F05A713A19C5B853"  
02899 "989A1E3C84334302CD2AF7D33593D9862769E8DA2AD07DE85CCB1504E5C4C6A53B81563A2E"  
02900 "89903B28EFCF22A6D0D2A18151C3CDFAFDE9A83149588F37024D67BB1C52670C18C6F4A3E67"  
02901 "E944A62C02F686F7B59C3D1D3966BF97A03F08E9E5641F9FD2654A0B0770748ED2D89498C4"  
02902 "EA6F4ACC0DE6F137C4456E98762DC12A9CBE9CC46D5053B64E91650A74D6731AD0C97B9392"  
02903 "1D54FC6B9D38E5050E6018038FD0F7B6C3AB555C9BEAD41FA1FF8657491762E1DA48CDCE48"  
02904 "FDFCD1DA420D744C37FB356BCB953E00EE1B14CBA4E39CC8737C6F32ADF53A066A8D9DEE2E"  
02905 "130E4407642FAAE0417COCBEDDEA904EC03DC4A1689B4AD3FB877426D77D5877CCA4E7A19B"  
02906 "CB8557F2FBCB536D52B7650599B3A3BB09821EFAFCF24542DB575986669E517E1354F4A04E"  
02907 "72E7E271C2C305C691336C741D1CE1CAFA745041EEB8A604D068CA2A42B97261B5EB81C048"  
02908 "FE5B0CA2F9289366D600EB4EBFDFBA1AFF091D51ACD973786B04D02DE66229E78C1EE1E78E"  
02909 "CBBFA89D2BECFCFE54614CA68DB7EEAE8FF01599B364EA63CCED34E32B23970B0E451A8B23B"  
02910 "F5D9074ED47B2E7C7F9E63594BC70478ADB8E1AC688DAA7F7F8C49AE10091CB5EA9D99A86"  
02911 "9F2AB21E3E6F7B72D372B2CADF0D9AAF8E8049F39DFADA726A264F63C51A87BF1EFA2B05AC2"  
02912 "8339346CCBC81147454391DF6DC6FFFE0C1646E27E8F5BD60F3B84CCBD5837721160FAA612"  
02913 "9E528496CE181681FB70E690AB67D84A36D1DB8A059B7C637F742DE45F2C4DAF3499D9A1E5"  
02914 "4EC0804639454C87C8C0EECEE734FDD47B834D8DB1F1048904DEDAAE71B571F52152AABOFF"  
02915 "4AD8A2DCEAF41E2B6B5ECBEC32F0AA73F79E039949991E5DA837799AE4263C2C26768B9A646"  
02916 "DED0E3DD50CFBD9B9D2D16700FB0558220300568D996609178987978F5792275FB05D86A48D"  
02917 "006F0EDD1FC9F2EEF19781E05EF2CDFF2E266EB84782D4BC345990253054683F05A7C1F265"  
02918 "4E835BE154B64D533B93B906E0DE8820AD154385A3EA770332A026BB1BF4E2483CEA464F2"  
02919 "9AA82D9BD32B9EEB9DD47BF476338A47F34B060871DCADDAFE1BBA9ACF87792D0BA7BC0D39"  
02920 "4CFE8A871F3B9EEF9471F4DC751708A1E1A48181E988CE8BE87466C482B8A4982F59C8E23C"  
02921 "E71DA45F4264B614DA599376C2E451705E88B43CB16507A2A75AFCC0A341FDE0C0A411122C"  
02922 "DF740A0A447ED016CFF3DA2815C87750969D869249CC1E017F4E952448FB7809CBE4703586"  
02923 "19865A2044730BDA7B1049132CEC2BCA2B47D75A3998847F910846AB186D53B5136D6CD245"  
02924 "8EBDA6B0F6EEE58F1477D013F50511E630156491AC2911FFF72349A99642857EA7BB9BBA2A"



02925 "90793EA3DFB7ADE08C191EFC60143FB6041A08A886898CA82015D2FE1CBB661897D208A043"  
02926 "0CC55641CC996BE7F3D6B671BF23E38FB63D799A8582715272D57942EF81E6D5C61D999805"  
02927 "D29B0445E28EB628A16EC2A144603A2453604052FCAE95A4C98D996DF79A10862B86341B50"  
02928 "7C02268E79A8028F00686E8585A70D68A8ABBAC43A24042457BDD5B5A8752F8635409B91B6"  
02929 "E880A103468EC26FDD32B514B419C6DAC5E29E6F40D9D1FF2B3FEF73971B2AFF38E2E5A80B"  
02930 "CB543FF5FB2B852773C3226AB4E642D8E16498C5BBD10CB02BA51B6BB2A461E7B57A3DBBBA"  
02931 "7589BE27E68A68335651914EF449057ECB16B45BD270292AD49ECABA0F9F2067753FCC0121"  
02932 "6E9AA55F177EDB2926EEEEFF7DE13FC5B019425C014", // PRN 32  
02933 "2C4DE12CBDB7F53E4926BBD2EAFCC67EA4F95146407343F4CA19ECAA81C5503E5711D96F21"  
02934 "DAD8D898E60D5C581F63EA7761FB345CC75511CCADC758C4B52DEB658F205D28D9BB6A76CE"  
02935 "2187B3385D843F47D5FAB744ED368FCAB9F4AC5609D786CE6D12842FAD027DB37BDBA39AA8"  
02936 "EB0052154B57F8970162D7912C186F287D96C93DA8B262DA07C712EE1FDEEBE9F1428293F7"  
02937 "257A407B23957A2130F7C1C603102833E3BFBDF5F7C1ADB786508AC1C77A3906A186A0D69AC"  
02938 "403F4D0C903E7520A7E60021345DED3C847074727C8F5C7CB63DE1AA4AD0B5A5ACEF88B585F"  
02939 "B5719409A13CE94D1961027849CD235B313B26ECE180F2AE3BE6A3050EE37A580DF114122"  
02940 "DB08488A915BD5BB9BF09DA2CB096CA28972DDCEE5E21A4699BEA967D98A49E41FA0CCF185"  
02941 "4F4D6FEEA8BDD773B5568FA0D250813E280B4F5B84C734EE10481B9E5B2BD3169223F19CD3"  
02942 "8FA990F211BB0F2B44576734778399E10F10A3422FFB01FA9D5E1A1F80C66CE9630A6CA718"  
02943 "C1C7750DED2AE1C41DC976AB3BD9978FBBEA6A5B1473374A01E80D8C619A01F3E503BDF09D"  
02944 "B1A4CE513F62AAADB43E065D4D99351284AF4AD8FFED53662D0EBACA6A284E5FBB413AC0D"  
02945 "09D5C70EAF493191C584C0C9E42A8DAC822FC0B0B26FE54FE3B3D572ACCF978CD29C78AF9A7"  
02946 "0BBFCF2F8C5C8DB51CFCD2D4DFA764EF9A5193BC6CF5BDBF6CB0EE950CD484D5F59E04F911"  
02947 "0370A22E6E45A1C042EAE5FD0F102F87821AF507D415F06D3F91FEEEE09511AB22AA3B4862C"  
02948 "76CF1AF16E22DD0A57F64C862436D1031D237D62DA3478ED551D620573DAA33E270B0DE475"  
02949 "1FE097F03929E162C8A40507829202459790F9835E2AE3F446C9632A2AA1E8ED2AB7BA584C"  
02950 "48392C926D7663CBC93C3C309D00D74CFBD6C352079132B1A0B7982372D8E516AF5432A2F"  
02951 "81BD970E3A761F1A2C03C9564BB40979F71041BFFBCCF8D3CC6BE7128BC4E44B674EE6F8D"  
02952 "4A184196CD7F6E2AEE80E6FDD43A233C808F5F695DA6AA6B499FE55AE82E392E86F89D12B1"  
02953 "78B680B3DFDB5A478BB981E276B439A63E7166B5PDF7E3CE7D2CE29679774D118BCC73E85"  
02954 "A9C2470EAF2CF98A36A15ED71BE2CD5A81C0935C1EF36CE67CEE9B9AB464DF3A5568F3FC99"  
02955 "E2DB9FBC3C5D71E8FB5673CC92203DF913F26C09B48FD209E5161D28A098432B78C7665994"  
02956 "E7DBF7FA305551580CA7767EE023FDD7EC2FAC65E94E0A852FCE2C4EEA0160B8F4A159E840"  
02957 "3F245EF9131D01EF343BBD8933F58D39B7C5DA3EB73B00D34E8442922C6B84E29179BEBAD"  
02958 "FC6EF1E8D1530364140E1CC6F29573196528C80F6FE620A7DD0FA5A8754457D3F6277F76C7"  
02959 "39ADC5E505EA369C7164D9AD19BB99B3712583EDA0A74689C5BF9365B9F5F1AF984D11C78"  
02960 "3DCB83CD1EA42332D483929CD8ED17D69D0085C51292669F62857FBAA3BCE0615FB06168"  
02961 "C073607B93FC59F6FCAD00A88844064605447BB2DC5E3899097CA81AFA9C1FB8A2ACABEA23"  
02962 "7B94C581230000E95C7D892ECBF0A2D3D61D0DC17293B2A83F9B4B05F28FD3E6973EF2F287"  
02963 "168A10E69BF6D50A8566869C7906F17326A88D1629C2B7F069CE2C6CE10C601B380BBEE066"  
02964 "022228FBB84556CB34EA5BAB3B0D3CEB1CBE967455B04F03E1AC0A214ECB060E14682A84D3"  
02965 "C943021366633D7AF0BD663BF7D82661953ADBAEA04841C4EE8CD95EF29C43A47B1BD33DE7"  
02966 "A44AC193313E79D4E2C4143C3DED323FF4754C948D0DD14F75A3408AA227190FC21E3BBD8E"  
02967 "E630650FBBC27CD298D925A4732257B2832FB4948", // PRN 33  
02968 "759E2CD07F623C0952E2CFAA733D9651B4D835D66AB0F25594BA0EBE210F9EE20994E1D1CA"  
02969 "A64B5167DADD1442FA8C83A4286530D52F2782938D0631396035C1982E5FE67B5A13F1DA34"  
02970 "AA2CF78997B562791BBD8368FB511E0AC9AFOB11FFA6FAB33AE6E547F3083B138ED810FB9"  
02971 "775761876511351E0F3917CD864D5FF1F58D33088F8EDD6378265F42B2C645FB31F4C58E7"  
02972 "E355C4B4D992F58FB8DEA14B2F3F9D68F0E4BE882706044C0B2FA8AC4D143A7B5520D89E8B"  
02973 "851EC915B5C4433BE032A5386A447DC39D6B9E9A8238E9659405CC6DB308273BFC2791AAFE"  
02974 "9FB5E7B32321076E3EC70A2EE9F02ACB4B2B144FC625C0E936F4F466140506835DB7412BC1"  
02975 "829E760EA120B27F4AF273A8B232963E6C757B3EF039B08E8F3E7C5929DDA32DF9A6EEE0F6"  
02976 "15467D390C46680CA4FB4C7A1CFOFE06DB55675F6182A37D2B83139200073AD07C3680C85B"  
02977 "EB7C494952B4FF7B6C855D8C59A56DD7C2888A5E5321FBFC4D8A8C37B3923AB5458DFFEOF2"  
02978 "7400CE419C43B9C3767DD0D7B46F963B6A28B7AB378EDE6611B3DCFB1F70EF207D1D81"  
02979 "023D45765F39597FB8D0DAF8B8DA1C387ECC2EBF1262CC7AED32BB9F96A8AAF5A2DF751C38"  
02980 "3A486F3FA629E0E2107875F7050FDA5B0E79401047F05BACD37992876C0A64D44E0CDFCCF0"  
02981 "7DDE9BDBC934FEE9C8B54489FDAB5D4A69ED3C9FBE2E4B2389DF0371E46D330B84FBB4718"  
02982 "1F2470B1AC1C4C1D60CDC05584308DC7AD76F5DE8FDF0A4E80335A3F672C29542BE6B31E2"  
02983 "8C4B35826694F43867E186A7F550D4C001BE786FD3206A13AF1D2FB0274DCF848C487E3598"  
02984 "FCD26E30A1C37DFBEEBF3EF29A40BC9F7FEA2CF01B299E511A640248881600CD04D37F0DAF"  
02985 "C0B2F473567FCBBA952591129FD0999A512AC95819F12D05D4306C13EAAAD30309577762D2"  
02986 "2F461B0AC5C6EBB80F0577EF8A1C651241B55765D5E588D2B88FD661298C8D0B0E6FA1FD04"  
02987 "36844C5E48D33E5F05B6023B60D445E85E75A7679365FC8103D93EF86614DC39AC50434D33"  
02988 "01B36AF1DF2C9D0BA5041F7D95F565EC1407213BA3F93D81730D28470D26907CC3AC7D30BC"  
02989 "ADD866117768B45500E638BF4B2C6D68A773FD06AC11BB6FEDE15ED2BBA7465DA43AA83"  
02990 "ADCAB61D20CFD0D42B3E795D66F9FD59BA1562EE2944CA89E6F710C6FF9E58E8CEA42326C8"  
02991 "FF4CBA582A812AD8836291A95C743606BBBE8625F832D4B9846590BC5BA7FC00CA0BF51509"  
02992 "8FBCE513F15D74CFB8E794FFFD3A36F35DD7BBAA4D557A2B27959D54326C3A2FE68977AD66"  
02993 "D2B80089EECC8B8D0DCB26253BE3028761F96D4B4B041620592A4FE11D007F2DDE67DF840FA"  
02994 "A4D32BA0A6A607298089C2DC9BEC00DF1D513D94BE286A9FF1E997F9900C39E8FDD462A4EB"  
02995 "AB39D05E61C079C1F7815B4D2985BF0855148398A8916840087D9A5E8B4CB61AFA78AC1B26"  
02996 "D99F4DDF43F672703DD471255BEAD3566995BF0F73553284D94F515902AB03BECD088C2744"  
02997 "D3E8226F4D92F27666671F6428517796C5ECF0565BDA6119E2E4552C06EC615E8708D821BA"  
02998 "EF1F372C3E457553673F930220B615D5D65B7875979B10CE16CAA10287E88A7BEDD89D2FC"  
02999 "EBA87B7FD2C5268FB376987637B69F6DE3D7391C767F2628EB499969204978C9E5F6947F23"  
03000 "0201790707384D0821D325D48E873734159B82CF59805F0217D7EB1D40DC10E9C60968349F"  
03001 "1B513BF7974248E571EF536CFA32300721F1CEDBDE15137DF780BE77607170E55945067A0B"  
03002 "1F1F30DDA7ACA3A06BE60A63D7C67F1B551B830E10", // PRN 34  
03003 "7E643457BE98ED5901D353050342D296D9EE39DFFD02D12F0BBE04F68E1C5E8B77E1247BA8"  
03004 "E8BF60C260F4DD306136F61439666BB07A7051046B44F7AF36D23429B1C3F54F782962F71"  
03005 "419E71897F7A41255E13C296091770303A19C48201F28379161E2F0073369034A2EB0664BD"  
03006 "E67567BA7E4D3BA2C2FA28527D303EB218370802163C28C61A9BE0C5E01011E079A4572E5"  
03007 "C343498D0FD04D758023B18A085FF74E9CEE9EE5E259924C67A507E16AD3164DB057274D"  
03008 "EF697E1D1CE51E790FC85AF4BF5B9525E0AA29FE86FCB44390365607998623CE35E8C9151D"  
03009 "26AAD0E648E79DDA8C81AF7CE1B97C97CAA774F56EB725DDFBC26599A50DDC3F474BF804"  
03010 "3870C0658263FBE79F4B130CE511A96A2A6A18C71FB439D126E0E7004E5DB95204756F886"  
03011 "13F082D4C1E817E28858DDC3648AC0192A84A0430E8C384603D08EBC47CC67E041AB7B774"

03012 "566176779377FCEE06FAACA8954624CDA4C73E0D85F640699D8765AAAE55470F114FEC7A60"  
03013 "1DF55DD053A5769EF190B5EA9FD0C66E3DE2C4B852ED66BCC9A3B4550382C4037CFCFD0168"  
03014 "60A1E2409AE75C06D6556627A6FAAEF4096BAA4EF8BBA62404A7E0748ED0EA6F38FOACBACE"  
03015 "612F53ADD502B1DDB6F71C35D7E2C6B06BB0EB3D9C546ADF66B47498DC20435E97C94463"  
03016 "F6480E25585B2C8712FBC610FEA0DD17E0F8C144A468DB9B71DE49850BFB61225180B95E9"  
03017 "9950FFFE2162DE93DEA411230DAB5ADE80CAEFA00A1F0D3403513AAC273A5BD070572731C3"  
03018 "94FAA28A37F6D2ED499D8986B89E63F17A5A1C63515129D588DFCC2FE2D6398510636C1A7F"  
03019 "1AFAD0A73CED3FE41117002417003ECF249D9C3AA9D47D1FC2199142D7F7501A57898C5C03"  
03020 "F3EECDCAAA4AD93F8C83CBF904080480544CAFBD2D7D530FB7B4F019F94BBFD600E19485455"  
03021 "C0204C74E2BE7A3F734FA5DAB605FA9A6F17C195445CE4468F4A8BC48D6297FE6D5EB930B2"  
03022 "831ECEBBE58B8E388B104F9EB81BE2D6DD87C7DCE9B8C9E865668FFC42C0F50418703010FA"  
03023 "08B8706057D4E060EC2069ECC456D28794F4C561BFA5A932D496D1C2A6B4FE4B4B448C033E"  
03024 "0D86A653BE48EDC2334CDECC309B8D43BB32CA35A11717E291ED206C4DBB8FBFB09E539BF"  
03025 "A4D96680671B15098C5D36D3302BFA6CCB265BF1E1D3F9EB07E73529D24257440CC7B0A07"  
03026 "C3189084934F068351A82CD08883926347BA0049BD18F7812A02B4E4567071C3E8862EFD0"  
03027 "FA1294A321CD0C6E90612F75340E5F1642BB3F11D0C35BA7789C02A03D73673FB4486DF539"  
03028 "B80E263FE9660EE962DABE1DB7505D194BB44E211BAF2D8FD1ACCC5A246BC4524F37C9FCC9"  
03029 "84CB197DD377FA46EABB86ACC063F36E386D582A8BCD28CA790EE2FDABD07FA7B1C4CFE491"  
03030 "B819302F2D8269BA6806F1D8060936DE375939480844AEF32400923E7D833FB8FFA4B8E951"  
03031 "2CA687D00E1659C768445C24F19733F9BC0673033777D430CF6AFE1039D5FD9D968F39BA04"  
03032 "2D99C4FD31D9B3F2D49168F23849C1C3D6612C7CB97371D0495723AA190CC10F237DA87D7"  
03033 "0772AC1C6C7C2C70AF94A3D47B296AB4B3DDFA887FF664676E1B79F3483A731AC0CB65AA27"  
03034 "1B9B7A7F5165FB0F2C3F276A0FC9E5AFDE12D5BF7A51F4FD85FD7A997AEDCD0384FD9F3A5"  
03035 "50548D2C1CFF796171FEAA5AC8C2CE35DCCFB0D994E035652C8789B8352FEC79AEFBBABBD1"  
03036 "1A7762FD01748AF96F4A83E5716971E119C3014D747FBD91352956FDB055C7E5B6483EFAF0"  
03037 "BD69494327478D8ED9FCA239EA825A3FFAD118AD6C", // PRN 35  
03038 "B43640BBC972AD37CCBBFAB07D9A72216905C960FABBD71DC692B3E98A32B27C7FD7350C"  
03039 "12A537B30427EC4C344FD54E6223EB77A11D6784A7AEF66E24CC4726C5F11BC163D4C44AB"  
03040 "496DABC3BD3C25F745B35D27F570DF8EC62B141CC4EAD5A50AAA749088E8C7F0F78B2E32E5"  
03041 "980760BC01BF7322E6A3C81127490704617B67EB84A5D802DCD74DD8A30F6DE89AF245276"  
03042 "178C04147075FB682515B9CB2CDE9086AFE2A54B9165413357373FE1EF0EEE00D5AF1D795C"  
03043 "394902C4E255697BF14E7B04C1EFC1312CCD0222BC924598DB55462750EFD69F4EA5B3ED9"  
03044 "6DB08993E58AE16798E1C008FA8DF3B65B81962E09BB2CAFFDDA2297753A9D38E97FFD554"  
03045 "CE382D61DB9EAECA9555310AFD99CD87C936DE30E06337B8B2A302B786DEEF1C4C5703FB87"  
03046 "9711E4556032D6C6C0D9528D714290C910C601AD65D210C5B815E4C1A2B710EB8EECA9AA3F"  
03047 "F311B561D3D7C660106A7AAECBA72F6F36780A4C806E7860393601A1742309EC34C8691F"  
03048 "A7AE1BF4704B07046C7D77C0D78796D5C35D378D15175F23B9CE6D5432958D82701A678BCB"  
03049 "E19C90334CC8A4E89199094A56A502118AAF9C070254E6A9D0CB2EB85C0D1955FB3D2C09A6"  
03050 "136F8B11DDE4DA7360C3730B8DAD5C27C70BA8ACC1266C9E499018D4E860E4A40A8B430FD2"  
03051 "7165385D365D3086BF443D63953F269CC81CAD0E0A7C5A49C856B24C0A278840CA9ED5E4F7"  
03052 "667B61E42CBD157C9905DE937D6BBF841F36A69332CD4BAD90BC2E5992D4BEA4732D4076B2"  
03053 "4B9437BAB310767734FA9449577732A3DC3E81185487B57D42AE400E5F551AD00E10A15ADA1"  
03054 "6C2634501CEA05BCB63ECD956081609485F2AB31E1F8CE022BF8AD3861ADC7A032DF769A1D"  
03055 "E3C86B775C3764A9554CEF675A7AFB72918DA2A4CBA8BCF94BDCA97819FCBD2B89D0F4A960"  
03056 "51058852271C5EFBD62ADA7EED9E20F9F333578D1AC6D49716CE28936FED5D9D85B9230C"  
03057 "DA5669EE098732591633983DE7BF41797C739A7E8F4D0FA30A60105D32FF913ECCF3F3E4A8"  
03058 "2EE93CDA04D4E826C045ABF89CDF67CC0BF70D32CC08656C2D8FDD2E23AE40FC2BBFC6BF258"  
03059 "F006C1597AC3143BB9637B1B7B0B0ABC513D7F96F13783F44C0112C0F2C35C09A46D27AE47"  
03060 "F0ECB22EBAA5DA5C4FCBB0923E461EE0A524DF59E83A791FE9E7C9E114235ADF9F17181583"  
03061 "888051571C82A778883DF35956A7F3BC550E8D06BB7F158A82A708C0382B54CE19D14794CB"  
03062 "7CAFCC6F782A3C3F6C1961E62246F5EB285667E32A940784042C7E8A3E3D661BA42CFC0C"  
03063 "B2414B37F5B17CAB201C1A7F894F07CB02A9EA757342F76FED55DADD334EE8BBA9C9AB962D"  
03064 "1F46FB47EF3B1AEFFD33961E887692B6E8C209DB819D4DAD1721A061F4A5C1EE753B518CB5"  
03065 "B56BC79BC426A6B82CF2C891A999566AC9B3B55D7694112E8DC14673ABFE7A6D5DFC5DF1FF"  
03066 "B3C1A4A181D6A087D4EEB0A821C250D0EA92D760A9ECD4FCE4B6501735D33B60F462FB971"  
03067 "CEE90C706B610EA022F00FF3ED9E48450790EC73469089792760D47C84A218E11A557E30"  
03068 "D036B8AC2A8A3BAA021D0E559084E4DB29EAE90A27EB3FB266BDDC99B3A9918A02C7F7573C"  
03069 "1638902DDDCAF7C72A33AA9F7BB6BE755240314D1459F3069746B824EEFE49B2019298FD52"  
03070 "41C75EC5E4A78893D7D2B9FF9CA8D0D23D25159277C240FDBFE93D90005E40B2B1B9A28662"  
03071 "F81FE38A044817042138003609639C347DC84127EB07966451DA55ECB8EC6AE03F23ECB42F"  
03072 "02E19FA2CA20DD145850D2DE8F0F8F4334CFD2968", // PRN 36  
03073 "05671B4FD9D018B1D63A7BEEBA93AFE4053484D26C376C757B1DDF83B3CAE766368B02886"  
03074 "DA4BF41DAD163D5610310233A45694AFDB4C7607F8F20FB2D4791F20E73206A8AE3DF636B3"  
03075 "F0BC8FF71DF1A8508CF7E369F3949D58EE11E187B6898FD239C01BEDF4EFA0915CFE0B1188"  
03076 "49407D794B8EA35D01AB0003E755900BA5B1A1207705AC2083E861E6A9F7D9D9E688A34291"  
03077 "20B95CE62698733522093BC64EA57CA48F05161C0FCB369B57F16C449D3F094C0351F8717E"  
03078 "FEE5A576454D3023DA435C8C553F47F86C505A16B9DABFF43DE50091F27D338C78AA5ACE5F"  
03079 "21F90A9E0451D0C5274FCC435C5CFDE6641D1CCFA50F2841832895FE01A95868661A8808A"  
03080 "B763FE2F8781FB92EBF1E946FA7ACEEAE032B8D5AA70933A05B361E275DA3C0F29440A456B"  
03081 "2999B11450D159B0E3BCC5E609F20F04CD1C65F35A98B69715A2599EACEAFF382FBD6E7F92"  
03082 "D3E3BA84563551259A186915B9EDF350EBBD785245BE26ADCF2556204827028FCC37304848"  
03083 "C04BD24709492F5DB034BCBB7E56CEC0DEA45AE9A4252D0510C1AC90F31B3B438D092C0444"  
03084 "710467B631853766ACBE05BC40776A149F6CFBBA920F7EBA4318D0BE53CED07E33C4105A"  
03085 "C204909FDED308608670A735FE5A6B6EBE4435A18A031D4EA332912C16AB7000E353ADB0A2"  
03086 "659F09846A552B93D6DAC1A906DA744518305C2280B37691BA418FAA976F301D49E091B5B4"  
03087 "581FD89072FE76670048AA5654A98AC42714E55D5323CDF0417F51ED4560AF43C7B8C5047B"  
03088 "EFA37685D08B928490121344AF91A907BD17932698CEB7BA24EC0B559E151E4EE69600038"  
03089 "4DFF1479BA348CE1328292189DB333AA9D6A62A44250ADEAAE4C7FA208A6E2C53C6378BE2"  
03090 "9D294043A97AC937B68E13F0F093F543D3BB039107B4748366EC97ECA5DF116B706309C98E"  
03091 "54F88411C7E13653FFB21429559DFD1A45A85189D11FC52A594C2DB73864BDA9326170363D"  
03092 "AE363A5BF7EE1E91E87C6E5F5246EA4E6833D9BF6A943082C3CC6130685E7DA91A3B401E0A"  
03093 "DB307FDA70B37988E2928B07E5FC572440B6972B430C13053F57F2483CE7060076FF2847DA"  
03094 "B897C247F40D00422867A78C29A17F1AD28D2E4EA5FA14596714FF3CC855EAA97E2FA33B46"  
03095 "E0E34A38BE1222F03964F1EF6CDDFFAF7A18F752894857F0CA43CF34C21C2E69135F5FC960"  
03096 "BDB8B8AD3481209D9FFC08CBF3C2807CF6BEE56B4B91D3B1980BEC5F52D6C56B4FD36EBCB3A"  
03097 "A020541BC1BCEDE9A3DC61E72F1D79874F0808FFA7A8F2FE1F183B77F4CEE38C9AADC2995EE"  
03098 "6F3613326410DA4BB99F7B33B77EC19F65C0072C33A38EE623FDE3C9A5D2D4E7C539779A15"

03099 "1C1D84B109EC2519EB625EC61D2ABBB54C3C9F7629287DD1B2FEADDD16983FB9492D4A3322"  
03100 "6BE55804A4C7277299CD3003CFE8F2E7FA946F749DFFCCCE465C7B2BD251DC4B0779D543CB6"  
03101 "873B6DD03F82B09BE43D8F114E2E71F10DED5AEF9EB1C266C5A9B7F21CE1DE0FC6A86FB705"  
03102 "7FA75FEB098E041C95B2D1912849BDE26BAACCBFC105BDB8E6BCA0B8F0B88255AC0DB9854B"  
03103 "FD8526E51656DE55E7968C4FBCBC1FC32D8BCE5BE8F1B132F1B24FE049C2B3C18A8A7C1375"  
03104 "9C725AD2957D273ED0BE3B34B0B8A14F173380BB9CEC925086B2734982FB234F3940E42A"  
03105 "82FD196AA218C1D488915144D8653B5EABFB18AD35223E35CA403A17E8FA0C1028CD048672"  
03106 "7447280E2CF9046EB60607A6AF84C7ACC03A7321AA3F3787423B24235A2191E0F13665A3DC"  
03107 "F1F5C0E964B7FCBEBA511246AE50B9DC1EF7254C30", // PRN 37  
03108 "8C6FE0D3BF7763C15CA86E3AC55695028D02B6C3411F6DB117FD982A5B7F97D0C1C05067DA"  
03109 "46BE7FE39687CE4DCB4D15D89D3665A24932CDD7DB871481674DA85C2982F5C896E7F68345"  
03110 "DFBC22CB465F43503EC3D7B5E0A4F17169D7D94D3A64B27911DC34DCCD3AA7B3CE2D8C9A05"  
03111 "D75BFCFE4BE5CCA3DD71E7FD059FA910B71A2400039A7EF536FEC7E93D4C386E7ACF1C8876"  
03112 "A27A2B14DD49A0C6CE150BFF180283B6C3CB228DA9AFCDDDE6A9C17FDC03FD7D5D28D8FAADB"  
03113 "B2CA49CDF63D269109EB0A7DE261123FB15D6108939B231070F60E174D30169D2155DC77966"  
03114 "1C570B5092781AB64B122C14CAB4A511A493DAE03D2A8ACB069F1FFA0E83C03837E62C8923"  
03115 "159E6140E237885BCC6675F906CBBEA9A5B5A7B6AAFCB7D03C503032673BCE6016096E0C1A"  
03116 "2028D99D61FC40EFD0AAA3ED0911506214B0BC37F6EFD7BFC05AFA6C3412FE72DB52F82DDB"  
03117 "0698B9AB4E5219D072CF803D52058E478324C523D765E70A277959F94692227F3123B9DC1B"  
03118 "EEDF124518C2362D35229ED62CB10E32CA78B19AB39B6B45D8E6B8AC9167157A715093645D"  
03119 "B3B545484F7B8CFD96CE19DBF8DE9F9F7B8F49EBF257472116CC6E453D417E0DBCDD11D2F"  
03120 "F7ABA5851DE6CBF879B3F61DAFC00EC007F9BC702A557B200FD0D0F3FC5C86D824B2BF5692"  
03121 "117CFAC14A34BE43AD48B1711ED9664BF1C5DA02595FFEB35A1963D41365C713AE701AE2"  
03122 "6B0E544B19A7D6495A9D21502C43EE1E7A981F2EBB1CEBFBF6D282BBD30B545385F58797A0"  
03123 "ADA4435940780881518EA6E2FE6B5DF96CEB4244D183D53E038B09FE6398EE7CB4D04FE340"  
03124 "C370074190316EE51E109740EB87187EE128F6BD1B8B263D0DF31A95B2FB51CB9C9CF34D75F"  
03125 "7013297069EBF439E4E5F7E8DDAC4FE0D6E4B7BEF0DE04F6759908BDD0B787523648AE1046"  
03126 "B27D06DD342474FE342EF6ECE8B156CDD8E72BCF8EDF8D4C9D1E8F84B282CFECB2BC08EA0F"  
03127 "86EB84D460B4F2C82B3DCA5603A76C46C6D09F213B62F8C97A150328ADAEE4565ACAB5CF0F"  
03128 "F721DA5F86181BE382E860647E1ADAE0A020907DB64E3AAC65E916B098D78524C9B9EB2F20"  
03129 "2C34E984C44C8361553AB85ADF025B5DD735D175B7A9D202844AAB1990DB99F93D92675FB2"  
03130 "21E99E3F1F202CF72DA0D0E8D525850628E0F3CFEF6C8C5DC04E935FF0DBFEFEF1C33A75340"  
03131 "E07AC5F4374195790B1E8E044971051F714988BE444407D944FCEB9E6C0E0C406E8104290F"  
03132 "185B0D65B39C0CF1F8C682DA4325FF0D9181049F2BB7F2235FA92673D6550FA1CB222390F4"  
03133 "286116F6BA9603F8BCBCEEA18BF472920079F0D2AD480A36D1951E9F1D25A9D9F33D3821C2"  
03134 "FD73433CB6E1F509E358531E14DE20448FBB68DA0AF81C95CAE30E4B65A95B5542533072BF"  
03135 "B07180CCC1E2783EF541CDB23C0B6AE07ABED0E4816A8FC50911B41B56A8C29A20C26BF5B0"  
03136 "08164524DF90D50B4D5EE3308FC42CF99558EEF1FD32EE9FAF0F1D1CDOCBA0876BAA24C70B"  
03137 "8F74D7E4B3A963857DBC6373C681767C87E8EED8BF91402D157619B7A22D1ADE73002210F"  
03138 "19303781021EDBB8678DA736FCA5253174E271BF93F55C31A444B63F19475DA09C4F5A5577"  
03139 "663660FE55C54DE04DAA4291C9545554F8C59299560BB9FA3EFC8FCF2FDDC328526AD3C0E9"  
03140 "E583BA70F5C39CDBD23F39E4B7D190CB07D74CFBC8263BF952EA8666F98110375FDBB44344"  
03141 "BE1D1ACE3E64D7BC42ABA097A8411E6BA288489141957BF304B23B9685500CBF0728F1E7BC"  
03142 "2622F8DC693A8A6A188A57838050974098D951053C", // PRN 38  
03143 "978D4E016D006EBBDD30E25C1E0C5843FFD41A9B72D83E97C511FFF9C43E6EE09C7D3EFD6E"  
03144 "F5E1A76184EDDAD3115D2C7679B2E02F01F6B1E96E2386BA2FE579879C428C3ABBE9D8C91C"  
03145 "0C71DE099396744BFAB7AF2C21F4DE6148B34967F9D51327B23E207C07F47BED4B15327179"  
03146 "44D203C29BADDD646321834B49B9B20F8C5897B20C2F383202ED43A842F76BEDD254B072FD"  
03147 "093E9ADEEC149E08892127E97F9CA9A3AD97C59B8446A880D6766C2FF2C1D596E4AF1ABB47"  
03148 "E5F70EF36539AFF10FF64A2A5AABF499DBCC1DA9733D3D374A17FB40218EDF9048930C69A0"  
03149 "33D3C4AA95F7B43E6272A138E501FE02FF900DBB21F5BEA5219BAFC0B929D1757D33DBAE6"  
03150 "341A78651801931C18C36CA71F5BB2B0BE5F4F481D49AEF2047DF7CBBFA95052C9139CAD30"  
03151 "E45B631F7C1263C8ACA26870AC1C965173ADE2775331146A8ACD3C28DF23D8AFB30292CEE2"  
03152 "90A6349B3535DD0170218A4469FF12A2848235360CCF2F1CA7AC67091178EB993701A8AF41"  
03153 "E5DCDC5E1857D493C98E144188CCE36C19450B57FF60CF87E13EEB4F7BBEED3946421C7EC39"  
03154 "666CFE0B4017ECA70B5659B39979DF346F91E8322EC121EA7A41095BA7234BC65C5E9679A"  
03155 "30104C74ADC46C6B88021C238174087B86C1EEA1ADFFF81CCEFF4E9D4205D20772524A21"  
03156 "88F7DD8C4F509CE8DA71EE773237F39C2895A2378DC6433E8F5B4C9F54F7CFDE18C8C02FD1"  
03157 "F6A66519A35E28016EE448178B437D32FFD3044FB513B913B2172F1B20174D344EA0B2C0D6"  
03158 "CCF0B1C286A530126083AA283BA66C5A18B37D3C6DAD86A2E95ABECF1CF0B31656F1EA5C"  
03159 "97A29883C9DCB3C739353FCBCB2558518D282FE6D18BEFEA9F38576A912376A3D4108319A5"  
03160 "F43C5A78DFC2C609F32FD639BBA98CFC4BE1BAB7E4CCFA0C75D4BDE947DB3DF9753B7FFDA5"  
03161 "8084C685DEB8EDD82F1503F43A0F8C25A232588A8DA22810307FED26056AEFC2D6795D0987"  
03162 "0D3365EFD81974E19D6F1BC0033382D85175F218B35CD607410092F52AF3B635F8FEFFD32A1"  
03163 "0CAD30A068BC9C08456D9A5D676B15605364A539F3E3BD881D86512D0BEC541D2F679A65C"  
03164 "B0392101681CF68A63DE7EEE31648F832DCDDFB74FF3848C2910997AA3E8ED3954C38DDCB"  
03165 "2403DC3FE8E760D9903E4F0B9479CF2C5EA7B191E1B2C353E180FD35413F53340C89A96A44"  
03166 "FABE322C73005A1FA423D2EB197089B83BADEB70DDE9FA1354B964C7236B4270DC31523340"  
03167 "0972AC792178FA4190CC132E9AE55FA4768FE302E0AF7220BEA4C82CD04ECC17A8675CCCEC"  
03168 "BF2CB4EC6EED8377795B391DDC3FA3C59153B0C3D196369D7A29EC2EBB911E75991C56872"  
03169 "657C853ACBD1DC429CC745711B00082245D617C2C323A86B73A735BCF2B3A9ADCBF76B991"  
03170 "6680E1F35A59CCF6493F58FE9950059AC4A26B3B26282D8AA24A0DBE150DCC8A7A455DC82D"  
03171 "76F8EA2793A160FAA04592B2D85D4095E1E62495DF2C19F21D61D2D743ADB2833B199C549D"  
03172 "782775549992915FB4DE07978E3A08E441FD938008033295CE4840AA95282B36C02C786F16"  
03173 "F51D935108355D791EFC2AE55270355DA526CC1098FF41E179BDF5C72E5B5BCF38689732DF"  
03174 "1AD18E90C737D2DA573C087268811047093805B01FD8082741E2A63B4C6F7B38644000FD10"  
03175 "70A470D54663C5398B2AA7BD0735BEB0CAEDDD8B7320C771F06CFAAF8B69ACB189EF86D9AD"  
03176 "344E9EA4BE14933BDE687D9B13E29894CB2720B28EFC50F9E12173F8072ECB4E99960A1777"  
03177 "6459ECBD324AF8BB76C9B8FADF41C2C0398030FCF4", // PRN 39  
03178 "5319BF9795D7FBB2BB137F880EE3295AC2E4582E7330721444D4CEF185D010FE362980CF49"  
03179 "AC68918AC66E8DC9DDECD857FA69DE0989E1C1BAA7FC525D07D84E07329FDA7A0FEF713E9E"  
03180 "B3279139A286FFD12AEF399105192599C5DD57B18CD6EAE028CF605B2DD8C774954D7213A0"  
03181 "109C34B23072720514B6F96E83E4A2798318AEE1B3C42296421572AD26784941110DDE0F02"  
03182 "AE847BEAD08EA15D271CF8A6F21DBA4D586F9E2A6729684CD27462AC84C3A8B018DFFB46AC"  
03183 "ED0316A27B3EC714933A131158746A09BCBAF89F41247460990CD87F4DF22776497306C644"  
03184 "34F69462CEC23597FE9F0B270631F0BE33C6F11CCDBA3C45B1F71BB159ABE23F15F6C3E1"  
03185 "ACFAB35B42A839A7ABC9CB05341B206CBDE37BAA1EA7272BE207F46707139838687B69BDA4"

03186 "76E44971F1E893804E04FC8EBED3F8B86294CF330A34066BF1AD3AE20173DEE10F94C8A11B"  
03187 "E66706BDB8C6376A400F1DC2C4E5B33043C643887FF010443444F6159D87C6EE703C505E5"  
03188 "5EA810BB9E3C65381768A1A4B9AD208A4896AD730990EA950381F6D93F53EBE814ACCF92C"  
03189 "D92DFF78663D43BB4F089EBA44BCB30522237EBE9EC5869E8EFC22326B62BAB02D863A7976"  
03190 "0AFAD679BC66F1CBB164E7763FBFB2519F8D99D0E759BCA234E4599FEC0985D9A053D9D733"  
03191 "CBBDDDEA4952C08A3E9180331F55703371284022AB57598788C8CE922ED788C4AFE1D3BA0"  
03192 "218EDC4CE5ACD412A6CB1A078CFD47538A3FF0B0AF195B42E1A51369D061BF4C3FAEC1B383"  
03193 "43D5D9C4ADBA22C906DE1D1C419868BE94F2DD34BC5AD40F14C288F2850264DF35ED3D9096"  
03194 "EC88E9065C5E031541BC72C32FB151840E687B697745EAEFD00020567F7BB2A3CB18EC6B5D"  
03195 "C4E247CBC5E93CE93795BB296D0A50F835CE840A0672A486C95DEFE9CC3DCC1C2866986828"  
03196 "B46107708BDF0795ABD57DC40CDD1C63CC66F9B0C58959160520DD6B345CBEF24D9796E102"  
03197 "59206B932F7B42B8EBD890C9FFEE5F7E149606B2D6760D7C94F0772BD7C9A32785A420FF8B"  
03198 "11854022E2D5082C6FA1ACDDCC173CF98A5E91DF8D82F0BBE27D9341F1AE32FABF650FD800"  
03199 "2B0E123EDB13216D464E3FBB0355015FB86CC9C5695A8B4DEDD117CC6215FE72DB4C1D444"  
03200 "5C5408E6D004B2A95F5F222D1428931A556C3DFF7CC46BD0870ED8D8396703DA8D16637585"  
03201 "425E0151523E3122E7C4ED986DE71D41EB795927C0E7503BCB36887E6925DFFE37C9A37CE3"  
03202 "14E76CF8902E63AE5C652D7BB7785D6B64474FF02D28535360CD1CA75596FBF3ED736F2AA7"  
03203 "286950AD5F417A03509A53E883C21E1ADB5AE2BB44B2AAA8164976FA5C258BF0EBD5F2B06"  
03204 "768ABAB6994AC24273DFACF41B6B8314AF2F7E07E6CAB02AB230B854016BCCAD1666E4A403"  
03205 "641A8C1BCF818733B1E08BDBAAB18C165DF851BD18D03C4C0E6E1FC48DCBDD1E6B5AAA263F"  
03206 "C5B7B6FD4A4F5CD5693F3C223B4963FFF2BAE559588ECAFL42A10ED87A229BCED177C0B4D"  
03207 "BC197A82131008D351A580DA5001615421B12E1113D7CAB59C0D38349663D898ED7F6F2913"  
03208 "CE4275C3C4BAB98678D7757FE1017159993891836AE3DB266F685FD9E726374407C2B86F400"  
03209 "2AD6C8E9DD5FFA858CB9A416605A45EA8A0F5F7794925A54B42C60BD043AB0C5F182474B"  
03210 "414CB83401FAD574425F8D1A5D5D85B5D768915209F2762F2FAF53C4A1640899C0BF97D952"  
03211 "B52ED5DB34D3F9CFC9BD5425071F6B564B913790600E6AB98C7C100F4C460FCAE4583944CA"  
03212 "DC1CA5B7D85D876C88AF100598722CCD82A016DE1C", // PRN 40  
03213 "5164996BE5360659B3E767D910E441BF5E28011571EFEC8423A22BC7B7C959A9E532A3F025"  
03214 "D09077B2664222DDAC9889FA15C5FA0150C8030322B9DAF467B854D0FA7DEE36BC529E596"  
03215 "C35D067B84AD6B202OFF2AF5FB08F2276D7F013A796783C2B7354B92E364EAD733771BF18"  
03216 "AA545E1358D262918609D3000E1E523161BE3A1FA10B4C7F532C5FE97F7D5B406A6DD8C5DC"  
03217 "FE82355646D4CEA439DDC3E61CE0043A8C4EE5B2229821DD52CA33A20FF386A8FC37D32EE4"  
03218 "4D94E17279536DB0285FC64B43EAE566A52D369D867090A9A9D507D78B67D1A10AA5AD5C6C"  
03219 "2F1BBE87A5C9208241DEE1C156FC9EECFAD01F8C108D60D08036197015D0C96CBA8DDAA2D"  
03220 "ED0BC52A9A04B74B9123581825B96781CA3F1CFDE320823AFA3201A90A50FB65DD53DF6CD5"  
03221 "3D64426D0630E92DD4EFA664AF6CF38674A177275685CA8A1CD7A600E8A47F84308D76D"  
03222 "94D05489ED157707732848896ABCA720CC13362880413B6536A8BB9E43D29E3FB4B1F97715"  
03223 "285A68CA909B5060E53E4C2CE4AA3C0845F4D69F2BB00D26541F58AE97532DBBB1DEFE758"  
03224 "14421128E41139713422EF3FD13E03D75AAA33B67A518F4B0DBC76BDDCACBF562BE4D4598"  
03225 "523EDF76B409833381BAFAC5259448B15B37F4C031FA60A64ACB67220EB7DB531CDA8BD7B8"  
03226 "2D4C3BF9C8CB9D2E0A7441C64A329A7EF5E2FA321C67E5DE16D82DD36AC6098A7FA12D7691"  
03227 "17DFEC070B673C97F2D8147B7B9F710AFC887BB953C486EA446E7C3B54B6437EA5E246D7E"  
03228 "79C0CBB9F767EFE31BF66D9E95A1D9527F05A84CA63B59A09BAA537E97BEAFBC5648CBC3B7"  
03229 "FC7F526BCF287D49DA8BE2E1AA2C9518B9D2E67CF6143579064B33715D11225FD754F48031"  
03230 "DB3373733D1035664FC26FF055E98288FAFE92C8623F8B0C2966387514B649F4EE8BD8550"  
03231 "EC31E9F11415185413354EB92DE3775E8F81C6C3B11ECDEFD5F46F3C295214DAD2187A5848"  
03232 "17BDE578D78A0D1BF64D50484EB8C691D83F38E7FBEB0FB38638DBD91A4E82545BCDF4740E"  
03233 "C9B2467F6E6BE67F8E448EB7DAEA82EC60C0DFCD981729FDE9B8DA994732ECD442874991E"  
03234 "5047E974C1B7162C792DC7745C28F93A195143F5C4E2B2255AF0E070CD9D54B28BE0ADB8A"  
03235 "1CA23F2C0996AF3B5DC54FC581F0D8DFADB3EEDA4F25E25C1E54E25E079F121C2132710F1D"  
03236 "844F9540D3C30548BB942B409216E2203F7B723F37351D86A49B2EE870CFDDE82614627"  
03237 "1B7C532E9ECE3C7E985B22AD3DFAC4E7813CC8BDF3133AE30081A1364E4203D3BF3617BA4A"  
03238 "639AFEF78C4351A50E5643AC44B3C51ECB9EF1B1073A6AEABD7ACAF57DEC2781868F535CF"  
03239 "0BE6318933659FD643D8EA8D0189AE66A0D8EC34E84E043313A8481367FAE12F1CC68494F"  
03240 "48CAB82097A50E238008433AED289C5B948D7B1A42B8074913EB531DF7196C40B499034D15"  
03241 "C81396B41A5E5C2AE85F775DB18E090BE56FCCE74C733933749C07CF695013FC804A1FFC06"  
03242 "D4DDFFEFFDD25B8EE9E0C9D0BA161E83DFD786EE54E424B8272D1E167F53F9043E23B5934"  
03243 "600C0F063BA7717FF53ACB74149DC1211FF1AEDE82F261ABF6B1C58259416F3DB7DED56F11"  
03244 "D1C148BF43F828DDD892805B080C3C68BC6316EB1C0F533B9ED6C81C48DF418FCE514DB7C5"  
03245 "8783B5F4C27AE0C3BD31E069A22CE0A95AE1C49099C839C64575D897A6AE55691B71B406B7"  
03246 "C6EFB460CDAE1126304B5A3D1BB438A9C2B7508A28DD58C42EE2A6B290E88FA5A8B0D77F27"  
03247 "0EBAB52B0CCEC8C0110D69F8314BBBFBC31CD5FB50", // PRN 41  
03248 "6E4292DC48E550CFD81C779B5DA6D2641C2F34C9E17F5D2D3B2DAC11035B1EC0F02898869D"  
03249 "DFAE322AE3AD6917A37610C5187ABCD89B839308182E7FAB6AE980157993F5371787BDFCCF"  
03250 "18D6D60C276FD44C33015356B3D39F9BED992067074532918B0A10309F40E7668C590EF"  
03251 "C0242BD003D4D33E4EE9FEA4625B98BF9873BB8D352CE0ED28B0BD2CC251F097F722A28383"  
03252 "3ECA53921476FD2EDFAF8666C198D080A4935B3DB5C2CD84918B97AB5E1200A052D495B0B5"  
03253 "BB5CAACBA0317575D8C6390BB21F3463893C931870DBF2C2C9AFECB410815C65EA33AA0A99"  
03254 "6D3217DD1B2DD826715DC57E103FF6166ECE2A3BF309552E755E7E41F7FE28D1CE1022913F"  
03255 "BEDC9EC9C45092131F54085B987D03750B35C8FA42128C3DEEF620AEB3A881CC8FAB17F1B3"  
03256 "19BEFEFCD93F65A0CA9E79A90857F6D3DBB5EE50580519645CC41E6CD359EA97E2E4D6DD90"  
03257 "CDEC944DAF3B1C461172BC1ADD53451574686FFC8FA6250EF3A82736F120F946D5B0721244"  
03258 "1087AA64BA01B99CB5648CA33040A6DAC30848AD2C3DD96160626018D5FF81D6DEA41819C6"  
03259 "FFC6C1DA8CEB6FC9C35FD109D571B800BBC4F43BC552BF5B05867B65C4582444A139BAE9BB"  
03260 "825F9B0EA9AA1540B0DE16CE5BEE092C80A32C0854DF3FEA26479A93FD5443549CF03447A4"  
03261 "B80F1A951ECD3B5DBF726B9115D053787ACFD2C6FC75804EBFBB41F2AB2C5482E7149968F2"  
03262 "94F315BDADCD25BD8469CD679D401040C5BF3B6E9E3E17D4C4855215E6ACEB2498A663F34"  
03263 "8569718E1CA6CC6981A595D5A81A7D169BB0B928B2735C4D1BB17807D7A7954E0F217427CC"  
03264 "F89251D039D1A2195F6583F76F0AD31A4096DC5036AC56EAA6EA9253DF4EFD1FDEE25E12DC"  
03265 "4C661BD1472D890010FA64ED067AA64B9DD14A36D1F8A5BF98380B273EFFCF0AE2C4E9D3AD"  
03266 "E8342569BEA96945682969CE3853B26845DBF403A3831E3D2159C80D2E97CD1DB0AA41E8A3"  
03267 "BDF1913E5E95B8C487D00FBE502C82F10FE800D2F666796724E39E1C344EACC8DDE3803D03"  
03268 "ELBE070E04B7C1F4795D585EF7A015481A94DB61155BE940819A5471FEAE67859F5C25E42"  
03269 "F2A2BC6F0AEB3E427ADA9C69BD2B0A441246E85777C76E6C9D5EC5C6D710348F1173C78DB6"  
03270 "89195D3B48518FB055BB4CD09C9340A4A1873304F1476B542D89B8D3220AB5E29839A26F1"  
03271 "2D898316470A96D271BF9AD0E42E77632351C92C1BE7666D39B6BA0B8D970C81FF3FAAA6E4"  
03272 "630A9F463125BF6C71EF61F7B28AD51F3DC63F370A69101125C9A319D58596A6C8A8418077"



03273 "F49D2BA213EA67B60FAE83770EF2DCA7991A1891AAFE767BC1BBFE388E9A0017A34757209F"  
03274 "47DFC254BBBCFA0C9DF4CD7E76FF37712F0A45A04323EB64D03F33E78FAEA8DA77B1B03CEB"  
03275 "E908B3707162BD8C5DDDA3536B1E226637E34B50B1958AC784321F99B4C781F6F940FA3F79"  
03276 "16531CA26D4C2EF4022B2158F6F05252C923506EE630C94244FFAA7F1614C11F63AE39C1E7"  
03277 "B089A1C526A98536C5453D1FF66C6ACF1E63C9B8E4629221FBAABF71E1043024106445FD7"  
03278 "E0C1FB82C20B847069C706A64ACE26624D38F2AAB85AC3D5C90320AB63375492541EBEBE367"  
03279 "86C34E7A199F97EF7C6BD4ED7C0700239ED0DDDD008A9C423984E60B2800D09B9E9ED49998"  
03280 "90136F322429BDF2F7416D39AC24705AC29A6507EDB8E62B7098DC EE4B8B3435A918B9D6F8"  
03281 "1E1187B712D0084E7F29D61675BA62840ACABFA1CAF7E8D005CE998B9330508ECEFA81372D"  
03282 "670D177C6DE1DF236E41FD37B76D899EABF6CD00B0", // PRN 42  
03283 "DC86A357A9D6AE4E3BF7D5910585996FF179F01F218A8BEA0CE0B798C277106D1AACCBBD428"  
03284 "A267C36282AB20692EF0C130313CEB5EFF80954EE294400687AFC89B72E20B413320314B09"  
03285 "A47E9C6D8F985DC3B8B8DCDCD4A67B90D989852E4E8F073D0AE845F3A6F86534BA2172CAFB"  
03286 "DA9CFD65A2EC3FFA376EDE069E3B15B7799C8192BB720BA47202B88CF71E4C820D90DB020B"  
03287 "BA968510D0C31409B070420DB16E6BA0BB5D67EA3FC3F4D5F46F59599C11166AF0A52E35E9"  
03288 "EE7EB3FED65FA5B624D2333F3CB13FEA3431207E7A9CD9E6B0BA8AAEDBF2B398D7E4A3AB7D"  
03289 "BCCBBAACF5D7A44A79C2B2D1D2E2790D10C84A1BE37D4E9E66B749A270F1C78989B0E019CF"  
03290 "858AB2F6A994A05E3A08B602F54B2E5C1F7F6629B0B775E093110A1B3FDFEECB6819B36A"  
03291 "B5B2A33BB733310E74E1EBA4160DA73A1B9125710AC3C127DF10A47DE23282AB25E28F2F45"  
03292 "80B949D5FA060D1C15E45C073CF4550A0026A0CF5FBB40CB25EB723DB6887DE2404921662B"  
03293 "C705ED718EA6490ACE13C42AF92A8412EB9843762C0B8F93E05015FC672CFE1E28ADC21EB1"  
03294 "12357F7B8730ADE449135D0C27F9CC593AB4331C8411FD1E0139C5E5D73571617A17103967"  
03295 "7624E3C217DA037A17F85F4895134F182F159616E183EEAAF92ECD8A50CEBF966F83FDE95"  
03296 "ED77F8B7E8F575B39413E1FC81B936F85FF47563152ECE2CB030F27C6E4823E3304001807"  
03297 "86CC7E034EF76DB7E56E1417D67A716E17411E1FAC556AF7104808185236262053DA7F79E7"  
03298 "61CB2EA91FD4EDD0295B8EF18112BD02D02058E960C70EFFDCE0BEA96614F36C0B664BE8DFB"  
03299 "GF72CF9143412224AEB6885808AD2BCA178456CDA74FC169117951DBACD567B3A7C928D6F2"  
03300 "CFE85A1BF6107F110089BD46C92D9D9F9DB263E468E05619F34779A32F9C19593B9765FF01"  
03301 "FDB2B6ADF4C12661C867C5D43AF7CF5E35F3DE7A603AF94FE779C8F56DEC5ADC365AAC7072"  
03302 "E226971B8DFD3E7199302C4F29D9DE4FD1C553BCF67D38136CB4F6DB7200A17ACD8F97A8"  
03303 "DEEC6B2807464042EDDF6B2C52A9C709411CEACBECDD980C93313D0AE88F123AF3FC456B9F9"  
03304 "E31077DAB6B371A49437AE46A28E596D449D36F8F0E999854DF6562C01FE5DDFF830439B17"  
03305 "44963A64F3A4F6B0936C8B64F15549FAC43682D60A6EFBDC56540E1A99359BAF03F1B32DD"  
03306 "C3AD66AB93E23AD8FF432C62BCB74E7077D629CE7B203D0DC73F1FCC4A5F40FA3D810C6F23"  
03307 "7D99193D3A4C8B6EFB0ED1A9EEE9A1F4888224037B467BD3E74ED7152BD568AB0FA9DA697E"  
03308 "07C4C1BD2A11FC12B3094DF1AB60644FEC4B5C6DD59BBC29B557896749107D0F4FE9532D15"  
03309 "F310BC31F24E536075E2B5E83398A11D1601DB530D87B74A118011CF2E319A86CD9E918966"  
03310 "25AE0912E5893A72C41CC82B45BE96148DF02F48ADA4F82345E78DD20D0159A5666F10658D6"  
03311 "A8AFDE72F3B83A62DE379503660B0C50D763046B068B730434E1429DA842D8F777D568549"  
03312 "49D0B5DAACF3B1CC6018371B2EB2F75B316809FAAED5FDE42F360B7FE1AF3397DB7CE28918"  
03313 "912085C9177F0368C8DC563CACC0425F5BA0EDE0E64B15F5B5DADE1B1435AA1475EB955733"  
03314 "920F2951C5A5F91EA72657EF19EB88FD3911DF4F2B4E6C5866199761B0BAE0B24461E00FFC"  
03315 "EF3494037ED568697D614CD95CE8EFC264C5DB140E810609BFC85131AF1D89429228A1E2F"  
03316 "0F7A17C8AE41903A5A891B1AB23859B083A6AFF4C9F66C31634F38515214180F41BEE63D11"  
03317 "AF8F7937C4B4E5F02F99D5DB9D0B42483E7C3BFEAC", // PRN 43  
03318 "8C46BE4E4DBD26FBF3C80BD9C9BF8BBF97FF93DE3C10CF5C04CA36180007A32DCA41AB460C"  
03319 "E2A0604107B5442819D5159AAC554BB2EBF6807D877F0AC3FCDD0C63EDE4A3DD19205710F0A"  
03320 "5A7B1D4071B3406868E7887A2CF58C5FD38B5D72C534CFD40B5D307C7BD5C626ABD64A07B5"  
03321 "2C7DE84189446205689744A56327DA0F4F70010A30D6CB22014FFDDD3780F5BBE687277528"  
03322 "5CA133E8FF1566C8526B66F0C6CE2D4A8D3A0183513A69B412AF28ED9D7AFA34F1FF5682B4"  
03323 "387D796BF3A78D6FC813CB34D2426304F67704624FEAA4CBEFFAF09D4E5A8A4E0E7ADABAB"  
03324 "633503193A2A61DE0BF49786D8A29032B5632036AF6267E9A5D8712D0F64088C0F5F05EE6C"  
03325 "226D6A17C6EBE74AD8412FC4677FE2C37F50CB86C79CD5FB402B3195C4AB474F4F2C3FE54"  
03326 "07D42ADC4197D752ADE718A88332A7EB0677407AC9083DF6BAC7F92D95CF0D0B5E1120316"  
03327 "145F91E3AA920CCA27FD9A32AADDDBEF0B89D9472F09DF98E961132BE9B2EA7BB77FFAD47"  
03328 "2097A25EB7D9B0682E954F24C7EC850AA24C30B08FABAA50048E6B6AFBE1C0D4A7CD07CE3"  
03329 "A08AAD56715D9C2A5B5851236AE1A495F8E598EDC3F4CCEE03AC2C3AFB9122CB8C114E09A9"  
03330 "E7CF326DC58BD4AAEDD68E7BEFF06AAAFAC5A46AEFE31F3C0B8CF3791B6244DA44F783BFD83"  
03331 "DF96F29BAC2FD07C15F7C845575F1984D375E52D3D560AD4EC430ACF108810665205FD7CBA"  
03332 "911E95A686B49242E2B4C5BFC6307A0E1584840103CB1F8A22943E4782CCC4B0DB04AFF8C"  
03333 "E6FAD12FBD844B365EE59C6302889081CED8D7528FCF3A1223BA65544483562971826E7017"  
03334 "A59D0B7E5171CDADF402FDB3B5C2E1E455A44FA8ADD4791BACDE5466F8BCBD584D09E0041"  
03335 "9FB4EFB0E01C8EC5C93D065F3DCA4DE1DA50CBFC4E47ACE0567FD39AABBAFDD35FCDCE20"  
03336 "9EE087B2FB6C474546B5EFE04C1ABF4049789DF1F2AC8DBC98497D87202AABC1287CDC2479"  
03337 "262ECEFF874DD219CE98D4CA01EBDE53EABD1B19062959DA6CA380E4DDCB7B7B0E1FE8168A"  
03338 "A4F0CFFCF761A3D989961933AF629DDA1F6CC20D27B6A181F615CF9C742B26234C1CF5E73"  
03339 "664BE0E373380BFD3C9894D85602E6EDD450BE5DB48E8C49A97D5796F046398E434D5CB5A2"  
03340 "9C25BCE7BBE8ECA805279570C229F7723CD41B9FA497645CF69BC102423153E2D37410DF8B"  
03341 "077BA65E4525D015E4CAED4EB21AA4F424849CB3B2D1756CC1766141CC8EB5D37BD68A0B20"  
03342 "7BDFE38705E16628B380DAD8109F85530435E2BFD30CCACF09EE506EA31C81291A01D52B1A"  
03343 "1BCAE835681E8FE2465BE8003A5AA8065B1148776D4DD9A6145B97BEB8243DB4CD771DD3B2"  
03344 "CA51558DF738B9D2704280A78F8D5BAF6F8053D75151358EA46936A0F1F5F1F63B2D3996A8"  
03345 "19798313DCD2C2C88F3E40B74D23C3396487A8DFF856B74230A81EDCC41FAED8C8FA8FE5DB"  
03346 "1BC92435D8AD4FB7B82D5E7089F6B1D204D7D4974AC5C6DEB932782145F3824DAAF24DFB4F"  
03347 "29F0A5AEB743A22557FBB014C1EAB0FBF3E0F9C1E9F91B86E6EE1D83152C0820F2CD724352"  
03348 "2FB85C0DEC1CE4AFF8D35669D3E07B75DDE2344CEB07F29FB98E70A56BA7A70A2FBA7C9717"  
03349 "592E7B971DE67ACAD1D37668471D74FFB60E30A027E93F40D5B08CD65265AFD53E371FF7BB2"  
03350 "2007441E1B10BE2A286A0B71DA6C7A337538266C2AF05543C0C13686DB9515D43004236BF"  
03351 "04A117FE8B310082F6DA9C9CDF0A9B959A1D8193A48CD3ABF8D15DB1E981DFDC9CABFBC61"  
03352 "68A8387486F1E14F9119C6B6C82E996694ACE70AD4", // PRN 44  
03353 "DF0B035CF6B45E44E8326F7FB58EBA2CBCB8C319E312F7898048794B18C591AFE9BF56845A"  
03354 "B84AFB1A3A72CD1D13680C786CED126F0806EB9ED73256ABED5A881665FB95A2B53D97AF512"  
03355 "DDBAE79121E071F4C79761C1EB98BA53DA1CEF51BFABCBDE43F223BC12ABB3E301B4A2C7C1"  
03356 "47813A1818B8FDE9ADFCB7F30D0F5A88CCBBB374734593C0DBFABE04182D64DC13C2CCC7AE"  
03357 "5C49FCBA35C0889AC9920911658AE7EBE24096F831FD9F546A32EFE1E0FCB39AFBC7A0B2"  
03358 "0566CBE4AC58A6FD224685800234693FFCEE697EA88197468A826C95512DADCAE58DB45B"  
03359 "4F3C6CE862C73F5CF151A044471CE862A3A1C46B71252FE40E740FC4627C9803B6996F99DB"

03360 "2AC44B4A84041DB5D1510D14A00A3F9B53F1C8E54A7DFEC5DE283D1DA9837B2C4B117AE165"  
03361 "BA63EB2F201D84274C0F3158105F6583D440B0D05E28CE96821DF99863E00DA6624A8EDA6"  
03362 "1F7CD223FB9A260E3A03BA2F8263082442395F67616DCD8BA7BB294A0642CF1FA30791BB8"  
03363 "CD5A0C2159F513F99FBFF776C0A57827F91D5AD076E0144B7D4A109A01C6303A35536D61DC"  
03364 "9E486E55E437E3DB7F94B26A7087EB21AB334F565624C1079EB1EB4741C5E0D8168327FC3A"  
03365 "D3800D8E2B93262DA4EAD217F6E2409F7218B0AC2636655E104E4AC5B791245391596124"  
03366 "E9532C9ED54155C0C9DAE8F79B28BE00229F3C4763BF9F8F075B728F8A123FBFA4F3A61B86"  
03367 "0367EC6D797F8546D6ECB449F638B84E42C67D5D1112AEDE51DD95A0EF3C575D8523586A19"  
03368 "838E6B85314B6ADB066405B7F6881148D6F2A6368DDE833F8AD5983BFCDF219FD1E8455EBE8"  
03369 "FB1075ECDFCF4A7C8279B40C227192043BDF84071A36159EB51CADD2AB9229984F186B1D6"  
03370 "97E58D963532F2CAE870C8E80CB1A985DD277DFC086AB82D9AA266CFAB2784CF1A1B3AF69E"  
03371 "09BDA9F0E493EF7B9038AF621C41C8F083CA6012BC74F02F92169CD7F779F13279D0EF7840"  
03372 "A9D2FB2D0254744325030C571BEEB83500733F04F88626B7E0DD91CABECC354438980770DD"  
03373 "DAD7B76E48A050EFADA5780CB2D88B4B5167754C464CAF45BAC5B2E33221837AF9777AC4F1"  
03374 "838E6B85314B6ADB066405B7F6881148D6F2A6368DDE833F8AD5983BFCDF219FD1E8455EBE8"  
03375 "0DAD449AD0A1C8DE947636C7A9A9049666C8470B1730FFA57330536C75B9BB13E0F6B081AB"  
03376 "45B2860BC2464FA8A30EC40F27871C492EF5E1654BF66692B8D6F17407506675E8EA8318F0"  
03377 "9EF95DC687329D828D5219AEFE922545502BBAC45A43AC9D8CEA133F92E97136CEE97F30"  
03378 "C773261933CC455F3F8F1FA45FA6AC803754625455E3BE824C5640337B06B06F258E687D77"  
03379 "56654FCD6226A1B2FAACB4CE6A92372BD41DC5F6DD0E3C11506629FDE1A730EC637253208E"  
03380 "AF7E2DA5E956CA14C13720C2277E1A3F12C46907215BF707B5C72EA4BB6E4EA74C9AFFDE9"  
03381 "2C1CF5C8B7C77DF3C876D18338D38DB7BAE231D51F4D21ABBD09A4670ABF92E0EA885EDBAF"  
03382 "58BCE7BECCD258FC2D43C9D1788FD4123402C075AD5668E8618D20AB729D43400BA877A98"  
03383 "20F5BCFB5B497482F9792B897C35E8B9D1A616AA5F1D730F0B50E5B48FC20BF2FA826F1"  
03384 "E57E725D96C95CBD8B277E3224C4C2274A01A312B738477BDD172DC577FE8823BF3C171629"  
03385 "10557D4738D8151AFE31940E3382C27E7A1EBDB3A9207287DF6C5793D0C747B13BDBB205BE"  
03386 "042F723782C959B73355D6B30FCDA9AD1E200205D7BDB747F3AF27F0745A4B054CEDE995"  
03387 "AE60F7AF99CFD8CFE91B6D4FC9E7CD5482618760B0", // PRN 45  
03388 "E9A5B2A753003C06D6D576D66B52988839CDC8A608EDC4DDFFEBE0BC1E1801BCF8CBCE9494"  
03389 "E2E2512229A5E6EC547F4962B217D02F859477B54F660E217B5C89BB1975B6E67E0649592E"  
03390 "05B4D1110C9DFA34ECD8772A515896DED72BEB9DA4A1AA2FCDC080875DC179E40A78ECB253"  
03391 "8F258EF9FA5B9A6F4D9D77E9D6F0EDBE75440BC6FD03576A3AB2A323EC980999BA4129E956"  
03392 "31ABED2451E9283971659E6562A45579B22A5CC67CDB4AC6CB8747672631B3E8C41590E085"  
03393 "A312BCA3BA2CD9679C4E21FFD66F97AF011A77A2E5885FF6D42D6276B6EDE10CB29CC5EB4D"  
03394 "D8A7110F0F9702EF12205CE35803375DBB421D167D65F508CBF3E1AFC6F5FA8D330143F8E"  
03395 "45682AAB866E750AB68AD4793CF8767EB4DE37C9166D2FA419F0FD3EA0B9F5549F657E9519"  
03396 "21AC38FBE05D9A2E2EE2BF5D7520C1D8389AD34C7ABED6C8182D2801E7435A44ED80ED3A6"  
03397 "4DC1E9C66F105350FAC87C36505E7211B873659D7E2817ADC8E74DFCF62B3508A0F1BAB6F0"  
03398 "F1654903CE1D853CAFBBF731406F9596271860E6986DC917C4796D57590560233F05C75FED"  
03399 "1034783A3C319377372B2DF2D27DB655123B30A8F6DF9B329CB8AA2A9F06CA13E10E1C9DCC"  
03400 "9FA9725DA1E47F0917085A1F6710DD804B380CD6354D980A3C64265F5C2C3A1CC3E6A3DCD9"  
03401 "5F62D4DA6CCD48F83500F74A0275469DC0D921B648F74B7F612552350049FDF94141F6C493"  
03402 "75F17FD512AA0982B5DAE9F9C332EFB2007E367882439BD6F416FAF26EBEAB6F1FF6FDF0E7"  
03403 "DC7B2DCC70FCEE25CBF3C22FEDA4B0B0A272F049379AFF4002293232A98F69E83121269AF7"  
03404 "1D06DBA5671169DF46B7BE8D5AF09920BA89998A10C30A8E101AFBA9C84D0B5C73AA622D99"  
03405 "6E030FD6E21D8984723C38CC43F23D1F242176B3E6C2797A77A99B15373AC0127DCF67ACBB"  
03406 "87EC231912E1D6DAF08B93934CEABA17BB70F599681177C4C870B27E5604026D15C0DC1DC"  
03407 "38B2CC733558F744AA0E981A0F8C8E569716B7C99040D0F317A05B70CDEDB32601E360CFB"  
03408 "A5078B133C4B1BEB4C405A66A425355EBBF93B5E53D97603B100FB3B84BD5D5404953C8D1"  
03409 "4AC969E3CE880898ACB097FF0849CBDA28DFDEC014F9F2969297472B3C64E12A8B8D039D25"  
03410 "D9BDBAEBEC99CF89C2177DE8B2A28C1A76E89A975099AAB1B41170AE54352F990B7EC7C60F"  
03411 "64315D35E4CE58D8F0C71BDBC0AF217EAE5B3C330B42603B4F544E22486057656370534066"  
03412 "23DD803D7BFFB739C55A022632873D314722897F65D841BECFF55BE5D08E242FB928DCFD1FD"  
03413 "1DDB599B28AD48AEFE2805E9CA6DC8D230ED76F920E7EC31A9E7899A328CF51ADB640A8"  
03414 "19E8CEDE8CD3359E94B4FDB0EA6963525576BC78BFBF24CF0AD8D2109F9F62EEB0B97CFECD7"  
03415 "947CF23B750BDCF1F705398B75A4FA80CE403A6894145F7AD4B875CF1B7EBFDF5603310A8"  
03416 "A68922074586D38824BE987A1B7A7EF61EA41AF880F0786D00E3C87C181C8494033EDD108"  
03417 "FA97A72C10DF99BB20D7D214CE6E87428FF896CF30740E7B968C40B59D4759CEF0F265DC0C"  
03418 "DEE77395B2584DAFBD202E9B7D4E88BF1B7846AC11AD416A9F4C109A73BCCE420C8D56C78A"  
03419 "7D3BC3140749E2EC85AC4E1C3C202A0F2249038540E8451677B396A0A2A493BD30530E81E7"  
03420 "129229A3F1E8CCC556F701F8E4C114D5E036A65E83C5198CCA79E8D2D1B4619B94C7F65B79"  
03421 "0DFF724F4961EB21BFE5D2147B39148677EC43D12B0A03D8F1D6C8959458B51202A19A35F2"  
03422 "F99D70A88D7ED3A27162E2C9452E5E25310CCE0B8", // PRN 46  
03423 "B0E553DEDACE943FC243EAEA373E685435F5CBFD771B220AC01CF6EDD0C094E0D1C86A6A02"  
03424 "4F4AE41255001E89D749CE52BC634A03A93BD0F96EC0FA963C84E1E7D097DA8B747EF4"  
03425 "2C5FBAACC675FCC0225694948C72F7D71A62DF676D3B8DB0175319F439DAB5D65E84EFF624"  
03426 "16B1A1F6D33078BDBEF2CB046F3903FE7B30BA88AE983585380E146132F61F09E327123A94"  
03427 "495B81917E9DE29484E9A2452FE9429549517175E292767A2E78A2ECA42B9FD03F0D9481A2"  
03428 "F2DDA62D3ABBD6F4E5B44ADD9DE7DA244553FEE82F7717410D99A54EFFC1D1B186FD66D844"  
03429 "9E08D4109BCEB024A685797CA4CE98ADC2CCE2EB207919EFC5F8615914D41B6538BC82E183"  
03430 "7A2ACAD541BDC0D316270359E6102D8B4CEBB91DDE907150A9A6A00A72953B3C4FDC3AB2B"  
03431 "27ED005A034E7B5FE10072A4D79C1994947A5E2098E043FBA08D2D54011F73040FF023687E"  
03432 "28989737C87C6D997FE78C9E3BF7DF6810A57C3362EFE9B741A5C12FDB5EF8E4EE7CB805C3"  
03433 "84025BDDC3B667EAC69807B83377AFE7DEAC5F9DAAD6382C00F556B50F94C9C3F3F9862AF9C"  
03434 "687CA48C762A71D9E5F95AFFA8CC2ED974E06A65B7AD4D9EC48B6D79B9D4267432915D64FC"  
03435 "C2141CC7155720D1951A300BF9CADC89E5DC8853B8055C7602D182CD676492CD185363A52B"  
03436 "726FCA90351E8504F7D53E7D19B132E55918719958D62FEF4BE6E3945EEA1FEDE18B3818CD"  
03437 "F0C9C13E6E39807D890A268579DC47286999E7EE6056AAE24F800652F3A6904B508946A364"  
03438 "74E08136FB57AA9401C29F1123E3849CB4156E6322F4A3A71404DB57F7338393E313D898B4"  
03439 "2B244EB4FBA465A4D6AB1554BBB61DFCD4EF5235D720D1083305074A5F8087C464758D9BDB"  
03440 "33CDC4C624320D4B883BB762CBDB5D92080EC3E8F5997AAD2CB54421591DC4A24778CDBDF6"  
03441 "CD39D43ED7C052C3D571A5880B3E1F0EE5629F558F9AFD94B230BE3EC7586DE128849F5FEE"  
03442 "E7A83ED21F9C8DCEC8BD07A2875EC0B3C94C62ECE93ACB2E3947CA74D9944CF424F5072A26"  
03443 "CD57E1F065DD256FF4BF6262BB40BD5F92846929003A31902ECBA57BAECB00EBB01EB90137"  
03444 "4CEB789094831A4F082D288019E40800E67FA1F8E7A7DA9277D2D0B807AFE926D0A78A5BCB"  
03445 "CCCCFF64AD4212762782EDB02FAFB1F40AF5D43C4028207ED86EEB0DB39D45F08188B6C79"  
03446 "E3A21ADF5109E331017739A61E102A089236C42A8BC221EB2194018697EE4B5ADE114DC36C"

03447 "001D5D21CC2678BB35FBD1B1249D0FB2B46BBEDCA7E77F8B12DA2141C8DE43AE6E0D2B7BE4"  
03448 "160DE9AE81177CBAC44D1FBA1FE14A19597EDE5D5A5ACF750A7E8DB72764958BA28DA59CF"  
03449 "D2BDFB89FAB20D1DE072C95526E1B3B9171616B0C73D0C853A9A7804ADDB3049190598099A"  
03450 "45C3AC32695F0E6E355CD4565783DF2845609828BB785FA38F727506CCEC74920A43EB37C0"  
03451 "BB5695C9BF02F14B8AFDC0B591A06DCB08C446C9492028BD91A2E793B8185FC14A4816BA3"  
03452 "B9E476C1F3ED607B783D4E8DE875BCB7972A2815D8EDB08D38F3D10C1EA38AC52458DFC22"  
03453 "BA2226763AD3954E8698831D54F8DA6E8D78D2428F3AE407A6C42EB4BD87DBD3D10A0B95E9"  
03454 "DF0F8BE9A9B26E067A1CBABE1F2694B6E3E16FEFF64D53CC599AE6A1D640A2CE921B67C182"  
03455 "009DDBB2A8D74C10DD35B518D0FF7F01354D8C08A5D3FE18204BABE389BF2BA391AFD134C53"  
03456 "A5212DCD30640163BF12155855848A1994AD1F9B017C4456AFD816DA9A797B7C68107E9829"  
03457 "E9277F338CCD29CDF526FE3F2FDD8BEBC40CD6C1C", // PRN 47  
03458 "07DBAC0D7EE0E218F5C403D29D1CF92F57C3C878B36FAFB77FC90EFD7C7D55BD0F6F9143A"  
03459 "3FB411962F0C116EEB29065283075BB555CCE1086B8BF5D9F632976D6C764F21E4F7097E04"  
03460 "290848CF54DD98EDB896405AFAD2329DF1852647BCDE28AE8B5155462CF6D85266B9A45737"  
03461 "0007223AA335E790A2796DDEA6AD2EBDB79D48970824484CF4AF8920FF7C955BE43276A20"  
03462 "33D3CBD82108D5D2799186347CA33D9C2EE605FD7E04D456ACF1A2511428FABF170EB3DFE9"  
03463 "DFEA5AA25A26768396B71378D5A16069DF73628705F9004958EA93C8BD5E595D6B69D3045"  
03464 "A8DD864D29241D269C2CEFO6F27999CE68BE86734C96CD8C8F030AF8FF81D671354A179336"  
03465 "FFA0400AAE9B659E2435FE17B10286365678A33704A2414D08C9D07AF9BCE755E4057893C7"  
03466 "71AA76477607F5D73023841F39AFB26A57A8D6AC821AA4FD013C838E95441BBCAA3026BBE"  
03467 "ED561E16FD7DF9F441ADA96C4BC0FD7EA1E4AB9FB97CF364F83E925430BEC3DFE327112CD8"  
03468 "EA8863E657EACD4910FE726720D3F67E35D8C9605B19FD4D0A2443F92052A54704E15E971B"  
03469 "51C50E4BE49D1D9B0DC1BEEFC3DFF1B865E757FF0B583E23848D5FE49245A94E5426AAACBD"  
03470 "D42979FA806FE21B54425937BD6C49FBAFD86292182F1BA565D3006AFB5F0CBA343163AAAF"  
03471 "45D586885A105C89E2284E1F4487BA8E08F673A64F49711E14C9A6CA1A431FA812F3CAB663"  
03472 "030307A41CD97AD73AFF004719E35F3300D1C4D94A0637C65A3885485C3EB6D3CBA1DEEE93"  
03473 "1114C39CF2D094E9FB0D9B3AA7826062DAFBBD41D30EF695816966BD0F416CED7E1951AE19"  
03474 "45C2DBBA6F7E1B277669A33E40DB63C07E33F49AC66D68A6E35C5142AF355D6CF23302D12"  
03475 "88F2EA8BBFFA1C0946D9F9391695D752A0E10883D010E93C19FD3A2F7F1B114489057E18A"  
03476 "B1BD1382FC2A3B5FA4989DEFBB888C4ECDC84506A5CCADA0964FF8D804A9B5D04ACE81F21"  
03477 "02A7DE0E6F85335E5788990374C1790982861E04836580A0BE12DE3B1E62105103BBC05FDA"  
03478 "B5F0307EAF3D65B75FF39584E9DF97C95C050EA0646C48F6ED5B1C949A2145295C054B23BD"  
03479 "55185D5B3671011704E7432630AA19AAE7BEB3E694A8AEA0995CCB45A11249A9ED0326705D"  
03480 "81ED456A5656E75D9997EDF60B16AA9A2D6E96067D0D3FE291C3B8B25E5C53B82863908915"  
03481 "2AACD3DE1C3F177BED52DB5A71F9B89162F2026626DF62F45D9CE2D085E77616BD1987CBE3"  
03482 "FBEA54786B8C15D7C2E5C65599CA47C33E7A0BD5098799C8C3B3A0B56D3F5ECC69C0B490E3"  
03483 "662201CC9E1097FEED0817891C3DB24E882335D5F3FB647ABADB8F250CA06591FD10D4F66"  
03484 "591EF9145541447B49D5B6E6334E7EE0AFB414AEC813C57C34D71B8FC70E3F2DA7B3DD964F"  
03485 "526D662A6FC32E32444A6D51AC4AC6AF073A94F877007C10C7E4B4039DC6CD1C2BF12A6C"  
03486 "A9121DD14B6681E84675E6512B353E638491CE256346E36B0428116DA5530691276954F0B"  
03487 "B164090143C8DFBE1B88348602E163BDE9D3C95583054A7FAA921D266F5BC309E627CFC2C5"  
03488 "A38BF8AA1877103BBD2768E267ACA5CD148FD49F1F29F5733ED4E0DD152DE3C0062D36F716"  
03489 "51EBF72F56D6BFCB44486FD7BB93A42EB3C29221A807E136A36DA79D87622E3EF90A7EACD"  
03490 "1F3601EAA1F9CC3AD7974A876E067E21F7E38FE09738EA4E4ACFFFF03F465BDBF99A95AC32"  
03491 "653BF6328752B89C07A09B1E912E6B51898AEB2E6C39EDD54AF6EEB77D68948ED1F0CE45D5"  
03492 "57D30ACF77CA25C3AAFB902761CE559AD5D9F3BD50", // PRN 48  
03493 "4778E4E7E70695F1546A20995B638803A31D9030B5C9292607668EA2C86D02453F513FD4B3"  
03494 "8CEDEBBF7FAD6E8BD2B0755C2BC5A8BC6B2D978957E71A3FA1A186ED8048147C86A8811CB"  
03495 "613BF3BDCFC49449137FCDCB65640CE18285D642EB2C2B3C233578DCEC221624C67CEEBDC0"  
03496 "F927F3DF062B7B92A69583F187EEED690D8C79FC60188F4AAD5D5C6DAAA10D0ED9232CA69"  
03497 "5DD215C20092269562BB35AAFF60F81BB6B187F64E9D0939C6E4AC6C0C8BEC5D71BAB85E"  
03498 "AB62C634F80AF40A9BD6C134E8353E05BAA97D0547A31F64082EDE4254EE0F856B7ABFE8E1"  
03499 "601DD965AE61853E65B0BAE45B0DC9859356874599C4903E65AE0737B8365ABAD24B743944"  
03500 "9B382CE3A40826F2654F9415427C168BC0C30A16B28E289EED8870436FAE32C57A8781967171"  
03501 "690CAF0C305FF7848A59DA8A8D499292EC903AFAC56CB98848B20B5FB6B2A052999BD8783C"  
03502 "65DA048C8A7B5845EF773D37237858BEA4373B5A77F5D2C1F9AAFF6A7A9B31AFD405079D99"  
03503 "3D2A4B74D43974D2438599871BE8FB259CF1D20ED12116D9568EE0805E5B4150ADE95D4806"  
03504 "617FF8319B1018EA256867F1FAA7CAC159DFD0C6B2F2F9B133E11D0CAA1B92547F9C5B5DBA"  
03505 "E9B3C50FF3D2A21D66DF72A632562F4C58F02384254A26305D11CAABA33630FE92B511E0AB"  
03506 "AF1425621DC3388B608FD224DAF096EC08F9F791F63569711A102BDFE7337DF5247BA1E78D"  
03507 "013C103B422825F87FA144EE19A6AC4BF27F6A2A80C0ADB4BAC22FFBA91874B3F4B9C587DB"  
03508 "3223E9CC6FF4815733C31C325EBE9B4A42B64A24274C5C9C06571D0C146DD7B6BB9DAFC8F7"  
03509 "96E02A4034A4C84ACB835B4CAFA5FECA70D59F420F5D81E5674AF08F205202D759F3D1DD4"  
03510 "5F355EE679F3426CF7682BC23F055F297D57C138EF3EA4A7D1EC78E19D5FEAC9061A974965"  
03511 "7AD6360B6FBB8017DA96BBE3CE3E83A65790F82CEE9B76F1E30863FC6972FCEDB23FA8378F"  
03512 "19A23BC858CB80A163BDE11827D5521F5C6D1E18C855CD0A7AC4EA42BFCB1A29C027A1410"  
03513 "C9185A2784B3E2E24BBC5272B37AB0879B70F2D8C3BDC63C0D1BC43FD6DA37DC32FDDDED3C"  
03514 "3291390D471C2876D0877CC7568F5B1159C39FF678D65026B4793996B79EF80A3128C719C"  
03515 "93F33665226409D3826A4B4BAB99973E8145004B5CB736198E05C249B19F7EB1C43215E4D4"  
03516 "E1E96AD1A1DA6B3EBB505E3FF61E98EA79176EAF281663B96B68A83CA0058A79548B1E3B27"  
03517 "3DA6F9655289369DF21B3D45CA48624CBF472842EFE8EC06F69DECC427328BA713E88C51AF2"  
03518 "D32BAFAC785346976CFBC268015FB58317C832A5DC130665BD30BDB31C0C74E4DC77E252D2"  
03519 "812EDF4A9DCCAE0759FC945A7C4FDB1978A2BD256B1158D3C79E7DC835718C38087E842EED"  
03520 "50736904A3053CAE27C10784A32E7819A2A541EFD9CA641364D839728A656B27D532766836"  
03521 "A5C2D1A968D5C97E8770B661337B280B9E8C6AF012BC76C2D43222DA5273D4D38725387B16"  
03522 "3CEACBA72F75B6E6B4E7D552B774656FFDF7615D037A8AC8F72DD845DA21C8E372F8C5BC3C"  
03523 "42CBA3A189006926AD11C6539C0911AF56B13386B6F729519A5195D6AAC61CF35252956E1"  
03524 "5736673C4E19F3EC7A2D2E5743AAED65D65ED6ACFEFCB8E5B53320B7804A6187703E53E210"  
03525 "59FAD6278C26FF96EB4F44B1EE2B03AA37C8C0F67C822A62C7C2C8F4F35D51BA0814A70F09"  
03526 "2A513C358C9EF6E946683FAA70CC88360F6A8AEE76E212627C4F110FCFE6D24A7B118658F2"  
03527 "A366678ADE49C0ED529677368E3CEC7A65040F03A4", // PRN 49  
03528 "37AF4FFE6FC0F844F8BE37B431A60B3FD62FFD1E73B410D218F58B791E84A90E0AACB7F595"  
03529 "6987001AF46748C6C573D3D9857FB1D18F916DF21863CAFFFD853ABE3AE024DAF4B5443C97"  
03530 "F24A4FFA884D085C118ECFDD2C66A0DA98903C7CCA662790CEF0908F67CCA1EDFEE31D71D4"  
03531 "7889AD299B1FE272EE0B04D8B21CE6C00F05961D69DB9F0F1FEDEF4F6065B86A59D92E697C"  
03532 "8E3EF1B112C844EF77C3EDD0648E02268DE35A4951F3BE26D9E2D7CE455D92292E48987830"  
03533 "170CDF2096D5B3C136A0EDD548E73AAD4A6A79523600504CF34AFB9EB85F03E79336404899"

```
03534    "9AF55BF3A6B347C530018F546678784860106535D2C73B44C4BAD39AE74426E8CA0AE666E7"
03535    "FA8C51AF6B6DCEFC4F5584FCD29960BB9298353052570364A1936E9E0118D7D6C0BA3E50AF"
03536    "8B3ED56CB535F454135ED900EF59114E10AD6E53B908F73B5ADFDDBAB01C95CBFA739B7F8A"
03537    "199714C26ADCCBFA46B96D4D4C79050408B65357241AD7BF55349F085B8C040F65287DD442"
03538    "CAD5A2A970959ED5B00AAC659BA6AF7CE5C26EBE51E4A471CAE001D74B13C71DF072E978F9"
03539    "647D75123092FA8556604BE736B6D03DF1C8656746E0906FE462375759EDF12793C475446B8"
03540    "CE6E2EFF44B164C2E7477AA22F50C348A2BDD65E6B1F021B5216897ADE7FBC70A0767E514"
03541    "9234E84FD7AF5FB9F22074129C8E4E15F935F451E36AA17F50B1BB698C302ECCFEF8B4CC22"
03542    "9ED8937CD4BC5F8876938CBAA810BCAB0751EE66DA1AA2857B54534EFA4D03D5E1F681D15A"
03543    "179BEEB40630868A2070D0C659E7EA3FD26308F5CD6D75B97803C017E18D4673F6B453ED69"
03544    "4B32083260C43C40EF81AD8CA230426AFB9020D8942A2A23D4022ED50979CF8537680D96B6"
03545    "CC165F9ADBBE540CAB59E2126FD5CE6852B8613C346BEDD11A46CCD6A04134FA298692C7CC"
03546    "B34457B8D0EA863C37383C4E990AD3918DC532FAF8B15D8F610A85B03C2CC4650FB1BB8C05"
03547    "C1C0A0AD75D2E80358D2376BD4AECDA2172A3250D59884A5728A74876F736B4D24CCD5EF1F"
03548    "833A6A4C6F2FF8648D073823DED852109B67BC9BD487E2BA991D958B2FFE34B403F355B7F9"
03549    "35CF8A63037C8C6100459863F58419847B389D92DDA586AC3A3539148E81C810FDB6AC7EE0"
03550    "E731DDBCCB9A3DC646C8EC2FD37C77D365B5E257BA519D07C4D87342117CEE0687D686090"
03551    "FD1F002274149E3EA3A2A91B0227170A6CBCFAB70E3F74F3F8BA317A5642E8846110FF20F"
03552    "2E0DFFE5818C24432BB94A2C4AFF87A7C72BA3E206B300ED5E5AA7624DD7BF84A52E4B7B5"
03553    "6B2353F2052883EBD1E9CB7AA55EA24EF402881E13381C0C76B35719D3D8EFED5C8B823F77"
03554    "A1A203E3DFB83E51BBF0FEE41CD5B975DAC38AF8FCC0527BE1828426F11D3ED5676897D756"
03555    "EB3FE2C34A51F6FB9136AC656893826D6881BB5A6250302934393CDAF906B2F03BD8865CF5"
03556    "BE9DCF78DFCDEC0711B6712ADFD6A2A71FAB1D6FE40CFF4BB1267D169FAEDBCF21F1379B5"
03557    "8CC76731093BF233951FEDCA86A2BD505F9E10ACF169943F20846A341D85297DB02288BC9E"
03558    "FAA7B43691DD53CBFC170B4FD7D3E11FF76C30B922C8AA205A376F94D61066749C2C7387CE"
03559    "7EC5C6952AB7C9F12E7C11489AD879F8ED43015D1C63FEDE9EB223D52D79BE1E2242F9B92A"
03560    "5C64C8B6D5C36498A9A8F862473C081A58004ADE1B4EBD4C4C0AA5978074FF22AD2CE1A70E"
03561    "2E1945F9B14E3D537E07C930864EB4C09DDA24B74264FA45903008E42059DEFE5CA0C04D45"
03562    "A44B716EE0BE551C47CE7D23182370FB896B9AB080"    // PRN 50
03563 };
03564
03565
03566 // Galileo E5b-Q secondary codes
03567 constexpr size_t GALILEO_E5B_Q_SECONDARY_CODE_STR_LENGTH = 100;
03568 constexpr char GALILEO_E5B_Q_SECONDARY_CODE[GALILEO_E5B_NUMBER_OF_CODES][101] = {
03569
03570    "1100111111110010001010011101110001111000110000100100110101001001001111110101011100101110010010100",
03571
03572    "11111100001100010111110010011010100110111110001100011000000011100010110101110010101101010110011",
03573
03574    "10100010111010101101011101001011011011110011000011001101110010000010100001110010011111001000111001",
03575
03576    "01110010111100101011000100011000000011110100110101110000000001011001011100001001101111100110010111",
03577
03578    "000100111110001110101110100100111011100010100100011100100011101000010010101001100000010",
03579
03580    "01110111110000000100001000000010101110010001101100100010110001101010011010010111011010001110",
03581
03582    "111111010111100010110010010101110101111000110100110101011000100000011101000010111011001010011100",
03583
03584    "0000101101001001010000000110111110011110001100110111101101101100010100010100101001110111",
03585
03586    "110111010000111000110010000110000010111101000111010101001000010110101010111011101110111100",
03587
03588    "01000011110111010101001000011101010011011000100100000111100111100110010000111111",
03589
03590    "00000011011001101010101100110011111100000001011001111011011011110101001011110011011000011000",
03591
03592    "10011001110011001011101111110101011000100100100001011001011111000110001110000110111010101001",
03593
03594    "1010001101000110011010010010001110011101111011110100010100011101100000011111001110110100100010",
03595
03596    "00011010010100100111000111110010001010010111110011010100011010110110111001111001111110000",
03597
03598    "001100100000010010100110101110110010001101101001001110100011010001101000110000101011100101100000",
03599
03600    "001100101111100000111010101110101000011101101011001100111110111000011000101000111001011011",
03601
03602    "0011100001110001111101100001101100010011101101110111010101001111010110110001110011000001",
03603
03604    "01101010001111001011110111111100101101011001001101000101111100000001001110111001110001001001",
03605
03606    "001010111100110100001001100010001001101000011101011111110000100001100111110010111000111011",
03607
03608    "0011110010010010100011001111111010011010100001010000000101100110010100001011011110001100",
03609
03610    "011001011000111000110011011011011100111110110011000000010011111000011011010101001010010100000001",
03611
03612    "111011010100001010000100111100110100010100010111000011001111011101100100110000100110000100",
03613
03614    "00101001111011001100111010010001000011011000001100101001010111000101011100011101111010111010001",
03615
03616    "0100010101101100110011110111111111010010011010100111101010100001110100001110100111000010001111010",
03617
03618    "1111101101110101011111001100100111100001100011001011110000000010101111100011011100001001011001010",
03619
03620    "01010110100001100010001010011010100011011001100000100010010111100010000100110101111000111111100",
```

```

03595 "01110000000010100010110100110010010111101010000101001100010010110111101101111010101000001100111000",
03596 "000100100001000010100011001100001011010011010011101101010000011110110000101010011001011101000111111",
03597 "01000011100011101110010000010000101111010010111101111011011110011011101100001010101010010110111010",
03598 "010010111001011101100100110011000100010101011010111000011111011000011110111110110100100001100101011",
03599 "10111111000111110100010111111011101101010001101011001010010101100111100111100010011001100100000000110",
03600 "11011010010000100101010001000000111111010001110110111000101100000100011011100011101100000110100100",
03601 "111011100010110010000000010101111010011111000001011010011001100110111110100011001111111011010001",
03602 "00101100100010111101011111011000001100101011100011000011101111101010010110001001000011010010010001",
03603 "001110010001111001001011101101101011110001000011110100110000001010100001100110111011100101011010",
03604 "0011100110011111011100101010011110101011011011010000101100100100001100001111101100111101111110000",
03605 "100100110000001100011111101111010100101100010001111100010001110100000111001010100010010011100001100",
03606 "101110101000000001100001010001100010110110000111001101110000010111101001011110101011100101100110111",
03607 "110100100100000110001000111110001000010101000100111010110001001000011110100101100011111110100110100",
03608 "1101010111110110101010001011101100001000000111011000111100111000001110000010010110100100110111001010",
03609 "0000111110100100101000100000010111110000110101110110000010001000110100001000111010101111001001100111",
03610 "0010011100101110100100001001111110101110101111000110010100100001010111100010011000111110001001011000",
03611 "0011001101110000111100110101101001100111010010010001010000010100001000110010111111100100000010110",
03612 "0101010011101111100101100001000101101101010010100000110010001101101100001110000001110001000000011111",
03613 "110111100011010001111100011110110010011111110101101110001001000111011110001100000100110101000101011",
03614 "0000000110110001011011101100101001101111110000110100001110101110000010001100010110111000100101000100",
03615 "000110000101010011011011101101000011010100000001110111010010100110110001111110001110110100011101101",
03616 "001010001110010000001100011010000100110010000111001101110000110011010000010110010111111101010110100",
03617 "01011110010000101100000110010111000101110000100100110011010100111011110010101011110100000000110011",
03618 "0110010000110001000010111010110110001110101101011011001101101110001110000110010001101010111100000001"};
03619
03620
03621 /** \} */
03622 /** \} */
03623 #endif // GNSS_SDR_GALILEO_E5B_H

```

## 11.705 Galileo\_E6.h File Reference

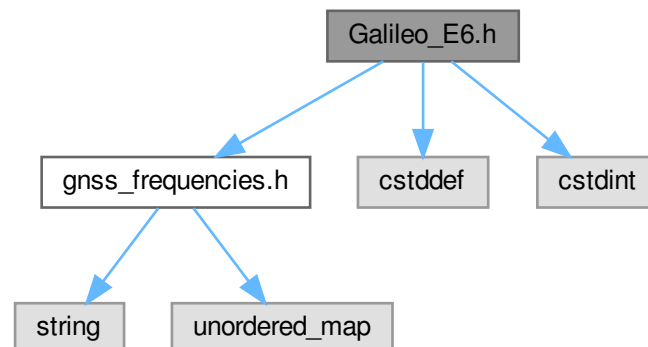
Defines system parameters for Galileo E6 B/C signal, as published at: European Union, E6-B/C Codes Technical Note, Issue 1, January 2019.

```

#include "gnss_frequencies.h"
#include <cstdint>
#include <cstdint>

```

Include dependency graph for Galileo\_E6.h:



## Variables

- constexpr double `GALILEO_E6_FREQ_HZ` = `FREQ6`  
*Galileo E6 carrier frequency [Hz].*
- constexpr double `GALILEO_E6_B_CODE_CHIP_RATE_CPS` = 5.115e6  
*Galileo E6 B code rate [chips/s].*
- constexpr double `GALILEO_E6_C_CODE_CHIP_RATE_CPS` = 5.115e6  
*Galileo E6 C code rate [chips/s].*
- constexpr double `GALILEO_E6_CODE_PERIOD_S` = 0.001  
*Galileo E6 code period [s].*
- constexpr double `GALILEO_E6_B_CODE_LENGTH_CHIPS` = 5115.0  
*Galileo E6 B code length [chips].*
- constexpr double `GALILEO_E6_C_CODE_LENGTH_CHIPS` = 5115.0  
*Galileo E6 C code length [chips].*
- constexpr double `GALILEO_E6_C_SECONDARY_CODE_LENGTH_CHIPS` = 100.0  
*Galileo E6 C secondary code length [chips].*
- constexpr uint32\_t `GALILEO_E6_CODE_PERIOD_MS` = 1  
*Galileo E& B/C code period [ms].*
- constexpr int32\_t `GALILEO_E6_NUMBER_OF_CODES` = 50
- constexpr uint32\_t `GALILEO_E6_OPT_ACQ_FS_SPS` = 10000000
- constexpr size\_t `GALILEO_E6_B_PRIMARY_CODE_STR_LENGTH` = 1279
- constexpr char `GALILEO_E6_B_PRIMARY_CODE` [`GALILEO_E6_NUMBER_OF_CODES`][1280]
- constexpr size\_t `GALILEO_E6_C_PRIMARY_CODE_STR_LENGTH` = 1279
- constexpr char `GALILEO_E6_C_PRIMARY_CODE` [`GALILEO_E6_NUMBER_OF_CODES`][1280]
- constexpr size\_t `GALILEO_E6_C_SECONDARY_CODE_STR_LENGTH` = 25
- constexpr char `GALILEO_E6_C_SECONDARY_CODE` [`GALILEO_E6_NUMBER_OF_CODES`][26]

## 11.705.1 Detailed Description

Defines system parameters for Galileo E6 B/C signal, as published at: European Union, E6-B/C Codes Technical Note, Issue 1, January 2019.



## Author

Carles Fernandez-Prades, 2020. [cfernandez@cttc.es](mailto:cfernandez@cttc.es)

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [Galileo\\_E6.h](#).

## 11.706 Galileo\_E6.h

[Go to the documentation of this file.](#)

```
00001 /*!  
00002  * \file Galileo_E6.h  
00003  * \brief Defines system parameters for Galileo E6 B/C signal, as published at:  
00004  * European Union, E6-B/C Codes Technical Note, Issue 1, January 2019.  
00005  * \author Carles Fernandez-Prades, 2020. cfernandez@cttc.es  
00006  *  
00007  * -----  
00008  *  
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.  
00010  * This file is part of GNSS-SDR.  
00011  *  
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)  
00013  * SPDX-License-Identifier: GPL-3.0-or-later  
00014  *  
00015  * -----  
00016  */  
00017  
00018 #ifndef GNSS_SDR_GALILEO_E6_H  
00019 #define GNSS_SDR_GALILEO_E6_H  
00020  
00021 #include "gnss_frequencies.h"  
00022 #include <stdint> // for size_t  
00023 #include <stdint>  
00024  
00025 /** \addtogroup Core  
00026  * \{ */  
00027 /** \addtogroup System_Parameters  
00028  * \{ */  
00029  
00030 constexpr double GALILEO_E6_FREQ_HZ = FREQ6; //!< Galileo E6 carrier frequency [Hz]  
00031 constexpr double GALILEO_E6_B_CODE_CHIP_RATE_CPS = 5.115e6; //!< Galileo E6 B code rate [chips/s]  
00032 constexpr double GALILEO_E6_C_CODE_CHIP_RATE_CPS = 5.115e6; //!< Galileo E6 C code rate [chips/s]  
00033 constexpr double GALILEO_E6_CODE_PERIOD_S = 0.001; //!< Galileo E6 code period [s]  
00034  
00035 constexpr double GALILEO_E6_B_CODE_LENGTH_CHIPS = 5115.0; //!< Galileo E6 B code length  
[chips]  
00036 constexpr double GALILEO_E6_C_CODE_LENGTH_CHIPS = 5115.0; //!< Galileo E6 C code length  
[chips]  
00037 constexpr double GALILEO_E6_C_SECONDARY_CODE_LENGTH_CHIPS = 100.0; //!< Galileo E6 C secondary code  
length [chips]  
00038 constexpr uint32_t GALILEO_E6_CODE_PERIOD_MS = 1; //!< Galileo E6 B/C code period  
[ms]  
00039  
00040 constexpr int32_t GALILEO_E6_NUMBER_OF_CODES = 50;  
00041  
00042 constexpr uint32_t GALILEO_E6_OPT_ACQ_FS_SPS = 10000000;  
00043  
00044 constexpr size_t GALILEO_E6_B_PRIMARY_CODE_STR_LENGTH = 1279;  
00045 constexpr char GALILEO_E6_B_PRIMARY_CODE[GALILEO_E6_NUMBER_OF_CODES][1280] = {  
00046     "E6648AA5EFF0907A170377FB20CEDEE1E8D253DAC2496831010336B444276BAAB17E599548"  
00047     "B1A79C67379F98DF0CB81AE8D914EE4947093ADCB94FF4B3916EE562A4CAFDA45A04972160"  
00048     "6E55FFFE26C949D7C8B0AB7AD2F7DDBBF88A9B091510D209AAA3F6C83ECFB8DE21B77E793"  
00049     "A9E759E73A7ED6330395FD9380E8E86F0A511E0FEC8E0B6501983FA7C693FFFFFF719EFE42"  
00050     "49EF295B578E514817D9E86EBC10917B3B7656BEF6C9AAB8200B20DEE54BFD818728A33934"  
00051     "50F20B08A7AC84A9F945398E13434BD3076087F32CA165AFB94FEA9686924E831BA4BD76AC"  
00052     "F62D7B216BAC53F71FAF1A13580BDFBB56DD555855A9E2541FFDED2ADFA6125D73F116006A"  
00053     "715596001D1538B0D91C8D445D0342B910219EA64B77A7A245E4E70AD547590C3D59DF32CF"  
00054     "8ACECB6D2F17ABDA7F89DB09C96A2E811599A39191CEE8F8D42A0BC2A5D00D8D7C318E8996"  
00055     "470280F551D25DFAF153766F6D237236B23FBE1E74484CA9485848CD5C1966705EA03A0220"  
00056     "E84B1AFC9F08F8BDC110D9597973C0F347F5259591EEEF00C399D9A6EF140B759465E07F48"  
00057     "7ACE9B8CE34EB219AC63C01127520C8E2D2FA7090AC754976B8DAE5C51F18001ED095518DC"  
00058     "659B5746B390226013A6DEF1C54DF4BE1E60CBAAF0D10D163E937BC753184791966B930990F"  
00059     "9DFA0A9BD2F9060233858CD699613088876272B23C6FC27288BE7D258CE3E83A0F2D5C3A9BC"  
00060     "FC0E7C9F02AA0A97AFC31A286E9CF85BE5885FFA22E51827EFB1066452B27335C01FCAE645"  
00061     "B56819B9D7C7014B4B8C050E647B1B215A45F8B939624433F0319C840DEA265D8631A1EE9A"  
00062     "66439DA99563DB168E9C1574E2730F6AAF709AA145B75DB885B14FB78E38A05948ABC6206C"  
00063     "793C33CE9189BFF19338C", // PRN 01  
00064     "A2AB1DB576F2061D6993AC3832FF041C388E44240AA6EAB61ABB450D961ED6C8088C05313E"  
00065     "8A31B322CC757C389FF8B033792BFDACA5D3D4926321497DDBBD7C40F833AA87EC0D27B11E"  
00066     "215E491528C0919B2E0D3106DBBA384EF22B1133538C78C76C4ABAC8493E18A30E842CD005"  
00067     "01EE9A737F8AD7E7D78EF22EA826E8973BD19945A5A30A4101295D4D9010CFB232E2187A98"  
00068     "AB9876DE1C9B3B23A8328028E9111A3AD612D7583B0E6A502AA3AD7E90680013A3C2E83C"
```

00069 "475C687307B9B4B71B668EE7B2573756F1AB9FC7DF4DBB0577C87FEA7031A1A3B55D054731"  
00070 "C1F7849296D03F6B6ECD04DC25C99BFEBD810622970C160248356A15A18C3F6907E39C4BE"  
00071 "82EAC8AD3874D3EC67E77FBAF6AB25DE3FE764BD22B3EC24C944E4BC060675369DE7642972"  
00072 "DCA19A687F6680CF598EEF1B29BA7A4C15D43A6FE33AA867B929F158E7CDECC08E11B6E352"  
00073 "52AD148ADE7DAED8B17648FF1E98094E560B89B75F72084BCDA4B93D237555F3FA2272253"  
00074 "7F47AAB9BEC228D7BA9DD7189B1165C35D9BCF7791E4823FD81A05131D9DDF0F696458C93"  
00075 "FF255DEDAB2C4B55B64F5FF621B8A4773C1A95C0B47D84E2F410A79D8F959C76AEA247846A"  
00076 "1666845653BA47CFEE762C6846EDFB5DFAE144FA31501FC18D8CB77160B8B3FA5500C30D5F"  
00077 "B1C8B0B8F5060182F278BD748E3B64D8DA7FA85A48E0179CA3A2441B2205FBC5E335BC49D0"  
00078 "B1FE60B2D7F94484C2921780B31B9968747EE4D19F1AEBD314F8018F47649B8EFE6D8F7230"  
00079 "4E810FA22301B620674202651D73A3C3AF84CC259F26AB775F46FD947EA5818B055224667A"  
00080 "94C29D0BE44E925053CB0A35FC670A1FEFBF39EB00D7D64AD23AF036915DB36FAB654AD9D5F"  
00081 "39B3F97F16DB0B9522718", // PRN 02  
00082 "D7C50CAAB17471A735C51D4759644BE4A788FEA8ECD5F17E00B373DBD24917481DA7418208"  
00083 "04EA5CB366885ADFFC988EE21A0875EE7A36981726B7106EE3C7EA4C1435890E415A2F74667"  
00084 "633E1DC7D259FF2C3A1D408FEC4D1792326E7691253F6DC3E7B9B43EB546AE28E6152984FF"  
00085 "677E4585B1637B8A38C7042138050888F88EA0479E974833EA5DA1B445350D4322DB625F56"  
00086 "7F6DCC3F1A8F4096BE533BF0BC9C7F5545C5D7430A9935A2001FD1742AA34E572F886892AC"  
00087 "061B4A1D306B91928F2A8C159E237D23A4BF9D08F6780F9936FA6A63684500CDC51ED6AD2B"  
00088 "23574C459B83BA0504CEB21980F324F91A37A48E8402155CB97798F2618DC292FFABD9DB80"  
00089 "00500C30D4B85F4D6249B4033D098C08316E570B10E5C939DE08C1DDE918B749332B5E6AA5"  
00090 "88A5AF937F609414E885C77A953D8CA93FD382CA84752F54BD2779F460F0EDEE48C99AEA83"  
00091 "7877EF01AFA9687F334218F8EDB23EBA3FC0AC8CF8BB5794562E6D5B0A356CDB6BCFC0CAC2"  
00092 "ACCE6CEAEF102CA980B23A36F89C35470D0BB41C8BEB2BE0E6CD0C9689DF2D1907AFE689B4"  
00093 "9C19D4539AA1A48412B6F15227839629D2A976249FF057261940C6E88719EE8672E8A772E7"  
00094 "CD10DF669B1E966AD93C4B3E9B9D4D0148BB53F272208F192B370B17291D33573D8BA7698"  
00095 "4B2980343928DE4F7AD5E335FBCCFDBBD31C8E9FA63015327D0517EA2F161EF5A08BCC0856"  
00096 "B6D5BD210FA3DFAE918EDD7C4E444956FFF2455A4C3C71EAB715A12FB9DF145C7FB4F46A17"  
00097 "C877797EE158737A730F8937BC6DF8325815464EED4F48ADE718C01CC7BFE5979D71D1FC87"  
00098 "994524F89CA78978BA057F9D499EDC7E1DC6BFF41920FC1B5C3DE263D5768B31237BBE750"  
00099 "617E5802094537DC8E62C", // PRN 03  
00100 "AA696973C3A5E818309F73264D9EE219FC69692F106F7FCB8ADCC3FCE9816C691C07B996C0"  
00101 "E5DEDF63501B63A6F9E5F219FC8ACB811E980346BA432CCC6E2A0FA95B930ED03468AC7D"  
00102 "1C0A59035359666F1A04C7673F7FB7C557F6B2C6548979F480754D4C7C31A4A4E125514DB"  
00103 "FB61BD199830855A928BA58531257E01F5D69D222C56C0472C3354DEA19D563EFF9E25150"  
00104 "2562D7E76A0F719B63AD6917BA13C69D49B228FDE7D4181B843AE23F4D8740D942F27BB70"  
00105 "3199D70A0EB5D3C0DF04F0BABA9AE21408C6070345FA82A69792BB88F1194C6C205008A729"  
00106 "3C077169BB58281CF57DF1846C976AFBE5BAAFBF05C4948AEC474CE85B0DDA450441DAC1E"  
00107 "778BC63D1B75D867532FA3534CEEC0B796EADA76B8EFC1FBB70F4902086E7C7926FF19EDB"  
00108 "75FA9848525068D591D1F0E83995BC10C0A25DD8E22245AB72558D27F620E785E5132062F3"  
00109 "AF642EB9C83944714CBF5A98EA18402500380FAA6EAA7823E00ADD6B70DDDC1C879E486C6C3"  
00110 "9FCD5ADD734798AA70ABC07AD7867F03EC31D478227DCC91701AEA7E509085AB319858790"  
00111 "D692AC9B1A85537E7A16D4682D69E0EF5FAE63B342D86C6B610DB41022FE46A8013A756A31"  
00112 "31C0731A213C3BB07981ABD53C22103C5094636CD630BAAEC9400E574DFED45FAC5E1AF832"  
00113 "06E62D1486A80ADCEAC8A89D3B8F9BC8C88ED5F1EC4A31CB16487510A054CACAF725E3FC4"  
00114 "AF4C65A7DDBE4D0861F4593F84CE36C5275C543B573B3ADC1659257DE53396F667E8A3F437"  
00115 "5DA4AB23EB622546547C1E51EB436DB27BC3F321B15C373DE2755F3F860A8A2CE62C90E728"  
00116 "717E5EBDF09AEC67FDAC2CD84FF8CE9180B4DA29D32F52FF849F64BC65278EEAD25BB54F1B"  
00117 "33A2735EE9AC5766F2E30", // PRN 04  
00118 "AA48B0CC726F7B85B9ACE4D423964A1E548D6CB78F886DE941E49360E53EAA7AD303BCF4DD"  
00119 "F3EBE640DB733727082AED108C3C497644800E14AE78DF95D3214D948BEDCB3D52FEAA68FE"  
00120 "F9F0A98662B390CA0E98E663EBF8B7459A8526E06F639630B3790B5195F9330FFA0BCA9C14"  
00121 "120136CF11560956B598A3A656A045E254BFAB4F4C9907F82C1D2BDF35958980D49CD3E9D"  
00122 "1DE1CD86D72A35385371CDA6C3BDF42256B13EE6CB959F1CBBFE6CA7493835123141089509"  
00123 "B9D8741BE692FD6980C1F72F328C670FA91882AF369D8D75F640AAEB20923738DE0DE2FF8"  
00124 "C29CA50557CE59B779D32F790D7AF8DB5C5E13810EB316844D1CFAD297A041D3B9BE524836"  
00125 "180E5039AE7F9444CA1ED352AD4DD61F7101AE2D556CECAA7E7DF1F7BF2E88BAE57F60906C"  
00126 "27C12AF03780E42C1F0F4A2222988313326989728E5CE8E09267363575F9B2EA976495E739"  
00127 "52C05E69F955CF66937A4BE157D5BA63F529A28846FC25D800C8858F8BE4988681FDB6294"  
00128 "DAD76E501C40D8AA9B816AE8990D4EEE2A029DC9BA1717EB7C013ED8D6F57DEDA4784960FC"  
00129 "505B91A1FE720F11B95DBF7900FDF182D0469B6764E159DCC1703CD4CE9AA2A2734D07E86B"  
00130 "376A2D415C2B11FCBFFB0C249DAB0B48D27685D3CC2074E0F77AD08206DAD6BE6127A45F97"  
00131 "0EBCL1FED5CD7723230D70369E539065188FEDFB2E9968425C3D5F38565380D958075B7EAB"  
00132 "6FCC7464BC4A77CC23EB07E81248A5ED68AF800C5A19074DB8B0AD6403B8875CD9C9E95CA0"  
00133 "9DB303DA964A68BC213F248DC27FAF92E32F6B07E366F43A86EF30F8F3CB09CF9409E68"  
00134 "546C800964FCDD7F413D2DFEB5CEAC12C176E49023AD74209838CF0992DB0D2679F7ED3"  
00135 "BC236707A1342576A7258", // PRN 05  
00136 "C529A0B4203614FD1D8451D10DC39ADA617BB03A918BD89FB81D4028113724B4C7E84ABB1"  
00137 "977E35F632BC3FF3316A0D6CF465DCD39062FA3FEDE7F0CCD0764F1B1E2F27445282CEC660"  
00138 "8F16A08FF994319C9C89301FF685294971382A2736062649EC280791B7724C35D14A9400A"  
00139 "437B95161875143C86B51D4E0C319B822E8D91C09BD965AA7F83C0657A7DE5E346A27642D7"  
00140 "D638A00CE0F0B77BB84D45C68121CA4B7F9307B267851AC06AC9F4D4CAD54FE8A6280A0C25"  
00141 "29AB277A981433597F668C44961A813D6255E2BA29AEB051BF9432C4E75272F7C36A204BD2"  
00142 "0BEC5111A288E8559D68F3242900FC227D83BBD201396E3710FB2A3CB3EECA47810A074A92"  
00143 "0908A137577B3599C2B737B23D97D8015F1FFE6396C2360D46067767053C7A71BCC58264C3"  
00144 "FD54BEB7B5773FD727AEEA2A19F864284F1E0254A837E11C1B1FBD2CA747C4DFC2E60DD75A"  
00145 "F044A971454C1227595370DEEC4A1F4DA39DF4A9D6416C3E4CC8B334F44EC3F5AE23063BB5"  
00146 "0025A5A5B3C1252A979E754C40FD2E4A6B779C7B0D5CF2C0A37A569A126E085CCC007D6AB9"  
00147 "1D8EDA3DDEDB1EA7D4B5B7DBF7A8F38552E663B49D4167B0F6A92203ED63CB7AD511473F1"  
00148 "F6E976034CF31AF3C07E728E5B991FA3F4FAD379A679EEEA273AF723623D22ACF8E07275"  
00149 "89E36DECD8F8FFED7A048B911FAF616E48B7FA1829FBBF677868D901C0DA8B5D1433B3A7F9"  
00150 "CCCB450812C04D280C8C2A13113B6149F54905636CDD191454FB05613DB8C8A36536E337E8"  
00151 "807E0BAB2EDB64DF986477CB4F93C91EBB9C2D85B0AEC22457CB07FC1BD98D52BB73E89EF"  
00152 "F59B1D5BA15ACA5CB9696E3410A6BA6E2E1D94B0DCC01D876F7442EA76DCAD9FB262EF1153"  
00153 "CDD2B015F4783916C555C", // PRN 06  
00154 "F04412976580A3D5106BA358575C5549A7054E0DFB114BDED09ED2F8665801DE8F7D74E174"  
00155 "D8393187850ED269E1055C9060C3441DFB91CA3CAD84A5C11FED126714C8CB9BB934D03F24"



00156 "A0350D3287A64D186FA5C945634C71F0D7A4FF5024BDD2D79E8EB75DE7F31985585D056EE"  
00157 "F1EBA9E4C100435776A7D3465C6DAE21B1D89A0FDF0264A5AB9F9C7A4418F6B5A317ED828E"  
00158 "6D6AA566C511942203E62EBA4DFBCB818ED622CF03302E1B3A85C0B0F05F1CB78F9A90C77"  
00159 "EF8565BF703DCF5D45C8D956765F994C8FB3243CE089E68E705F310BF9E9B81B044E307C72"  
00160 "33CC1D4E3237F6AF5B1A699ABCI1FC4C1DB7A998B9CB19454F5AB9ED7D3FFE5EE55A6184AC"  
00161 "FA1D3A481BB74AB2E70E28809F92FA15CB99EA0AC7CBB1A385BBD3A8D565EC2C3AA379EC08"  
00162 "8EAD486D782FD28C5433EC11F01CC996A2F15BF3B5B36752A188FAF37BFDC170A54411D5D0"  
00163 "B36D52064D88FF4C79C1F0E2825CD6BD9C6811C0B136F9EFA727591E98E7D673AD2B81B8F7"  
00164 "033235689F9EB78A4DA85D3C26B52F0B3D8227048A191B3C2805311072B556E8B8C3DD961E"  
00165 "72CFB6235DC6ACA2D16441AC2151C37D9AAB1B13BDB6E4E82E8FDD47FD99E8BA085131C4CC"  
00166 "5D801C6B906080326F65A5C5FCD0FC22862AF6AACB79152AECE0DDF3D432E0D002539014C"  
00167 "A8C7C782F5E402B12C34F440EC7439F8CE1682FA6C2B489C0FB8650E7FB20AF59341655229"  
00168 "C450AA29B24B279F9B8B38B62FCFC166E00125D6AEE53C3CDF4CE41206D829E00BAEFA79B8"  
00169 "DB51CECE4E2CB66A1E30A1C39A2C39598B8675A935BBAA4E929DF27CAA3D410BAE9B74DAC1"  
00170 "435B3E86C9C27FD5112A9B9D9E7F7A65B14F08962F34013DEC906A487FC233DB8BD24DE861"  
00171 "6A8EAAEA6F4BD18AAFBC3", // PRN 07  
00172 "E77B8CA081B04234CC2E148ECDCC38912DE5D2239953512C2253D3624D52BE57DD3F3813FC"  
00173 "CDB94E83E074CC6023993704D862A4238F377C5EE88DF60F41D9F3952FC85CFEDC500A75B7"  
00174 "116E85C30D978B5396FF09A922F9F065ED050BA7EA3FAF59590EA167A10270595C08F5F60D"  
00175 "34F232DAAEFF2E4D9E0BF50568E619A29175BEEB97A17CFA4D733AC70269F76219E480162E"  
00176 "E76304E0D07FCF1033829C13D2932619ABD732CA49B9ADD983725E2B104C15DC495E01F9A"  
00177 "F7A398DB00497A9ABF8E2698D4360C37364D752403B0B625AF7B5988D7BBC11ED37BFEEF087"  
00178 "D9BED79958FB2699B5382D318AE8AEDB97476AA69E56FCAE2676F0F3182FC2CD8959EE972A"  
00179 "99D54BD3336DAC9AC828EA8C421C6557B4504D655CE7543112C43554C472B50263B519D59"  
00180 "F5515766CFA480444D6C1DCB15A68386D4D518154B639F2D5E01E086EDF1A73A0005AD91AE"  
00181 "618B7B05ED0C3760DFDF522593F169CEBCFFA228BB960350E3BF67201FBD21C8A2B82328DC"  
00182 "E6C71912BFF568097730709C54F271D3201AC563443361D55A2A3E989725EB7C510823A2D4"  
00183 "F72FFEFEC81C30012FE8EEE479CE1F53E1A12D15D2BC51AEC23D5C86F0AF59186691022FA6"  
00184 "74E5BF405AA96A7E00398BDF02E4F9E66852C44E5E7D583A7D445CBDC54A6470C083C44340"  
00185 "A65F6F7407A4C7BB784EB4D09A68FD41E27568294BD13DE5C2AB7D4C7811FCFF67A97842B"  
00186 "71219D9F3AD9899E8E334960DB56596A41C3645FAB842FC1CF41E8997CEF1277D961620C8A"  
00187 "883DB761F1A126ADCA93A417230A419C00E4F7BF7FB3175AEB1EF614BDDF11AD3463715F39"  
00188 "A61B1B001D5B9D4D5FD356F2E94E769D0D23DCB46B17E5DCD091AFA2CE2CE819626F01A14"  
00189 "CF5473A1389AB840F9EE4", // PRN 08  
00190 "BB25FD99C731209E762D4328C053ECBBD00118201FF7A8836F8E7651EF75F52CD8FD48662D"  
00191 "53F8CDD9AA770C4CFD0B2FBFA07E59AE1BA177684026A24E4D96BBA3B856E56103752A11C3"  
00192 "7DACB4426213D5EE47FC82C0DE70BC4973FA024DF51DF567659E6947CF51D39CD21CF7B891"  
00193 "19172653543FE54F1C57232D3ADCB347691B425D566996F5F8DB09C31E309923AC3D5422D3"  
00194 "760F1F22B9EF7403A7FA6F30A611CEE87B7C46DC839AFE30BEF927AC6F40478AB4084C3A5E"  
00195 "0864BFC3625EB494B2FEABD0A0837A3394DC9D21D03512FF94E260665540C6AEAE2C535295"  
00196 "FD4BA4ADD9F9F34C1EFC4807E65D599498468F539FD0C0008AF77B926244C3AAB5857B9F0E2"  
00197 "510856BC371210BE0D054D96A87B57480EC43BE4AD421FE0E00A1AAD06422990312B86DCA"  
00198 "F2E26F2327CD33DCB057FD4CDF94F83B578510C4D7D5660A864C6CB1102B7CCD3C9F43F586"  
00199 "2443B9034B58F1DA374749369C97DBE6FEE8EF009328254D5D66FB9F45522354ABC7A442C22"  
00200 "28EB53C41C7368C7FB798000D79F0C593F1CF1E6389F1F3F973354DCC74ECD94299D2CCD"  
00201 "3AE11320AF5B88027641B13245D7149B6DD1EE8E10B828B6F2C792C597CBFC2985BCD6304"  
00202 "8EF90154A509D352B9AD20259EC7551572CA9665F7B3A96F19AF496BCCDF42FEB6340DCE0A"  
00203 "27A4CEFD4755E71751DC9C5D6C9A5428EA2B7883E9937851962E3C6D817CAA8EC566F93EDF3"  
00204 "98CBC117B9372A01B2519225356A8DA72CD6B515DB8B5DEC3EAE7365650114E82FBD84686"  
00205 "EB11DA3F5E9671471C844988375E4105C37E00CBA220A3B1E27A049B9753D064E21E79513F"  
00206 "892FB5620AC48E3CA0070FEF89BE933C52E6C151FC3CE36C21155007D1B67DCBCCB4E0CFD"  
00207 "3C2C59A81330B720C3D58", // PRN 09  
00208 "946B6A0BECB29F278563193CDDBADDEF407C8789437694F61A4178F85002A7D0A5E67B3E"  
00209 "EE8B91F13DAE56EF335D953FBB02C56BEF30D21C00594968C5EB59C09D778C637D1283171B"  
00210 "39026884B8412E2AC160D222576490B5E901B7DFCE1FB46D6166968079E9695A8B9EDDBF71"  
00211 "ACEA00BF74CD6B0F3BB0854B9BC014BAE5D6B5F51960D4768FAEE75FBE96CEDB25F2E3D502"  
00212 "11393659553850CFEC40D40BF5C2FFB8110796D9C8B1E73DDFD086D289B4028D926462D585"  
00213 "01194D092E0C6A4100B46320158B7462615EE9CE6430FC9D412BC3659C04DFF276FEE903EC"  
00214 "7C7F847CF72F3CEEE3760018FCC0E27517F80EE5A1565F7CA3CF54B18BE0222EFE2438F306"  
00215 "B6DE2EF6C771364083338CD76C74DC0C98FFAB349B1C9E7DA35751676A390445131AE564C"  
00216 "9A7EE77EFAFC0772931463EF926DA482ABBD23E0172343D595F7299B17633A34F652E09BBA"  
00217 "A2378EC84B849DCBC21D64CDECE910779E1D4763CA753D347AC02D270983962F17BF8EE4BE"  
00218 "62AB3F92C9831CFF23D44A83DA6BDD85955D1101D93EBB481593450F23C8DE87A2B1201616"  
00219 "E86E05C835B4726DB6B6A90EDE6178071CFABA3C7CF19B4188F0C679CE4EB16B26DCB1E665"  
00220 "998691A079B5772BD72E9E44A70F28C79D9ABB42250AA4116C75861B7F8AB411B9DB6AF52"  
00221 "DF24955170874250450DF1C31782F002AD88747AFDF2FF542139E081A300DBD69B2B775798"  
00222 "6D6DCE8D349892B1528922BA05F25AB4CDDA0EA8CC3600BB3704363B5A20AA02AF3A3F703B"  
00223 "2284D5548F0019CA0C7C4E7180D0A5E7BFEEB3390EE63692BC66BA8A7BCCC502C06D953301"  
00224 "6273543991B2B37B1ADBAF15DACD696F427F47AF62E3F72791951DB035A4D0B7BD3C8F1753"  
00225 "B85E3B45C177440ED15A0", // PRN 10  
00226 "F28C3E75420656B8672590CE0B35CA3246A13D45516426E9FBCBC7451AA52574E14177063"  
00227 "D74AC2E20CC7C1F4FB895D5AD29118DD3748E106A9C7D7CB859A32887E80D842F92CE8B3C4"  
00228 "E285521358848EEDFC478872A42150DA67328949408855488BC9CDF3B50A89DA0243E6C01"  
00229 "5321AC375063B6BC648C8CFA3D1BC3E01BF50F6FD3BA35D725E5B05A308C577296ACB7571"  
00230 "DAD976027BF602FE4C3446DFED2E82A086BD132FF872E269F6770F0FA7E36ACC56F7DA2E8A"  
00231 "5044319C2B8CE0999510F1C467EAE4BA80E86EFE289D929953E561BBCC052CA5B8AD2F58BCF"  
00232 "E1669466BF2FF823A66ADC01819FE7977AC2F0DC7F2262411E3A790B5318FEF83A8BCF59"  
00233 "29063928C1AEDE28877A6374C668A0A2E54DEA5AF1730277AE3FF4BAC240E0CA97AF936232"  
00234 "3D45AE4BD7886D3A2756170FE9DF751F4827684A43030ED6E59587D242463CABF5F9404AA"  
00235 "6565F305B6CE1CF4C6F979602801530E1760E39BE3102184C98E633EB4EFC86E479F6D2BD"  
00236 "6E16EDF573A7CDCCDF005C993C5B35D6517A5E703C57037656D1CE0B66835ADDA24684FBC"  
00237 "9DA33002DF1FF7A46BE81CAB3ED4EE9C5C9EE0C61CDF473D500AB3F00CB21BC37DBEC7550"  
00238 "E5D604A8A58BF8770643FE8E165268B0F3B7CC015D2CA76B3230D14D48545F419B5AAB0C9D"  
00239 "0E230FD44618D5A49C5749C7BED668C7336806758B629D823F16894FAF8316085A88DB6BF5"  
00240 "19AFF6E19412A767FF1B31ACC3CE1327E97B47AAE48E368EE95E1D6EF341968883CD2E4885"  
00241 "DFBEBB7D1E8FB51CD20E5A62B321F010AB1738F31CE6797114340AE65CC0EDE548D0220A2"  
00242 "81F4B2BDB733DEF7D792F401E3393E2BAD9C279ACC051AC1ECD1819DFDB55603E6AD7146A4"

00243 "D233DF28D3A4D85C16DE4", // PRN 11  
00244 "83E25EB35FEA9489F9460F1074A44727711FDF5BB180D3FFD3EEC59A3B6956D8F262BC9471"  
00245 "D48479DF7DF6119A0B3DAEF645168C4AD4A70BAD6A31CE0AECCD882EC41E52CAB2EBD3C3E1"  
00246 "24BA6DAA703172AA6BCDB685D365E4337F07594B1988F67D0B2B48C244D691415CB50FA2E4"  
00247 "F32805CD12DC96358992BAAA34AAAF11F77CF3BC6F8EC079CE00027235A9518AF9FA168809"  
00248 "4F9A43685DC578B5C4F291B86B915012F16E35E4CBA9941E6CEE2144BE736703A55704998E"  
00249 "2B6850B4B6E426C4591884958CEEDE53C6C5E21A773D823AE2851C1841E2A93FD2A48E51ED"  
00250 "26F00364FCB0570348219F5B528A546B00850E012E114B3663B00F5C211DF22BAA1F2EC924"  
00251 "8874DA57C73F639E66DC8893FF362F89D776A1663425463FA2419583890D34387FA2B59EA"  
00252 "B7FFC8365E7F6CAA7C614C283E53CDE60CE85672B66BABB95E2B74ED9EDAA8DBFF2ED92859"  
00253 "0A83519ED3E66FB0601ED0211D29C8E188EA63A9D914ED789812BDD3A51F7C97C3E01EC1E4"  
00254 "7D6EC642339E7EF2D4180F2E4B3DE1213FD03F7FBBE2CA7A0298195AAFE8C1E70D5548F13E"  
00255 "DA2CC8788D6B1D121F577AD1CF3156E87C9D72FB97A6DD0BF3760A82883C74BBF01D856D32"  
00256 "96EC93C618CEFF9304107430A24B70DFF3C6359230E9A59DA41FD62DA7A3BEFA6443F2DABB"  
00257 "112540257B3FD7BFA9AFA0B3173453FC4443D23185FE4AA8C99D1099C735F65893A75345E"  
00258 "E2890A7DD39BF900BD90C634784D051CCEAC8C2383E06640EB1CD74AF88EB8FBB9494B8EB2"  
00259 "00D8A197A8DBF34B3D1F4823ED06843FF370883C5843CFB9838D4CF9409E55C89F951AE2A8"  
00260 "27238B2B9F72C141B659C2BD18D3CAEF1A8B7B3ECC28331A0968E749288E74F9E329D7DB7B"  
00261 "6245E4600EEAE3F300A58", // PRN 12  
00262 "DBBEE74EE5F69871BD6AFDAB05095D22D9C722E36DF5651EE4DBEBA48F299D800BA9FCEA06"  
00263 "6814492C5CBE2EEC67E10422F250233863E517CE10A093A44015D7557EAD2FA184A6CA933E"  
00264 "F405FB67547212901078BC627B5DEC3339FEFE0C8B724D88E8A45D29CCAA238396C123DDFC"  
00265 "A9FF04B21B0A0DACDE515B74B33D80AD3990B6F55069981C4026151C947473F1004B60D5A4"  
00266 "E2EFD3C3FC9498EBED180601DA8DE3795034AE8DB8627F1C88A4CA0CC19FB2D6B77649FA85"  
00267 "9FF190B626EB5571DD9200F58A14A56AB43A8E95ADDE4B54EAA11DBC0280067B1D3CFCC377"  
00268 "B7AB21CF46C1A94C5AFEB03E1B1349CAEDDEC0616068311488A0691795324DBA11EFF29D74"  
00269 "7489241741D222534DE334F0206DD7A3593FDBB1EBB29B0A820A349742997A3AFB3B5F26"  
00270 "63EB71F47A89FC30BB759A5679982DF3DDFA12F05BFB6C73F188817CD8D5D3B3A363AB9120"  
00271 "8D9A3F743CD71DBFE4618EE0D949FC4EC73E5E30C4FCE75C0938CAE89944048FFFB67C43F6"  
00272 "00EAA7F5D73B0C31C48B88A1CB724A0F5F6B4D258991368C0F1D9D8B237ADC87C72DA0D2B1"  
00273 "7B121559D30E3875679A812F82C386A57B53419062F2C9FA664306E36CEB512238E867C8F9"  
00274 "CF9E9C2213A08B08E4A24619B0D92B5486DBEE8D2E50CD64FECB48D4025D294B59051E52F5"  
00275 "95E2457120F21203CE999F8ED7F75731BFCDC3E356D629FC5C98FDAF8BF7D80C312657104D"  
00276 "B3889BAFA852FBB709768E619AE13FDFBFF186CAA895DA7E3E20A280B6C3A6B97321FEC27"  
00277 "AA83BDB572B177F12FACD70A0C920791E258D81730DD705C65715E46F842DBD2FABD35E386"  
00278 "166F0EAD43805D3237491AB40A40B7C864AC4E69F13BDD254E34360E439DF0960A54271B"  
00279 "A93D706653691F0239FB4", // PRN 13  
00280 "CD41661B18894EF1C3CEB0CE7B301352D1385E2C50A568EC04856ED4D4434F964A47013724"  
00281 "40602E3156A45416C3F75C21C0262242FD26C878B93952330FED33239BE8D11F4E6C8576A5"  
00282 "761C8A34C36F8C0F8C81FE0663DB900B49A572C9C9172ECBDD004AC4FC44257C6B3BD8DA6A"  
00283 "619390C626AFA0A932EEFF989D3737F77E2A40C4FE221E125B63492729B77FDC493D5A4FA"  
00284 "A2BD564149AF90E07497B57096F65BFB2E3A32044933C42B87B64F7D8B0A3AF0EF6694B8C3"  
00285 "6EFD97C97B63175DF5110F85809E478FCC25D83D8EB303C387151E031E4FA4A4C926CE9D89"  
00286 "3A301F948347B191BABD5282B4A64ED6C389D1BC22D324AF8852C7956005661708A77075FDA"  
00287 "82EB3206763CFF42F74A65DCC3BC33E408D561D65964F5F2599469550A1C420BF4B4614F"  
00288 "530715EEB669EA819F5F288BE16D69F07517A9523528B1F5DB4826B11297F1EC01D933FAEA"  
00289 "FE177F6ECF5E4C52DFF4F027AF5D9F91E5822B1EA23254AD3C2AA779C6DA422B976FE08144"  
00290 "35F7348BEEAB926FC75690EE13CFBC421C305F565DC5E73B73B58111F1A87348E6EF9CF00D"  
00291 "16FB3592B75F2888A1FBF0115A60899ED57F8D94843EB3FC87F7AE96EBFA5B381F6AB355CD"  
00292 "0C301F2B6A5B84832EA0D4E047007ED0A8648DC36E16F21E73E4A0E4992C1C2194C325646B"  
00293 "CA5EC465D5BC8FF9B1FCC676E34602C271B32C4E93740E13B84C4CB3A01451E0593104EBDD"  
00294 "9387ADAA37755DA90A15331263F904BA907150B52A7DEC815CE5A8E22DB7732C06C83DFCE1"  
00295 "D118F6904C37211E0E56F534108D539307D44A578A821567EE4652EA0EDCFAA09EC34DF584"  
00296 "977EFB692428CB1F42D72E2004EBDB57F90DEA9C9887CC6EDE6214F5F8F686CEC80C46FB"  
00297 "9DFD71EE4DA766EBF6F2C", // PRN 14  
00298 "CFEEA7B366F2B3687C364AF56BC702885EC5B1D0FEBBC78201FE6C914CBA004720730BCF1A5"  
00299 "88A73A8E31F13853DA6619DB947B00E9E4580B7925C9198F887788A986927FA2B911489016"  
00300 "B959482950F61FA58844601EB766DAB5447AC006753247A4DCB3A541750AA4A6A0ADBFC7FE"  
00301 "1F52EB6D2142C6E0F0F65C6E99E1EC23064CA35A3DB1498C42707C0581EDED461B55FF49997"  
00302 "5F889B79E4733133854B1066DA534873678C4FF8953A43C02F57D1A692E3B153FC1C0F7FC"  
00303 "28C8A0EE13FDAC23BE991511B8F15ED706CBB735599F55027E89CF58F0185EE2AE805CBE75"  
00304 "48D927956F4F75D938BF5A17220888211F2E7D759028E32CD6DE895C1E408B4727541C4DA"  
00305 "51156188B6DE0376F1A3BA9F15FC1E2C3470C074E393380755AFA3F5EF8E68228CBE07CC42"  
00306 "B7536307D8AF414F599F901DF5D764DC72A7EEB976CCF29AA043C5E2B203D9128ED2550B8F"  
00307 "D14425E09DF4BA05BB7652C4647B2DFC9A50C20E002B0A50F5B3C6112F9BC9BA3120F2792C1"  
00308 "A84415D99D8B762353838B1176EDCCA57C82E9307FBDE66AACEE5C4A2599D9BD826B53362"  
00309 "22E8E9C9F2489356A96B0BE97F6568D383AE1B6A8C53252EE81070B86172BD2276EBABE7BB"  
00310 "70AC32D13D589821CB53A7D6EFE1437BF6CFEC2650B139F37516AE11CF67828937C0BA6C0"  
00311 "9354EC70335F286B5C3B6B1C0E5C9CE4F2C53D54FB53342F7832D8C71C9A06AC78792F9EF3"  
00312 "5431E61150112979F4351FAE2F3A0CC33163571DDF57D3C8821734D779B0C5BCD7E7BCCBCD"  
00313 "EAA726085271ACF8AD0B48A8BA8495A21C958C73134CDD44AC3A96F0A190BD8BC6E1444F06"  
00314 "95FDDDDF34FAE8917C0101B1FEF10710DB19751FAF76E4E7DCC2A7FC69149F5D9F2C813A6E"  
00315 "3D5858446149475C9709C", // PRN 15  
00316 "D1331B8F157416095A0ED294C78A4EC987D3F8596ED5B9A558DBDA2FDB262985C9380877AB"  
00317 "EFECAAE078F7EA40D8682091CCE5DF5470562275B320D44CD754A08A1ECF7EED89C377D17D"  
00318 "BF35D6064B82466C93C3099B0F9C0D096B0ED064BCA7B6BD8C026728213D29C0A58CB48DD4"  
00319 "6063C9ACB1931C0FDD2C94C03ECFFB7668799C3B4A24B0F05AC39E787D942C2E2FCA85FD5"  
00320 "95595130665439E1D2FD994ECE255349A86B8F5B3104D33EA40B236EDD75F80262A3C04A81"  
00321 "F051BDAC6C289A437BDB99287DD787E9A4282E75C82C62E9E249A8CD40217F7D036AA5AC1"  
00322 "52029A1159391AFDF4703EA63CFD0FA47A425A5CAF2121EA85DD57498677782BE381F4826"  
00323 "08AD3B8D4790C3850511676007AC02AD1BB4FF01C7E87FC4F8A865A61831454D156ED442F6"  
00324 "3AEFFA71BE075698TECDD9633B711E0E8526AF93E2D2918C427C58D1CA751868ACC69DC2CE"  
00325 "567DE9B634046DCE84B56638B3A867ECC2A0E5E44D209100EABF083ABF96DE603C5CEA39C0"  
00326 "27860947FA020E6C940DE756E708A8DA50F4EAFE3F9B07D7BE8D1402101937F616E9B26BE7"  
00327 "AF408020BE781D7AD1AD096B3303C62443CC3057F7FAB70490E3BD8BA8567C744942536BA9"  
00328 "31D26D27EB1FC1AD1BDF12AC583F01E9F721F4F77392514E63555A60371CB5B5B649C3F1282"  
00329 "ECF6625B4CF566C14B5E66B57CBB5FB3DBF7984DBD34DE76FA2F5564DF43B88DC5CADC2978"

00330 "A7E29293AE2D9FA8DD53C8929BA91074D46A59EB2D88C5941DB092D1A1DDE5817C4B8CB7B9"  
00331 "8F7820EFE599410A6717D248AE2E1084EAF8C67DF3E51D143CD2CCC69377C2A74544056ADF"  
00332 "03AA640C2DDDBF022C419658D9CB05B2BE27FB3570B2A6B3B87799BB2620F679D230ACF39D"  
00333 "319D05BF0CB7DFCF4EF74", // PRN 16  
00334 "A6AC8E0C4BFAC1FA8AAD408EE1E32A7AC3D4CAF0FB55ED10C80168306E61CC5A25713D29C"  
00335 "4E7B150EBA2EFB83402D6EF7DAAD35E7196EA409FBACDC0E472A45AB74CCE67D92558B2DF6"  
00336 "D3A73AEE2569E099AD0EB27CC90209448FD516C59FC5D4F6973507BCA26BB68FC997DFF7A3"  
00337 "094A4E8B5ABA414C20BDF30D228467ADAC6AF196B7E5E1864BB3FF80339815E8A23818CD5E"  
00338 "5AC8D72F46DC63FD8D55E0DE4FA34A7B1BA75C0B2F5F3FBD128774C810E09D52E95747CA96"  
00339 "EC3C72AA44A3BC2CF4C3DDCAF857719980034D217C53DE0884AD51AA6E9A7837038640B549"  
00340 "77E2328158B8E310B3F99F5A63C4785A2B5E1C86CB174E81BC594AE50F53C55802B25A5301"  
00341 "6C2D96400FEB8C65DE47A597A7CA01A3C13610EDB89DE57608C5824A5EE5949791ED1AFBBA"  
00342 "4B92CF17B70E38C5F5C18110DA33BCDC6ED8FC23B0719013CF921EB457A3EA663598DF2256"  
00343 "34A3C17DC8C4CF0F219C1202BC973A22E416B41B82880BB20BC2BDAEE1DBC3FE3AC31AC9B"  
00344 "D5817E3F0465079C3A98BF11A466FE3B206D2BFA4CA84FEE08B21F1D44C63563E278143AE4"  
00345 "4D7E8FEF13F2B8CC1851E6D59BFFB38E3EE2E29D8B13DACFF3B7A86C0CE0064712D3A2770E"  
00346 "3E4EBF2BCF885BB74E2DAF09A6F9A2ECA620109DF36AF492E9F1C301D3FB5AF474FFDE956D"  
00347 "3717DB57852653DAC7D1A756B5908DC969143D56D315540ADF080A48816A1395E93A9E2392"  
00348 "0C3BDA3CCD2A275A848EBE2677F90AD215A04C1097A1A03AD9E2109DF4EC1BA189135FC47"  
00349 "E034CC875325440D10E9C82C85ACCE3130B98E50B047985B13903283790D40DB1CCCE969BF"  
00350 "64151CE530B316DE2DAF6BE27B85B3B5AC022E6BD3548498828F58B13C771DC8F76EE600B2"  
00351 "DFF7D65968BFB04E65D98", // PRN 17  
00352 "927E57AF56E62F2F8AF99E1310FEB6E609A1784D6AAC0117B3C693D423BDB7E10D7710257A"  
00353 "13658CC6543FF4F8B7A4CE05D9FDDC1DDA5493B9B3B74DE75C184EC37D2BEC276FF5990EBC"  
00354 "0FF710C72BDE0BFB805E3B7DDFB0AFAD6E210C0062A62A9ABB7B69F94625BFA1096F5CD3E"  
00355 "52DBCFCB3268096A738D85225D8C3B49AEDC968187D266FF9B452DF79F4D2CF77659306CAF"  
00356 "3EB157005096A1E94F708F88276CB80056683F7A44C3197C519B559380411A163D3A21D9BD"  
00357 "EB5325AE8E0E9B3F293509D667D25C03DB6474B29C4A67D43B7F627DA75BEE37AEB2C78EB0"  
00358 "D768834A3A0E5552F72621D18C446DF3174C0E6F8CB9456DF866F455263B1D7790CA6F0681"  
00359 "8EC1418D675E1CD490F9E80A6FDB8E55483F9453DFACF1ADF77EA084E8E749678862E8E9D50"  
00360 "5F885ACE09EB89918137ECF53C96A7E25BD8CEF5FBFB8E1A39A445C172703D9EDA9DA9D9"  
00361 "2EABD3A822E16EE0C9DECEDED407783B4B6B6E0130A9C67871163E73450253303E5E394906A"  
00362 "86D11B70240B79A9A91EC381A4CFEC3AB81BDE8D9CD001D414B3155A7C497BF573A27E1F9"  
00363 "2A7FC5206AEA7E400CB806320E103C60A9A8AF58CCD0C0EB03A4EA24CA7D106875B2A79FA3"  
00364 "A8A8D1FAFA248DB4F69EC28FA632BE82D439ADB1E06804FA985F542A9852CC1522589EBF4E"  
00365 "93CADC364B3E0D62632BE05959EEDB8E45BCB65D8729E975C8988F8001482CEF2C7AED6C1"  
00366 "007BFA9331099D833505E1B98D51411033C9E85F4999A43362A89246E0AB05D399C5B614D"  
00367 "CB162BC6AFEFE14EA5640F100B2A8AA248C9413F0672B75AE08B4172CAE79AD16E43AC9F03"  
00368 "A34CD16E34ADBBDB942191B2764C9F2EF3C0518DC58DA052CA4E18E537846E50F10E88800"  
00369 "54F69C3D8E0A5392E8538", // PRN 18  
00370 "999334A54881FD1A03D6C8FCC0F974F27002DB6ED452FECE0585F7E67C29FA76EDB9179141"  
00371 "CEA6DE2C9E71210B6F6926E7225BFDD2E1D7FF19800B4BABF8A8C973D3531C30C7415535915"  
00372 "CC5C82980CBB956A1D7965B38D2E0E61A5DC4B71256509F08BD2471A3BE363E30B9E95D1CF"  
00373 "D6FF87B1D67C5BAF0EC1D0B364BCD0FC6BC3630FF030EC72DC9586DEEFDFA399866DD4ACBA"  
00374 "16D118EC33D15979B2D65315FA3E9B3209681E99D8D9A74E60FBBBC2578441DF89634033254"  
00375 "96EF39191573786976D2461888459BFA938C553C8AF97D124621DD30851788956C43301887"  
00376 "84297DC079D748468AAA119FA35606A8FF961AE3D8DC0A83CEA6F4C525A1867191042DE0D2D"  
00377 "381EA126C5A427AD4791335AA76AA903FA4C5FA8366A7AA07AEFE023360A1AF93EC576853A"  
00378 "1E75C2F5FC6EB395C65DE74B746706AAA558FB278B2641F3E2EBB912B99458D20014F588F0"  
00379 "70E80A38F9A86A6202C17D7AFBB32DBAE8CCCB6C7DA1F4ABEAA76ED0BF1A9182051A64DBE"  
00380 "176D49D482F13C0CA3D23653FEDB0F342458617C4CA7E1355AED9B173F03C503CB8AA7A96"  
00381 "215C5284443578C2BA885859A32DCBF2D6A0945D70E0C100AB2F1457AA0543757F7FEA800F"  
00382 "BFF2FEF67B936CA81FC91E625D35993443E0458981D76B93D658EF83DC3AA01CF068E3AE43"  
00383 "09FD7ACE741C4B3F1B3A04743B6EAD6A3DB9E7A0A231DA26844F4E1BC7A7C3D8D059AD5184"  
00384 "D89625D00E9279F90A5F468AC7F76C0367D72AF1A9B3631A23C16AF444843B30612A27920F"  
00385 "0AB07B6C744C86CF7ABF281B2FE35EACBD1973D3834F06CE3919E9A0605634E6CE3D2995CC"  
00386 "951D550CCD38E1BFF1BC02A7D9116FDF6EEC56A4CE44C89B1EA460625596D2D13C5423362D"  
00387 "BE36E6F8129E0018D6B38", // PRN 19  
00388 "81B17B29E24D537AC89F10FC1EA01C8A954A4295AFC1FF83540B172156B063675D4D67A24C"  
00389 "C84993B423F481B3B8227F4F054A2C0EFC519898C2EFC0CC321711E5A87077586DC59120268"  
00390 "A1434ABEA3F4812157C0877F6928162BF9859B3E42990EEAC9CE81777379B67FD36A26E8E64"  
00391 "5F6BD00474F49A4E037DF6D15AA5DBA827E8E133D2C9E562FBA458AC5E08976D885AF96570"  
00392 "39B9B8847FECB973B57B9CD756F290B8F82AED64BCCF4361ED525F766507DA9E874B8F0FAB"  
00393 "5BEEBA885603C0162656FB691755F244BDC9D758E323101E9C6AA9A51ECB6430E64F8185D9"  
00394 "33783F1B324B31A15874E70B3AF6559388189C0FBC74EFA9FCCE23E72E1E99DEC398200C7"  
00395 "16937D4CA98C6F0E45081A5FBE3731FE5962315AD0AF432F0C8D6E8FC6B3B01E083AE1F4"  
00396 "DD2AD886E15F29755CFC4EC2B6E644FB0448B2AAF86BC05966A01AC191F1D47A29DA34A77E"  
00397 "11CD4C72E9BEF50B4964E4D4F37B8B218549DD6ECC1FB6EDE38B45D2ADBFE534B736AA246A"  
00398 "838F86CC6851F63A9D10F3369D151ABCB63FB2B519254CEA1D8A547F77CE6933A6A00F709D"  
00399 "D163326E0CA45E892C8C143362F23930FC78DA9FA00472986FC268E5BF5756B1E28F0FF466"  
00400 "9DC9683EE150C7E71ABC4BAEC0EC67741EAE2B973463D5B847C38997D7E22F0BD78DAC4EA"  
00401 "4D0CC230F6CD6ACA4FA9B92B2E340E33C852FF0FEB83C113874A2C7FECACA4E54E818D7342"  
00402 "880093F171F062F3DE7000BD475B62850BF7E6B85174FF2B1E6737B258C6A2B735137F3C1"  
00403 "B2DF9BCD5CF524B594AAAB386420FF87245E8692D6BE52701CC6AB24CC47B0647D7629FCB"  
00404 "F02D620063C14F9E6017AC95815138D75C650E391062163BA3A213C00F55AD2D418C7CAE60"  
00405 "3A6B397FB20AC9830F6F0", // PRN 20  
00406 "ABFC64FF4A31070A7973A18D50A243C32E589526985495507EE173AA47D24B043F6310BA9F"  
00407 "2F1CD9A39288CC61974379CB8DE105F84540713FC22F4E9A9888DDDF37A5BF3195E0DC5AAC"  
00408 "393B61CCB343AE9E4BA94710E5498AAD096FBD6E7CC5571EF49499DF0CB4E5E5749A182E"  
00409 "473CC085970EE388AD16A0ED63608FDB43B4742667FDF068A54EBA314FEAB2C994BBB06116"  
00410 "DD3B2F0E9450085A2D14ED14DB3C6676FA8C49A8B18504279107B137B7F00D8F6DB63FB24B"  
00411 "3CD97D8DCF86A15A16F7DC58A40C9ABEA0A2A4A7F0A53A9FA116CBC1F72BBB625F58C1EF61"  
00412 "1CB8D603C2788351CB9CE94DEE3B0AF607EFFA84D6BAF86744F778E51244EE174829BB5998"  
00413 "87EF1F174B96AB7EAF58F2EAA99A67D4F8C219FC49972DCA210BD1173A4875F497B80B4B30"  
00414 "93C801D4D0AFAE4A89EE7A14618CC2F9AA743D10D0723A518DA46C42A9AD0B7C3EB2D72D2EB"  
00415 "C40253860E6A08470F44680353C9AD938467C5B08687004DB4BD6FEB29BBD051747DF4B8E5"  
00416 "4798076871DE9667F3EB9C8D87FEC1DB9744B1685A7D5FAA06CEAA8CEE6934C96768801D00"

00417 "CC84596E1BCA7C1DA93D9329047F54EFDB99F55DF2CB34CC3638306E193C1E33D840E22DD2"  
00418 "E93C54F553A816EC643CBA23BBFC555A6FDBF4F8D6A935772D0DD891BD80A481A8A6F2A40"  
00419 "228B94BBB85DA6FEDC5E77CDF0471C58A103043AEDA66A811800C0EC300FAAACBA690E4D6"  
00420 "5AA344715F667C6D607E3AFB12F7300E1B0621E1346A73F73AC0D58B5D5D8F9FE2B359224C"  
00421 "0C3B0CE84F7876C38E589C285F93854C9D66D81B47BABC5FF03B4B8FFD24D71F3869753A57"  
00422 "6EC8BA8DF81797283786C450528FBAEA9E84478D4E66077E3B37800508738B80DD40E657A4"  
00423 "98EE423859A5182DBA698", // PRN 21  
00424 "B34C68E9590ACA5456510AE673EFC8AC2655D2BFA5EBC651AAC11A3547F47F61DF980294DC"  
00425 "38FA1DCC62447EFE19A05E99B5E0CCA2C0191A554654239F665B401E3B3E7F2328B3687347"  
00426 "A04CB1A5F67AD7F92FF83B381144016908E6AC2C5B9886EB30443B7C7ADA88A5F1B8F5B871"  
00427 "C9D66D8447645C473403B0FF54861773875A52B05672CBBB396092EB4351E3F938E693C5B3"  
00428 "A20FB88ECDAE480D8170C82F2279A8FAB2DAA15918A4ACA70C56B5124F12F310812F7B35A8"  
00429 "AF7E9BA97523584F6EC9EE840DF2156A59E7EB42AF6CD6D2FA9C24DBFC843ACADBF061D0A7"  
00430 "4582813484B6A23970A4D0DFD330C9C6D3B85B1CFB60DBF8C2E5D5AACAC2DFBB359F014E2E"  
00431 "6651587A7AD0D6CAC4533F5A1962DBBD2FBF8364D81764134EFC65DCBEC7C59C332766F1BEB8"  
00432 "31D8270A2A927F1201FED121082ADD3E4824C848C93BC1D1C8E79239C6F0C431C5526EC7D9"  
00433 "3C71DFA537C95CF583AA8020D4CDF88EA14C3FA764ABFBA553C0377F48A1F61F6B3A865E1"  
00434 "D45B9734E23983EFA4B98D9422F572DCDA2A2EE1AC8DE3ABBFC56B7636D4E874F469AB276"  
00435 "9039DE7ACC7C2D0FBC12CAC79B18E159D4EF632E35546FABFCC1039BCE062CBBCDAE923914"  
00436 "C783CAC9E0371A3D201B8D0FA5EF18D50138A0FE759A20597A00EC33C85506AF6195A88E03"  
00437 "B95540580F1CBBAC39CF1CA732CA943B2EE5E9DC7CD003DCE77E791CE4DBB91458404A60213ED1"  
00438 "A7A7242824D77F54642CD660DA5B92BD49BA2277A4475A190379D33F1D79EF412279F4B35"  
00439 "E810C23A8F311D3787065FFC17FFA2CDE8C174476B4BF5308F799164A2E6FA1572A6E38548"  
00440 "0B2E7B152EF45AA4B6B9D8F8C30040B3FAAF202EFAA0DBA1461C11178B12806CC3FF787B6"  
00441 "DFD4C42E259E48E830A", // PRN 22  
00442 "83ED043E6D7B295FB8011BF3380B80F4DA4E93F1A605AB5461AF8642F2DA006E539582C0E9"  
00443 "45B61E9FF9B93B0DC96C05FE06632B8E996D0B2A49920A7DDAFD0B1F01612D1ED8133D76F"  
00444 "4DB5FFFEF461FB72A784928F87431CDEDEB4F97F3E377FE8346D79B064C42234662116F04B"  
00445 "BF66A0296486AF42E9270303E739D12817150DA4970DB7A3C996C1E8D6A456317F7421A7"  
00446 "27AC811880F1CBBAC39CF8B95B998E59BC35BBA62BD6FD860B16B1A4D716486F629736CB50"  
00447 "FD124C8B5AF1149F2879F5F05F21012022F1E68ACAB1899BA6BBE41C9398A354BF21C6BC1"  
00448 "EEF80A8DC95A361EBCDAD5E878443FA259E4E8036AA7B8E46888602F4B5B72BEDA6305A8FE"  
00449 "C7B6B285589E9B388116A79D84942C0CE1B4AB2EA18265DD3CC3FFD7B41C3CE9E79D9D702D"  
00450 "497ECDCEEA241FA144547469E5F8993BAD9390B57568A81ACD0F2CC4B7089F2D4C6B16DC"  
00451 "590620E021DA341DBA59DC8DF95F49EE4431FD8820B591414C6A7C651832731B94DA4118C4"  
00452 "198F6285589E9B388116A79D84942C0CE1B4AB2EA18265DD3CC3FFD7B41C3CE9E79D9D702D"  
00453 "29C9A3E643C15B95BADFE9BE6342AF86F5FE52F0A5835ED05097549D99C6848A925576A42"  
00454 "CD54F9BE6E25D1F59DD967E989563A47E428410879EFEC2CA0EF3EFF0E45341C2B02E05C16"  
00455 "760BAC6B25904D1E993509E24936BF2E5EFD7C757258FC662AB3784FC24789EC5F792D8B4B"  
00456 "D2D16570FF25EB95498E56AE80C1B003CAAF9C957E75E08B3BCFD39253546B186C779C41D78"  
00457 "B05F25B4373A1C26F766D8C88DD7FF86BA0E2E7145C5C1341AD33816E02D37F2420FB55978"  
00458 "1F2D900DFAB1F2665826FB346B9EE34607CFDCEC22AC1CB8CB5F12A88C5BEA927F083CF861"  
00459 "15DBD41AFAB4D7ADB5510", // PRN 23  
00460 "9458B1CEAF59FD9721A1D5326146B5E0D33295B5A62E4D00BBAB17F5C538888B085A502BF8"  
00461 "F5323B2EB8E087B7EF7525F92E0CD2BB1A471BD6A868408F61775EA0A51F7B7CE488B3AD8D"  
00462 "34E127D6B7A46FA74340C9D31FD72E48D42B7999B9E71DC5605599E146F494FB3EBACBEF10"  
00463 "558E3C65DD071641F72D3A02FCE12BD3B67CE1FDE526B5716C59E9C8EC73C90CAB964923B"  
00464 "A3F1FAA33C2DD0C27F8688FC515B6DB28C26C456BFDADEB92F8A0098E085B124F2EEC85E3"  
00465 "E05379327D61D9B37752281D20F9CD1681ACB3BCE48B51F32A5E81C2BF833988F69242288F"  
00466 "5AEA01911276F54E432C72351F1092A3C5748F82E106AC97D05FEF53AF26F9A8FF28933974"  
00467 "83D0F1FF064F60664B14DECA1007341349790A8B47B6A8684D8F6900C15FA1A08AE80DA578"  
00468 "164EF90E88835A098C6F4346450DF340B07E59629034B13F45FA4013A5D9B2675ED60C32FF"  
00469 "E809B646AF1094071E9BD7686C1F0C461011B42963670DF1593B6FDEDD2081BDF6B5758968"  
00470 "83D0F1FF064F60664B14DECA1007341349790A8B47B6A8684D8F6900C15FA1A08AE80DA578"  
00471 "663777A54E5ADECB315B76BA8DB4DF945176F4DBF1150D99C6CB2FB51A5D2AEA97D233C7FD"  
00472 "5E80A87A767467E0012C603DA17B90CC406D8341C1D8859C2E4FDDF8BF09F76DBC389DD287"  
00473 "DA04AF833887DB917233817CE393923E9AB5BDD72B1572E6F6CEB7E155A9F55397BA532C62"  
00474 "0C35D6843EE297DFE0DDEF71644418D673F7E688D8A043D2758D0827B0F543D5BA98AE892"  
00475 "40E71CA7717EC4A6CC76E550B2746C78E055B9D324D9875CB5F6A6FA8109A98CE68D025119"  
00476 "FFD817BF4224A4A08912E195DC203084C784FE514621E5B40E4D2A6B8420623DA289225"  
00477 "84F3C75E1B48B915A0770", // PRN 24  
00478 "A38DA6E20CEC2CC1AFA8488C231069AA2E7CD81C45FEA7633EA166397BAD527ED16ED5405A"  
00479 "5372344A29BDA4F3366325AA0E5077B14342857800C8B797C0984DAECA66607C52FFBAFD0"  
00480 "79F2A00E71AC705AE27007F13AF8A4CF9A422A1B86902BD43117E6A397DAE0CD6C0D1C9F29"  
00481 "A27A641B41ED54E65BFA43528A12BD26D95567EF23A8BDD26EED9E4A4C61CF5A655F1AF76"  
00482 "A85993D9DA744EBBE3CDACB353F7E80C155197A4483D51076FDF207AEF957526CC0F82731E"  
00483 "2AE6251875D6EC16B6A69185DF14800CDCC1B4EC9A31488C6857641955DCE7AF20112DD11B"  
00484 "C4A49C37A45C8C3AB05D09A5DF55DDDF2FAA76AE9D0F61A7A53265EB7264F37F266D5EB21"  
00485 "6CBD884DAE5C119B7FEDC103A18967CBF99F9884035D545735101B13842D3AAD7738B06FC2"  
00486 "CDB344896043DEFCFF0BE0F2BA9FA1898B550CC7CAD948A7825D9E6C4845E019EB086C5F19"  
00487 "F52FACCD74A3BFFF9F4B9118B400194CA10CEFC91D466CF9219BF1C9B707B915C381A80BC"  
00488 "7BE858DE2CED2D22D8B2912D03043A66C902241AF16ACF7EBB829840079FE505D39BCE736B"  
00489 "9E148D33927EBE9308BABA5476E38926DC0F9065899D9403D137EEE209131F65940DF42DF0"  
00490 "79FD23A35C2A4BF72EACE7FB1F7A569C3C525986703410739B77487BE171AD20427634AEC3"  
00491 "95FF6547B488010DF6FB60E2C878F22E9AE11B745255EC0004BB59B27C358EE369704F4EEA"  
00492 "9ABEB5E2452DC11AC02E0A6045F8D7F43C25956B49B8CCE2238E30CCD934C0004B5E68139D"  
00493 "436887A8BBD650257870F420FD5FBE4C2A8E284DB47FA4C658FA9607673DB86DF4C9F33954"  
00494 "C1F1DAB5016DCDB2EF30E467FCC9851A597EB7CAE543FB4AE08B0F96B563ECD42CEF362140"  
00495 "E330BD15627BE4AC033E8", // PRN 25  
00496 "98E4C6C9A946COD4CACF1CA4256701ED297270D2B732FF0310069CA84A275A64ECFAC9A1A"  
00497 "441133BBEF6D3F2803E1344008489915287BD8734B20FE4380229EF469206E61EC8B2766D1"  
00498 "0D910095EF0BD15F08B11A16F65EEE1252861E2C2C358BA18A8F93E6BF7774BD2AA61E4BB"  
00499 "D9FA79E21A2AE8501449D2B9FB9F6033DF636B1B09E1B8B9C4C9935C7C09D0325DF878ABA"  
00500 "8672073E04A37A48A6ABE9E354429E60D3E4E022F40B78AF5EB204D32C051D6BEEAC52DA05"  
00501 "12B311959F9275987CCBC81A3257C66A345D7862A0B20F6ED657F28A30CA3FFEB3BFB1545F"  
00502 "GACCB0C4DE2F89A177C4F6F0D2CB545EADF3964368AC2B9DE3B76642783634E06B01A9A6"  
00503 "94519B9C2C3E54A93D74DE31B7D706445B3BFFB50CE89123B9B3DC8D5B9B7F2CF1BB64B"



00504 "0D91393682C3804DE3F44C23F2D275C03BF0D3C61617EBB755B3E104E7A7261F58D95FB64E"  
00505 "9F9D2C6847A8FEB46D88BD0C38FACB70FBAB609F437F7C43E64EB8D8971EA6024E05E55048"  
00506 "1504253D28E1A526C4610AAB5E480E5318A286504962404540F11AAF8F5973E8F5A57D94A8"  
00507 "ED01ADCE79D9DC8CE9802C4486268A6ABFA49A69DDA19F30138DF8FF48C4A835A509E8A0B7"  
00508 "A4AC8AC0B0E9181F187A55E203C3A150B623337CE0530ADDECC807961E352E1BACA97867387"  
00509 "7BDE210914FA399D6C25B366728417B9FA4D72DFAAFCBFD3D24683EFCB517D11E7E0EB415A"  
00510 "68236D11B54A719C21A823D8D5D83B7168B491BDFADE5F56587B970498B3FE2861FEAC6E1"  
00511 "C2294EC5E6B73F574D9610E42772E69D0741EAE23113D301BCB1C33BEE2526E87B02AA8722"  
00512 "1A29F0EBE607D1DB76A18456FD0F93B6338366B7B1B36D7DE67ADE6DDF928EBFBEBB0041D7"  
00513 "754C5C5D5237EB8DB8768", // PRN 26  
00514 "D50D3B57C2BBB2A77424C4879123CA284B8E9F91EC4D976021278756FD90DC8D6890D278AE"  
00515 "BD2123FA72FF16D3A21C177C26C5702E166E12E6E70F2D310BB2961FD3FBA77F4073F4DF4E"  
00516 "CA215CB3788BB645F3DF328F36A1A84FD9AA372EA282DE6EDFD22E51D74153112E440FC5AE"  
00517 "DF230189DE5475D0F8E985972CBFF70959FA6A0F3F1E6DF9712A67FD0A77290467595BC382"  
00518 "C7D40D62B32D2DBB4AE5FD1E76AC30D0C18AD15C4B18058A9961B7BCB9DFABB535D207FA8C"  
00519 "C51623F40B8931A21F46F4CC4D70DE064BC77663FC603CA8FCC47332D01018CC5167B6749A"  
00520 "C6332C1BC19FB34A27C77DF365D218AEDC3D2E7A95980F542289E3966F9269C76D344044A2"  
00521 "3CAA9E9077973D71AA3348F8809F6C94A281EFA50D61B7FF4531A7FAD52268E7F788D6FD2AD"  
00522 "0129E0FE8CEE9C39AC9284425ABED8396EA69C4DEB7C4EA12F24997C625465E122D2778DC7"  
00523 "A8611D3794BA7A5E6BBF936A91EAFCC2089C242E3866FB19E1BBF894402AC50B429E7D128A5"  
00524 "64E46C63B5097EF0A594DCBD215C2AB74FBDDB1E6F3E6007238A36840DB45E61E8EC8AD9C3"  
00525 "B1573A878C0197CAA649490EE9C6BF3E3D8246FFC7D7A2291FC1E7592D066C40C74463D9FD"  
00526 "4A441C103A1AD10E2136E6A276331A067B1CB1861291FF4F54274F25E96316DEEAECB101E"  
00527 "685BB21AB90CB613D8D4C9DD69C16E8E501D19AF6D91E37C04080ED93E75F77A8BD89EB0E"  
00528 "94F99CA83E1A3B63FCED36BB302E17AD569A6A74BCC7E73D2BA8E87BFE309E6E06538FF340"  
00529 "3768DC9B4A6D545C0A7ABD48E72DC11B1D29247EA52144256ADA7864921071E81B2B94A90E"  
00530 "5D0174F3AC5B3D093BE0E65510108383AE1C18BD74B431053541C7DB7E10512B00B94E983B"  
00531 "81842AC63B11749450B34", // PRN 27  
00532 "8A9FA0D16F5292CCCCAC78FC615E47D177FEAE3B3C3593C3D9C873531142F3A8625A7D4865"  
00533 "37CEDDEF32D0BF39AC9BA4A8C40D601EBC7252645C1260CB1C2794F583A7A1E32E50BDE43E"  
00534 "488539F08CC64B96907CD4FC4DE220279DC064EEF6491DD4EC2DBD4B1DD444F0EE6B6B5AB1"  
00535 "388B3F6389283B05E2C7FE940BC72AA458CD3BCD5BA2199E193931136973EA78D1E2A6646B"  
00536 "6000D4C3273F65D5E3CC04A4AFF7AEF33D05C56B24EDAC65B10A3C29B68DFB4996BBBC439B"  
00537 "E0E5CA7B0F2DA8D430F00C726530B582C4EA992A6C0167F19577686125E057DF86DF78BC7"  
00538 "0E2CDE25DA3AA7743B0AD608F714B1152634CBEDD1B22EBE9B4E88F6C9C1BE37CC969B5770"  
00539 "DBA1AD72ED79429D5280A885D6FDDC5EA7F8BB7540D1D175173AD2468A4BDD56316D158462"  
00540 "0AB85542811F9BD724608B8BA4028C82E91907DA5F0EA46620281FC105CCF3C0C676CF3C0C8"  
00541 "A080ABE4F65B8C4CFE20F7898479B5C24929F83CC4DF3A01BE91F27C1F2C6C3A5DD2AFB398"  
00542 "BC74E33F358DC5BD6E8E49876A19BBF0BCC01A1A57250F045D7C219D0481249B40FAFC"  
00543 "38357D972B250942A3EFCFED09FF3DED7D57A9447D14A3FFF7CB69DE6B83B6034E0708677E"  
00544 "9A07F7541E58D30EDA01BE99C5E6CF25528646184D49B3290D135924D251D216236EBE58DF"  
00545 "07CF875052A055991DA4056D564BB96524CA4FA16A06916FA3C8FE1C6D576A8014D126FE5A"  
00546 "103F76DEEDF20C4699361664208066FDE6B6B68F637B92EF08C99F6041C80FAB3C6513BE3F"  
00547 "DACE09E06C498F07A4BDFD709A102C8A823121685441B7D0F68F8F42CEB1BFCF77D9ECE55B"  
00548 "8E09D7256BE5D194B8BCB578EBFD17564EAB1693F25AF06F52FD24051E7C0D4B8771B44017"  
00549 "953AEB4B8F59BE155A008", // PRN 28  
00550 "E0BFF9020D121FB7A7B467748C46D13FB1CC76C6A6984D9D1AE83077626573D9BFBA0D52AD"  
00551 "FEC3EE53432D0BF7617E3FE12176CEFB42685724338D31D69994AB84B6EC07C6E7F8139C7757"  
00552 "3EAC93F1867F57AE8C0AE8099D89F053209B446A07C1370D0F337CA407D7C0D551773F7B04"  
00553 "F2D218D309A452551FB9AA15BF929BEDDDEFF4A8211E5EE581A2A5A7CD49937A98E95BA80"  
00554 "5C5660945503A657C4CAC945F93A744E11ABDBA765BB57E439CEB8D99466B7F268DB2CA8D4"  
00555 "08E952941830403BDC556395B1B469E0EBADA20E6B95DC6CA542FABA68CFD473915A8A4994"  
00556 "D54355BDF8DAAE162E574ACC2D28D01D25B68EA87B8BCC741D47185E44461BC04A0479B35D"  
00557 "29DB3627142B4F1CEA11BF0EB857D2644D17F4AC7008DA74EB8B7373AF6872A80E508EF712"  
00558 "382A5D3C07E6C11932602D896B2A6A806DCDD0BB3E595C7F636D4A8AFBD549CE12B4680EBA"  
00559 "0F34CE10D81908E71D18C5217296B639E5276E713DA8C0D1323B3C0F2610F86127C465B957"  
00560 "2E1F33E960B28EA2F20B7140E672578618FC77F6A5959DA75C4EFC63D3A7DF0F019E171302"  
00561 "DC3E2332CA9B508EDFA9FE1B44027DC6BCB9F0F8887981A29064B503879D56720DCDFE80F2"  
00562 "9D0A4A6B84FBB0712A7E30A2B0A1D707828AEAF3E74820BEFB794B00547E0F47DA978F01F8"  
00563 "B7FEBF486E738A38807116B39025945D6A629639C73729CC6458CEA4432A1207EC792CBD53"  
00564 "10E7481955945463BF049F568AC6C72008F3ABA9EDB11D655FAACE8479BAF73A3335C19D88"  
00565 "299A7D0A6B67670CC0F45CD60F7FD1EE7832037C60BAC3290B34ADDF2423CF1CB1B4442FB30"  
00566 "DF081E56EB137CDCA4C58C6F9DE24104BF0439B7DAF9362FF6817B5243CD96A909953D7CE4"  
00567 "94B06CFB171BDE68BD744", // PRN 29  
00568 "E0092C347E292217BB60B9003170F90C6F4DFB40F1A5BDF35F9873C768A872ADF2C8C1CFA6"  
00569 "0B5F306AE4FEDCDA8FBAAAA5639452401C1E8E9DF02D5E73DFE731B98AC49956A94936FD32"  
00570 "391CAF16ED9CE4D563605A9498EA7CA1E2DD9A18B928B0FD6CE0D923EB4392D0377118C160"  
00571 "BD2E4DC2AC380D40D259560B0AC23C4E6D5C658701FAAC39FFA7EECE177CC89D91FBD849F"  
00572 "400F86B2AF9FF62CDE4152E0BE8132F362A05BCF138FA4312CE98355EFB07AE50ADDE5EB4"  
00573 "BC6053CB76FC12704B39719062A3DC1A236402887C80CA7328A37D4C3982215250AA505F58"  
00574 "6FAEEC18B99407C5A156CBA4C0EFE4A2C8C7FA6CE48690DF2D6A5666FF81A642797EC0893"  
00575 "32CD1C31ABFB25AA134B39F6262CB9ECA326E05A32C6F61771B2B1711462D4C359C4BAF872"  
00576 "3643C9845AE5A4A6A8DD6EA2962482C4A7B3AD4F6605A7BBE7EC7088AF50F15BAAD86C76EA"  
00577 "EC79DB0F2EC6CD16D374F31DDCDD347E9817BF8204756754250A1C0D445C2EF4CE1ED48FA"  
00578 "B988225F5365606DEB2C08FB651812459F3A48EB8A0F7C6987B0FE16D4430766D3030E8C51"  
00579 "5DAE8ACB2DD3FF0035D2C27E5E0DB97BF127916C5AC1B53B6656B99BDBA4E4461BF8319705"  
00580 "CD7160434BC6DAD8E1AE87834B030B2A21BFE91B4F13EF77FC4342372FB91A024EF01783E9"  
00581 "9BE73CC30832D695022F374D41D7B758CA8C49F4116F466929BB7404279C5E42100B9470AF"  
00582 "B757C1D05D877078D50FEF84CAAD49A8C5673C67438F2FF2D96978FBBD78141BF78EB96828"  
00583 "6CC6BC515682682CA8D63EBB9EAEFF166DC4D778F74C929E9042B48A9AB975A074BAE2C"  
00584 "F8C0C6F757255E74B03DCFC8176E2F603BA71E3BDD266741ACF022325800F561A2B2E442957"  
00585 "35CD0C18C18D75436CA34", // PRN 30  
00586 "9C6807B92D255E04EC3FFAEB477D0B1FB8589BFC29DACAC0A771C7C2770CDA8E560CD7D760"  
00587 "EC19F23568C442CF5A4A99DF979C7DC0A80E046DDFD91CDA6BE48F8D5BE408FC46266ED688"  
00588 "E58E57A8E467BF2BE77095AF0F6381341BE16313661A4246D5A63D82D1F5F0C81F3051A152"  
00589 "B17C14324D9E2AA02582F96B4380A652ADAFFEED1667D12E3838E0C36FE2C5B73F5DDC9C90"  
00590 "27176CA06CDA2AF209795D7C959EF2DDF3D1418C9E4E2750F3F10CB14F43AC243A53774AC0"

00591 "11AC998A01706E304A4979C21D7966075300CDB08726130ACE32440FAED229AD95B5A0B7F4"  
00592 "000318C768FDA5724459E29753FA7C359205271FD802551967FC89A303C3F5A9FA78DB000E"  
00593 "CA4115BABA8D0276CCDED2BE4A5D8D5C2A31E6EF3AF3FF3E685BA0E2745A03C4CD6397A6A"  
00594 "FAA9F35D4A79C2F3FFB0CCB8E87B3557F6C2D7DEE600AFDA16C1DD5D02B26BF504540936ED"  
00595 "1298D6A78E0E9C82B917371114069B176A03E3CD08C47819B08C8BB259CDE7707FF97BD47C"  
00596 "17607CCF2B4DC53A9AA0E7376E89B78962A54C8EA1ADB8596853A9B35E77FC63C360315783"  
00597 "6C87831EBD61B31435D8CA0DFE5F8FDD92C1D5F95DE51424D36ABF2C8F61910AE46E74116B"  
00598 "8F83202B32E3496EC9E107210924EC35727A3AB6A483FE6A94435CCB0B6D5C49975A3494BF"  
00599 "3AC0343910AF755F0EF81CA00FC24B6ADB09E55AD5B0C5077FB0BAF13DBCD869913DC2184D"  
00600 "0D774FD4DF36A246F1145BA370BD4B93477B97A6BB0C48ACE4284CE02243449962EF25F267"  
00601 "40B4873EBEDD21C94BDF8C1F01E314E511F46F6A31CC11F54B0A2B45D64ED7D32E61F524B4"  
00602 "AB5D8F52732CCAB3DDDE1ED8C31F800C31CBB92F8A9365CCF985131F99F8CD1E8E37BF228A"  
00603 "8D375384AAEB1A403E180", // PRN 31  
00604 "F99E9EB34B7FD437C0DCB40A389AF3D5B659FA3A0057A7101416656FB21BC92617AED9D8F9"  
00605 "095DF1E562F1D10497906583E294C59B86AA92DFA1D4DC77A816C7301BFEEEA9201C71CA46"  
00606 "97C1642D4C7AF0453E7C3D1AA14E3F6C329A31D9830B22C2373D6EEF06B8F194C14838C6AE"  
00607 "4E0D516DADBE7F26ED76E38B906D1B1428E11B1466048B83669214A7974683350AFD25405B"  
00608 "49A2225768644EF5E9F38C602F1D2ACCF1014F09F7C7F78DA374CC4DF72AABE0515DB7A0C"  
00609 "EF22D7BD60C86B21FCB28603E4BCA9F7FC4BE57BE52DB892ED0A9465907E5AE45BCA4DEA87"  
00610 "D2A6E73DCFF2FAEFB2B77A163569727C898D868336C2F69A4F0635E30303000C63A3A9B0BA7"  
00611 "E6ED712171FE608E34BACEB857E1D314937F9C6238830DCE856BC686BA9C3645DBB022BCC"  
00612 "8F86A4931191262CC192001FC1F5D45C23644A44F9A0BBE8EA95BE4DA4EB4EED4F1E8BF6A3"  
00613 "3515F8C20487D08CBFB14275D3014C5E1CE35EE607A1DE9E849572A6FC93FF60C516CBC"  
00614 "7FCF163DF7A54CDA0720B1B43A658C2A1C42C34B50AAEC27DC9710465DB9C101BF8E2CD43A"  
00615 "BD17ED5E28EAECB03F3A7FFF22F3999F9D2D35912685FA023D6C200A6FD29156AFBB79432F"  
00616 "A2FF3847DF7D1F64417B910B82F52AC1DACF91E78B4395EA5AB4D3716D98B2160C3A3064FA"  
00617 "49A302671863E881A96CCD65D5816B2B64A7B2C6A09D906CDD85BC799EE5A433496BFF9A43"  
00618 "2D5429F539C74AEF39441049E651B59E45A11A751CDE8911030C9444DD4F0E1E60B9CD4488"  
00619 "E405E31A641BD82D5CC8532C3BA01551F57BF4496DFBAA0BA413A2E2EC8B038875E8BFE7DD"  
00620 "4E707C72D539A403A011EC5DCDC9760E3EEC518644D68E529AC88A599217CA5DAEEEE04714"  
00621 "2B21F6508749BCB6DD91C", // PRN 32  
00622 "8EF641023FD4C1660E83C65765C1E7C41D8C5EB1EB85A14F1AC9939D8E8C656450D07FBAB1"  
00623 "FFBA181113DA059E2D53EDADA800C04BFD23F44C144B2BA68A53C66E3DEFA76EDBBA679085"  
00624 "D0DCB2479A8DB56B76B880EFBFE7942ED642FBB8701F6E425E87D5CF36A269F746FA66ECF"  
00625 "1CB1583E8BAEF36AA90ADAB08E9DD296529AC25DDC6BE31E7C9C0E0C3EB8D0CEFF6ACD7AF"  
00626 "9DF6158CE75B382D2CD35F155149DCABC4B1C03935F8105CBA0C0AC84CE83B276C29F1CBF1"  
00627 "319588634F92F412F8BAA7B7D179F8485034E6997E9DD8DF476FBE851A305C59A5082B48E"  
00628 "BA6BAC334F0E7F6C64626870495E13EBEDC01C4FF544B67CA08FA319AEC58E839CEC80EDA"  
00629 "2CEA9CB7925E1600AFAD4A469D89AFC6EDDFC0498CF02C9BDFB927C99EB6BB4E22FE27"  
00630 "51A272856AAA5F4C13707427FD70ED68F2817D68A632006C1C08F3C637A2ABEA9775F35991"  
00631 "30D1A500B4E4AB968E6A72B8B21061758A8F80267B55F64E7D79898A71C435B9FF2CE9E43D"  
00632 "AAAGFF1A335B15838E7353DF6B73482D03D3A8C33C11B2A2061B7ADB60CB2F7135D73EB3EB"  
00633 "67E0A34994CE3FE985EACAF6725D83649012E1817CBF06885DBC4B320F8A541215AC2A99"  
00634 "312C16C45882914A33718B61A0F04B125D935AC59A59E76F760E55E6043FF9A7BFA2CA9B8A"  
00635 "1E077C632402AD6D9A949C5B46DE2D82E4012D0520365DC05211698CBD3E7C3426BCD2AD43"  
00636 "A1BFA1ECE0130D05C734A129A3C9A5397788B49329922F3B3860AB2C3B8C62235F4CAFD7E9"  
00637 "0780C37A57073FE00D941819D6C1DD347AC32C906C2B7E5FA0A74B40FC7B797E7C041CC2B4"  
00638 "486B57642D3B613BB68921CD9D78390242C211EE459556645D0369A7B76371C2B50F1602A"  
00639 "80CCC833FEB49EBC969B8", // PRN 33  
00640 "EB1EF5FED3E30E84B5745A5DEBF56482294D82C717C7DD60BA44EADA63C0F4391D0D1E1EC2"  
00641 "938133F5F637B36BCDF69AA8E9E458D56B0BEC410C1C736DD5D81C98A91A11D3BC43598E1B"  
00642 "E12061C96C9E5669D9689DE1220BE22F4B6545ED59E8DBB547F9B7A20465F3CDA2F56DFC6"  
00643 "D22E4049659609B99364C74501EA3D247D223EA6B7C29D8412AA922F6E638DF61C70B81FF0D"  
00644 "A0DF4FED3740AD6D8489E4785D60620224F8E53C251B28CAFC8EA978E304947850D220667948"  
00645 "C880E769359628710410CBB7C65FD59BFF58E7CC020E11CB72BA184C10C6CC07CAD27A6234"  
00646 "BE29F5614680C7EC9CF0E39411D65B228EF6A1FBA54E12576599D74B1C7DA858F7BA89C29C"  
00647 "BDDF4FACAE05B64963364C66EB88C955A345739D74944628EA55A5A0E769DF1718F0899BB"  
00648 "DEA90C56EE7D54C5E297605C347A88F1B01C21233EA56E42E623C47EAE9B1C20C0CFFB015"  
00649 "988945C65D8729D831D9EABFE70E494B7462B096B07779A72F59DD003E4BCD0F81E4C4BC41"  
00650 "0AD855A826EA58002DAFE56B4309E33CA6722CEA7C94E76FA814AC580342746BF446FBBD7E"  
00651 "081AFFFB4825C9448D5371D25085EBCFF0470B5EA7109E4A84BF64855FEA7EE84D65D7B80"  
00652 "D327C09C1E2A22939A211B159A573B881E0B6BE5C28A89AB5C1A9B814FADEBDE7A9B9B1EEB"  
00653 "2D32BF9EB27EED64C1DFB100DEC107D1C779FB7416F48E72202497CDB111D74039C58F2F1D"  
00654 "39C619376DB32B9DE280EE95839A952DAA4D130BC6BAF3B61FC963F8451215194E820A69DA"  
00655 "77E41F6530B9D5E801EC49990AC5D2D00A9FD24E663F8896C328F6389891C5197CA84EA373"  
00656 "8499E1A99BD5E5EFC974E263BA05393910F452324C7217B73DF5AB29060BCDDB7B5C1E9B6F"  
00657 "207F7084BAEB56F483C24", // PRN 34  
00658 "A248DFC9DD738A7E5AC3D93E93387534AE2B707B5036BE6935C38A62E7605DE427332FAB29"  
00659 "E251CD5543DB8A7938EE2967EEAB3CC7336FE0C64AEBD4E8E444A156FEBE6F5B6D97F08BFF"  
00660 "1C37AE4A0F94A253A673D1638DD6010416BC0DA9E67F6E8CA6C1C0D5AAC8FA44798E72FBBB"  
00661 "713DA01BB57200F17C641574E5C9A511DE9CB0E5825EE180F3B198B1D677F697AEB668C20B"  
00662 "68B55580B18A14928C9841F86920A5E2625FEE4C5E16A794B354D889CC5EB8BA85419FE6A2"  
00663 "B47AFBA35E1AF3D90F56176F330EDB45FA91F3328351DBEE08D79ECBCFC0CC32D57CDEF9A2E"  
00664 "823C58DD76AEDDBA3B73A05A3FE8011C68BB6C452BF290E3AB09788E5815F3AC0DF9BD29B6"  
00665 "4110DA7E215517642290B91983C0A0B0FD8923D5E80A5CCA9F79F2A0BB1A268740ABA3C8410"  
00666 "F3823B2838F31F25953F41B8EDCC93A71946B56EC0B08909D101C2E1D417ECEBF06E220114"  
00667 "952B2704D6D2967280393679EF76381DB071E04D86674CD340FA2A798DA90ED12D6FDEBFB"  
00668 "C92C018F6B71B09E87C991D1BA7425DCB10E988FA14C76B1F4D30C605414A40C3EBAF1C38A"  
00669 "1D95A30F26EEA24900880F3932772B0333F5432039E8AF2D0974EE30A733DBC9CA1A4F62AA"  
00670 "BCB23248E0EFAA97CE93DC0A0018E3BF6972A7CEA1AC941D168AE6C071D1D4A7B532F8074"  
00671 "3F9A5E2DBEBA6920B25E1C93C4828FFB4E6AFBE8ECFDA7D3D2078F46CBF3031723DD00A891"  
00672 "41FA5B712CF5778704790526315F0018DCF8FAED79FD18C34F42D820745AE480F32B9E69A7"  
00673 "1451960276AE39C32CECEADA41F6B5033686FDA44CA1F571E9CAD1F5BE7335C16C66134F4C"  
00674 "8A20BBFBFE05BB26790E85C519141757C8125CF08AD6B24AF41CCC5D73B4DD2050DBA2F884"  
00675 "E1E857CBB36858754F638", // PRN 35  
00676 "B7C32F27716B8A3100AE52B060043B639157D6B277069C9E90F4DE18A25FD568595AF67C"  
00677 "EBEDD29E898F8445B73BA0EC2FCE412C550BE56206F9F118FFE2187A5CF703F17610A38D62"

```
00678 "FE1B04632A156E65462751EAB89B6F6B98CE0A3386ECDF62C5B6DFC02BDA8245A127AC9AEF"
00679 "CF7C4588CD0704A07ECCE68B34F419B03B9107D4D8CE7E6D6EE45EB3480C08A5B8FE584E6C"
00680 "CED489DB46799107888AB5F4E49492E2EEC223BF646BC4363F45447A06328DFAD05C68EEB2"
00681 "A4F4ADFF68083C7E7DB43FE8610DF167ADAA07BF14BDDFF009268A87174A96B568EE0E3B15B"
00682 "C86850EBFA3C08D3559A41BA195289F3AFBECA053B22AE37F3DB19D7C0459E009D63D9203"
00683 "88D5218AEE6D11FE15E8CDF744807F8B1EF95E21B06338B628F32BA4174B0FA7589CF31A84"
00684 "74C9C51BD3DA8FBD92B831B9C849F4B75A0A6ADBA6C2C911C8FE5F7AED582F7D53F2BFF5D7"
00685 "063BD8A6509D2A4DE67EE3CD955DB4F2F8C268B25C4FA36ACD175BB8E718C17E4538BD82F8"
00686 "F6615FE24DB816D8C53958498613338484D85875287C141D74A698464DA0CF9AC4D00718D0"
00687 "2B4F59D12BDD149E6747D750949C7D150FB6823BB26DC33C1B90CA43218E24EA802A701909"
00688 "76686C088CD60794A43A9035142C698FCEEA9E91FB6BC6D678EBDC6293A9285553CF8D7B51"
00689 "075DEF99A172D2CF0444AC8B16875D5969DB07B7F1EF6EEEC9AE4DD2B18908560FB49BBF"
00690 "1151D6105FB7A54772244DC0CB1034F03A89D634C1E932D0479A1A31147D6FC4EA301682A"
00691 "888F406336AF4518714D0AB746CC79DA94E2F33FC5EE8457A16A4175B0A3FCD07B897CA0E3"
00692 "E0077131359B9BBF62CA363EA01D4C2AD5DF84CDF06EEAE2F67DA1E56E46027BCE48C90753"
00693 "6474982F3F341E90333C8", // PRN 36
00694 "8279B8CB078D12AA941FA064F72C2C727C545A653BAF5C0DB1F4BF1819AEDC4AE09A5B844A"
00695 "49692BF122BA0C0B839C6283B66865E88D41858CDBB705F1552C5083D9433B98723FE914E0"
00696 "C23ACFC1A2E9121AB2CBFD47474268A7E0894C9DF8AE4DE368A4EBB7CE9A82AFD5E56F5702"
00697 "2DEB8F9CEE4A1F7EF01865C7D02C9AC3125197D073666355C05247B7CFDDBCDFD37C232238E1"
00698 "5DEDE0734FBD5FC95C8425FE474F05F3E0BB64BCE72A1BA583EBD70C2C45FC883B630C86F"
00699 "16DC895CB5C64B134A5A9345549350FF529762068ADF5BF3086379583805AFF7145BEB9B0"
00700 "9AEF802D63A96AAE4BC148702465D89368AE54F6517C44AE557B74A6A83B332924C4363A3"
00701 "E93E2035C1C4706011042653CC7A6F41DA2EDEABAC3AE3698EC18D7E081F2FF70F325B36E5"
00702 "C3C46B8D69D2A5AC5C8E6F6F325F9917BC4CCC03D510C4CEF3BE7E0356BB168F57CDDA940"
00703 "1DB138E388D3073D2E12B1B300972F16B2A41DEA9A3BC0841C9CD754606252958AF820C6A"
00704 "9DB4813CF67A7AFBBA3CDDA285A9E284D27D263D22345DB22D0334AAA630FF88597474AC86"
00705 "7BED5FEA84C08A44422BA1DC92BA27368B634DEDD198313A06EEE3DD9E1C90C4955B4CDA5"
00706 "999BE46D9E6B3ECA9C0227442D5AC2D178ED9CA97E286D58FB191080318139AE4332F4EE5F"
00707 "D077561BB2FC5A35F548544340FBE617601BB7F35855EDB147382BD22D6B8FC84995BF60D0"
00708 "5D98D7D109F87A4DFEC1379307986508BB90DBDF29FBB19DA552C8F41C28871DD4007576B0"
00709 "157E5BFA13BC1C7F694763C3DB61FD02032D31A3BE7162341C7D0D4A96AE56125865090560"
00710 "D9F82611EFF34A0835A1365E3FAB2B802DFD358681D7FFF1B7E1A38670F1535AF0D1831CD4"
00711 "CA1ACA1CEE8D8B158B30", // PRN 37
00712 "DCF8B1B7E674522419627D5AD4AEBF6C6447F9D38A4BF29C12DBC07DC2185232B1D0BD7189"
00713 "BF7B08AD8DA2517524A597EFAD7D331BE4661968E1F8C43FD2A094B42BE350128FC4C1C4A4"
00714 "28CBF88AE39E129463A2177FC1DEF2AFA62C95F909AE17DCF4032329F38AAE29AB13FEBD3"
00715 "8FD3DA208CB395BD476F33BD8C2FCAB7F1D2450B002A6154E110AA2FBBB2C379531690C111"
00716 "3C51CB5D4175FA6925530C0E88536060928670BFF8DDF90154F60D2615E7E87D6224C19015"
00717 "3FEE0434C6949D82EDF633780F82836993E4D839B400372EAE98EBE21349AD245728977A6"
00718 "C78B358467769AF9BEF0168A7126B000331D28ED1316AF7EE798278EBCAD7847A7D72C6867"
00719 "9DB4813CF67A7AFBBA3CDDA285A9E284D27D263D22345DB22D0334AAA630FF88597474AC86"
00720 "7B0DFD43A47C461270EFA09C971C000EC353ED5A408C9F442CE32FB0181A18A5D5072A5BF"
00721 "621FEC23BDAEB0649BE3309B8A208A61B29512D823C5A066E4B2DFCC6F53FA2FE4BBE14B6"
00722 "3A9E0368A7E23FEA409A930A1FA3ED41DCE2AF953EE06E7BC5EB3EF4E479CC144F6F8EA123"
00723 "8EC7B0EE0216F6CE75110BDC4B8224B08466C5FD8B0FBD7CB52DD9F92A3ACF9AED1F265A69"
00724 "0683374EDED1BDB8CE9F6BB9C8B12D7C2E4ABA5C48EE6B1491BE66BD1A485CBF07EEB932"
00725 "A0CC24482E38D58A9A20CE3B225B4D1E66695418FF20ED2F6098523CECE06B5615DB4ADA0"
00726 "DF999D717A3229895A9EC0288035D2CB5307D5B9B27652DD2EBDECE8531074B828C6BE1524"
00727 "2B2C41674CF5BEF9B1B2E5E05DD93F398737DD1E6BAD21A6552C7C5C376A92A89EBD5A11F4"
00728 "ACEADCA39256DBDFB1CC0FF7D1207D0F5C0579B39A32899A05C4A7C5E6240434D017441C"
00729 "B703101DE3862122C9E44", // PRN 38
00730 "DD5F5D29A86DDCBA09CC373F11C14747D12C01974E74CC7C06BDDCD5F2D45256FD6964D932"
00731 "87BBDCEB414D74C6EF98F1CD72111DC32815E03E13B2B6F85785E5871264BEAD6F8BAAF9C9"
00732 "7E7650EC187BECBFF59925258A84A671791C33619FE983EC5868FE5C377D91C690BEFAC9F1"
00733 "9AE1FD9B5C4E2E0D0C52A2AF3F631C5A80BFDDC16EAE7ED73F66A69791E8DD2815848D6BDB"
00734 "4E747D6DC08FA6B606876C3D1A7AC9E3344DE02744081E0C0CB522B091439F6263D40E7DBAA"
00735 "159D31BF49BAAD94BB974C273E037E49EF6826C8E1142B768A2E606F3A76204523024B0D6B"
00736 "E9B3A19032A5B877590E2EDFEED880BAE7E22E439855D2B0217974A6D7C6F80C3B0D232ED4"
00737 "96F931E5E8A43FF6BAD1AB581EE9723DD70C0A794C25281CC55207EE31D34C91A1A4B74D73"
00738 "6A0859E62B25A93DB39998D26A1EAB3E4143F01B664858183F23238A26EE2047FD1FBF526C"
00739 "D6E3CB6017820E8F83495B4DD93B8D70DFF1C06AFCAF11E81E904B37F717A98DE354D45465"
00740 "741BA1B4F8E72A612582A2A1F2E40A05727E147238152128F03FE0F13AD4B3CDB2699CCCC"
00741 "D2FFC2D6509C01FD4B8E7AF42D3F2DC2F7F0FC6230836BC2661F1CE7838D636D7A931C62E"
00742 "891F23A37FC7642BC94438BEB34C28AE6F000C1C334C41D5C0E29F9DA1C0FCB5F8FAF77B78"
00743 "684170A26962015E86761F17AE5D621B35C8D92EAA6BAE8C1A30B356386B99693A725BAB3C3"
00744 "574E265EA4B819502B886620D4D10DE43ECD300CBBB5CEE96F5CAE0567BFA11E4AC8CC1288"
00745 "A1A90E6B19C8128402C6B09A8189917AE49D14AF54F7AA9314F0423649DC7A575E731C809"
00746 "9ADE2AB3EEAE1F8F4CC677F920B664AF49155824C08A54DA3179C436D52536FC0B186B521"
00747 "2FB36AB29765CCA6BFD5C", // PRN 39
00748 "87F0F71E090A523A05795BB3946AAE7225D4123530FE4930BA0570C51495C6E8C674AD68A0"
00749 "B1F9DF470C752F897677332299D7875F4969820512161FCD3E6183AB96E296DD58DD06F3D9"
00750 "FC28D934545E5DE6E667E41D23383F68FAEFB13139D1822AB1A50708E4598E0B877D08E004"
00751 "A75C75268FCA08BAC51532F3332CCE45DE81F112CD07ED277D1407AABFFA852A21BC1B11FC"
00752 "F906C8A24CE101E048606279EF6CFDFF73C6E69CF6C26C4FCC3DECE232FA9900B5D16AF9E7"
00753 "6F54F35718583F746CE3ACD0D9DCDC3495C63CFABB2D8FD15845D7FA1BE3E1E04B403F5572"
00754 "F745EE74B0B3D6E912EC7C9FA465B14FD1CDDCAC6D35DCAF09ABCBB78B7752B17382046D94"
00755 "EAD4EEFC56FA0B92DE2B690801C16117B8A1772D9634E13AB0314FFF209DBDAA17BF512FE6"
00756 "BBB745641BF8799C1EA745264A2E9AF3F5EA28F01449E2D65678513806A3CD198CAC919971"
00757 "2FB577DE5376BE375018547D4D143B4C7D87705604DF1335EF12029CB54D06692A8EEF5D51"
00758 "9A96F6A448B32D663EABBC6EE600244137847F367C12C245E06F3D963D00CEE03A308EDA0A"
00759 "D072E4F5778F74DF945B011D76801A5599FDE954E257CFB993E2E15A8323370E6424A82668E"
00760 "4EB080708FBD239416EAE3D23535F198498A8CA3C630A864503716DE942937BA462B80CD2FC"
00761 "6E0294CD708E85F39D25377FF63C43147629876C6192EA1C34ACE520D7FB71AD03716641CF"
00762 "901206491096125B0E33252E7172BE6EDFB583E6034F3A72FD01D9C8D091F78BE4975577A6"
00763 "006128AEAE8DBD7E5AD97A7DEF7F51942280E2E9A5C880C7E871CAD8579A6691B44660784"
00764 "8D48E9583FF341E9B6CD5D4A128CCDF534E9666E4EBB929EECC03408A5F95276B65C0996"
```

```
00765 "2C2A1DC2585B617C527F0", // PRN 40
00766 "82A6C8E99391CCEDA8359F2DC980A844B26BEC60C7312CB540C3196433C05DDBBD2B6D43CD"
00767 "89D91195B9AE50B5382269A50B63EC8D3C32354EDBF22C4988CE66E76CC63D97C9EEF15C1F"
00768 "0714035ACC70F809BB82AC0E33C47F672BD54B133E160EC5C0B3857A69FFC00968650455D"
00769 "7D99A689C9071929DDBC1C552623187D6F5A7F1E7E7A4CAE0571DBEE0CA1120184E545DCE5"
00770 "4216227F485344FC12FB10850DF040B5DE5B5D7FF45A414E1E4BC61AE8BE695CB805B0FB8"
00771 "48D39B12A61428A53D5F610DF4D62101D22E4BDFD1ACA5B06B6A6725364812F963C81300A1"
00772 "9414AC98C7043C3D96FED8577A069FCBB2CACCA0CA2B28537834FAB53887CAD2C9BC10761A"
00773 "132C12893012F48671FE28702497BC2A39A3A5BC0E866A73E53FE2FB90F1749A3050B9554E"
00774 "D499BCF279DC427CE6499FADD5AA64C5964E4C676F9749927145244E8CC1A71BCCF4BB6236"
00775 "6BC9CD2DB7D2E3FB4A56AD4E74CDC48247A3014CFD376B2E650325CBD7C6ED016A4941EAF"
00776 "71FDB91C2809E9E01A728C9F48851BBC4EADABB23FB3094CF69D08146B2305116D68A591DF"
00777 "11362E22D537323D0BF90DDA00B6BAB7B0A200FFD7BD0AE2B12B9A99D581CBF43F9EB06A8B"
00778 "F59885A36CB112DA1F78D2513B313951088A6AFA2255B6FA69B40CD9C8BFE0BDB837CF9550"
00779 "44FA5621FB3030FA6E04FCDFFE2B3DAADA5BC4F77F71C780352CC7AA7D051F663A13C45C6443"
00780 "3832336F0A9AC78A62C0077754489B6253EFA75EC6DEAF2A9EAF89FCDFA28AA9D43E39E25"
00781 "FC0FD3B942F756C5D3BB437EF7C9ABF30FB4DA227BFF021E1CF783FF1B8BC7BE2FF6553FC"
00782 "84FB9A93E15CBE07D503D1CFD00B4D8154BCB075EC100CC635B6615DF5FA558F4D58E0EA9"
00783 "8200B97C9CEB1C80CA850", // PRN 41
00784 "EAE3C7B357368A15F87B0872EB86194169108BB59B4BFC6C8BE058C57FCE77C99502266EDD"
00785 "2DF7335DC5BB3DACB0C0A042BF0F46B3773487F811B27103B791F7099F58F43FC82E682AA9"
00786 "09EDA6A9F8BE4669D3185A704FDB3FC9C3E312612C3A2DF427C120AE3BB5FED4634E98E7EC"
00787 "9EAC66E317773FDA171B52E139336C70675CD65FEFB6812726C075FC9D1F822351B4E2234"
00788 "A754ED28B3FB3541AC0C26306AEC93DECEFF769AD8BD106A484FAC54D0AE027711D7E50096"
00789 "CE24D2DD2015C50611AF4EADFF16BD8BCFC7B9989EC1235FB1F7E6A72AF7A8AFCBF2321306"
00790 "702D3DAE840127F31926D71764B771CA482C7ADBC4578C5A0919DC5E7F042CE2549B3EA3E7"
00791 "D68FA12716695F765AC2950BA62602F579412D515C24D7DCA9E66B6AF120B94863C9B701AA"
00792 "D75EB3AF04BD7D32F330946AADB23F505953F2B5370CDD277736CA17E0FD4457934A2CE132"
00793 "FADB6A603D0EB3706629B1EB1407F794C2018B0C70074BE4061C909D5D760AD42FDC10089C"
00794 "D94F947B4CADB2263636BC2964884702DFD17A750B5C1129A120312EDDE0FA430E2AD5134"
00795 "5ACC5F6DF2467EDA1875868DE3FFC9988F1536D0124ECC339EA2FB3C6A4D8A81710CB6D6263"
00796 "C6617928A21729B2B983BEF6E0904CE678902E41C2DE8810F2DD02021BABB046D4559566E1"
00797 "82B029974573F5AA03E061D3A93220D626FB6A3C3EEB4F4C92680CBE244071F4A800958F0F"
00798 "3A0A1B02D0D1E78ED5FD88EF518FFC9903EF838133CC4A5687BFA9B3F4D74DA8B0BB74ADA0"
00799 "5C276E66BA76EFF8D255F667C4F007F70A7627AB6A9A43F18472FBAA698681E9F2E2B99B8F"
00800 "500D6B5848F5C7E5A3E915A9A927AFB240B47DEE3375731611938905202C0D19A4BB1848068"
00801 "93C5A1EC3C2DA490F0AC4", // PRN 42
00802 "81D7FDADDA0B9157C00450C3DA7DAE5E0CBCBCEE93039EF567E81961E115067C80BE548326"
00803 "1DC0F0FE08A5C1EA39CDE8B88C04F34225B7E4AB0A2DC241A72B9A27117A7525320C08ED62C"
00804 "D6E9B8C1A4E287B12CBE6503376C9EFC9B2267663E7DEB56E55B54D2E91623EFD3C2C81D85"
00805 "3DE51E73260B3FE1AF8DE75D8A9D0D418496C3F8D7A73317964ADFE9A0D45A5EFFDB81D7D7"
00806 "D5DBF582CDAF815DA4B7DE87A2C28FA6D8311F0909D6C6FE2D06AEA7F98E4DA3581F0F2F47F"
00807 "328608E6D84EA2F9F2A5A4FF121BF8D1522C90D2D5F3CEF9E3781EB5D4E385E9A8C58530A7"
00808 "9FE70E1399EFD0BA4FE93758BA67422B037AB39F4DD20FA848D5B255511F2F820C2B5B1187"
00809 "57B09DD262E100D295E5B74113D5DA75FFB7049E5E5D32C5C3CDA6B2BA503F48C82B95A76"
00810 "D9184ED3E915722261D6A1F004DA84DD274B556B17CD1FFB158B22C393CC06A7D6720ABEE0"
00811 "861021B98247F4A06FC04A28CCCEFEF7FBCE802E083410A646B29A37B732B99E2F90C1CEBE"
00812 "BA828B2AC07E8AA9C32038D8C7C7CD824ECEBC834DAEC4C80D303329D5F5389ECB4599380E"
00813 "A68EE5D5601EB212D4ED327B75C407F3A0C0E7B57820419F377288D4F04F4E53458931A51B"
00814 "EAA87CD3725E6958F03CA9E6C7B8598E445F4CE28379A76EA006DF773ECA22C9597501A5B1"
00815 "CB509C04D9375A966B7BC35CF5C575A4449F1A307E4CE64BF01566A7D4CED5BA410687B43"
00816 "E3D35521D03054C37B1E7D14E664C5442E1100F8DB7477270E39FB30BE495FE254CF3E2C3"
00817 "FF133848BD75846EAA412779AD1C8455A7A5F1C951230826DD80165C4EB4BA8826E09BAB44"
00818 "6CAB6949397D7A88C3152A4D25E1540C716E669A5B0D70B1EE558836D8C0F431981F019B6B"
00819 "6724F4B7FD17086D81E48", // PRN 43
00820 "B67814C98514A451ED8FF4C8FA70190D6087809301EB8D5D17BA587BDB954D631907779697"
00821 "28C625B11A216C51E796BE9886EDEB000C727709151169EA8EC67CE997F748DB608E2D61"
00822 "3140A787B579E31A5F9E92994A3C9DC21CBA91AA6B6742E0332A9C538D366DF211A7219864"
00823 "53439CF1BA8D49D940A2A04293927B1094CC9EE1E33CBCE7678A676E33F7CD5DE8D93D1C0D"
00824 "5AF9BB9D6F7A51FF9EC56B80D83208FE71B0C85F3BFA60ED013BA0191E07D2B15AEF426A71"
00825 "22D3E754AE749B17E6DB43C772A71D7ED7645FF24760F83F2AD6448A7A14C2A2472032BA36"
00826 "B3238C5AABEADEF6240CBE68606954B60051D393A3BB014B0D5386097A02399E5BE3866670"
00827 "86B1541D939EF67620B6D83EB004C6C71FBC6EB3DA8EC6E50A6B05D7BD4CDA52CF441AE032"
00828 "E693C3C297E0C350CFE95951DAF2E1E56B04C232B4765A2D0B4E5264BA5B8B4A3563EFEC09"
00829 "8B9F01A78B5C6F146B828E47BACBE4117CACDF2B2C882C01F787BF52A30476881EF03A47DD"
00830 "63F98804CD8BF8A77549F4CF893B1E5CEE4F2A7843AD78145AF3F186DD7F0EDFCB5AD04D59"
00831 "8095A357CB7FCF10FEF6E9B891D1AF98803A5DCB68AEC494B684054C13C8EE429458A0E90A"
00832 "126EC823C02B7E7509F5F344C679B670B69E7D290A7853EBDF9304A66E7AFA4A16FA412F3F"
00833 "F810BBA9A75DA1A3E3EF8BBAAE081B76D5C935132497112E36501554B017795D5FF886581"
00834 "7154C11DEF9F42ED0401BE65E06018377796B29E163DB4B800A874052E2839AC97970CD38"
00835 "ADF8834C56FA78DE936A366B931DDE81992AC1A8ED5DD54CE7FCD8F7578CC725DDE4C8ADCC"
00836 "DB3DAED876946D166107F088C5B48AF08CFD419A4A0E0B9DBBC3A4A3BD4C5719658BF28BD"
00837 "B6D036E82D02EF9822208", // PRN 44
00838 "EABABFF30EFC6AE2E63A7F47BBC4728B98F66463D3781C97BCFA1C6308F1BF7A47860A41B"
00839 "0601E35C49CC72C7C1069DABF2D0FA2DE88F83FEFA369C447AC31483890EAAA9B3FE321CAD"
00840 "3421D8DB9AC305102A9393848098482BA907E816BA6BFBC386D99191DF8E4DE74919DAD8B2"
00841 "7A303DA734499525E036C3B7FDEDBF1719BB6D48A385B055867EF3D9B4C10D9F1A08558AF"
00842 "D7ADD124C31AE9387EF72B3AE22CCD91B8964DBDE71E142D399B9E0FEEBB86A219D6E04278"
00843 "6A1573F495E5469401961E9AF48CAC31C55561B75BE83245F7CA56EDE68AD2A9CB3C04F920"
00844 "6526D886E2B45AB8806728309D13C23A2F8D15D291DCF7F0A84D11301999E0C545A06A5628"
00845 "47BF179FC3FB40401C8D20D479BA89A595C6CCF591E0828A457BB3DC0EDE6648B516788582"
00846 "E55C22D8F8A84CE4CDE3C749CFF7F8F8540CA8A2A1B472FE149313DC5F4EF7647D5228CF13"
00847 "CC8414B6FB85F5AF69173D30E4CF90310A7F2070E0A6AE8A425FA381B8333EEDC45B3B887A"
00848 "9AD687E3FD5A9413A68BDC144193D9AB0F1A10299AC0290C7532A5B61BB8EFA1E48F188AC8"
00849 "83072AFDA5A4E0D494E7CCF900FC751151691A06D29486F2CBB984AD078910C6D39BFF75C5F"
00850 "481C15733FA0EFED29278BE0517AA25240F8C736A04F11FF68A1077C36E54B0CA2617211DB"
00851 "B5A819290C2624D6CB778767654FF61C4B820CDA94F4C7F59F612D6BE3BFD1119560C8715F"
```



00852 "A50B0A5EB921AE8B9E15BE5F6903D7B1B607575DA92CA93B3388916D9843351F33E97B5EBA"  
00853 "DC4B30FFC8852DD6EAB0F63352EA7573335EBDCC31FAB5CD988F3F8D62B5E32B8EDA81BAB6"  
00854 "5646158BA57FA37237F575C101450C223A734CDC13262C4FD4E8914519FAEA599F5DD3A17C"  
00855 "44727A8DE22320B2F91F4", // PRN 45  
00856 "998031E003AF13E73A983242CF8102737EAA04F5C475FE94EDEE67570C1794910266F096F2"  
00857 "4B02A1C9DA33B7C07442A49FBD6BDCB5855665529CB1B4056C159AD1D8D2C0E9861DDF78A"  
00858 "DFF68A8FC35A1BC798C1246C5B15B1E9820F1F960D24017250B0781480127733FF1B67FC30"  
00859 "A6CEBA8C66A8DBF09ED2225AC25A790F1080791A93D73CE1793A0E4070E326E8071F1E5C3C"  
00860 "A70FC475248DD1A4F623413E1DF6154128B9E9CEE4361FCC269C32A54BDFB1F2AD1E7BBEC7"  
00861 "0DB73CF27014825BB5A18BF924D1DBDD6C7C929DBCE3E46DA1FA260245BAD192A7B56A703E"  
00862 "C22E51DC2F58D00EB249638571E2EEE1EF396C04E740D9F5D44AABB160EF2DAFA01B725A6B"  
00863 "45478FB55CBBC24AF4E4AFFCB5671658DB913B43B134C68A291A4C876CBC811BC05D68B2C0"  
00864 "73DD510C26E490B9EAD5F050401D16BA7405E570D7F2DD874865492603959B578153AAB478"  
00865 "AC4FE885E1136B62C947BB0C867C6E8B87A4D9C38C143A13C7E3B954DC1B66BF3714512F46"  
00866 "B6E4EF9DBE4CFDF5A7927B33A9E56CCBD56591593905BCC7F4419901281B14F01D4CF56337"  
00867 "FA90613893F12E7BA7731ED016C1FBD414BB33E5DFB9DD2B1761733CF417320B1D568C32B7"  
00868 "F7BA02EB59183AEBF5218C6BA3895133E7E9E80F8D25BB0B2AE72F0294CFA3181388633642"  
00869 "E487BFD4631F2C81F2593D10127714D2493869E81AC9432AF5FE57F71B2C2AD813A55028C3"  
00870 "8D2D6AD9A26D9FED2064214F294E0AF3514535F56F7F659B814B7ACC8D7D13EEEE5777C971"  
00871 "D94C657F8888DD478A3565530F7DDD4232D21A4A25D5D36214D68417FBF893ECD442FF657"  
00872 "8B0F702F1BC07349E873F4F552F48926BAC6282FCF3998161035762F38C93C4108C7BA43D8"  
00873 "5F76A78720A5D97865128", // PRN 46  
00874 "D58F19B469D3B8182F9EBABAB5E2B3A2E7BB74601BAA44FBB77AADD78C9366DDEA04F21AD2"  
00875 "8FB0E71DEE8A169D41116E63F56DD3ED0E07FA94DDA8ADE454980BB900E8DF6FFE21409B5"  
00876 "82F5AFD684B5DEF4A875C3ECCFD18C4B27A7A80F3FD6480BD09C738E0040C234FA1F5881C9"  
00877 "70E167000169345C587D2A7817FEF71C1CF9F173924F39F3D76107A526358FC2520A59DC75"  
00878 "3DA8AB9FE7E5F339888BD797B43A5070E8E33BA0C9259614B7B89D69F4F541A5BF9B2BCF4"  
00879 "F58431FE2B0D6809804C005C43D3308A97308B3B9DF6FF1150AB170AA54261C088C2A6C6CB"  
00880 "E26D3B7B6703E3A565B4DBA21A09813F0C19EC8F801A34ED9F91A6114D465BB7E9E04FC15A"  
00881 "369654D15D78E125BCF43E2A16FDD875366806448F9B10C6F176F39A9B4ED27D2AF8FDCCA6"  
00882 "8DA95E96B2342902D9FA7A07F1D2C2EAE7B91FF3BAF211FC69D8C72AEFF5BDC767E7D1DD3"  
00883 "1390110936BE0535EAA37B3B2E11E7FEF3FDBDA3D90A9F264508F7B2A172D95C3C43EFBAE0A"  
00884 "E706361BE044A157352A719B25A26676C9161AAE58E8A452BA0561AA27DD5486702FE2031"  
00885 "A55668CF2D92A920D08192EAE64691268C3D50B5248A3F1714BCB3825A016F6888C2998C2"  
00886 "9F8129CA6BC2D5D5B3A9CE8425FCD39C84E5F311ABF8DFC84EF58543B3FDE633124B329F04"  
00887 "2AA398D9386631A41464E6B32527435C342004A8FD78C175031E43736F060DCFA3DEC4E59C"  
00888 "F18599121BE6E817B518A93C803EAA1AA807530EB7E27492EEC49470FE4AC65833EB4EA3EE"  
00889 "8CB64A13D1AB688FB81609F068A876669D618CA663638715B2F0ADB5D4254D797F19D87DBA"  
00890 "270DEE72ABCD540C2174E8F88C8D99EE327952D0CEF2B69C2D18DAEE031A42E2C7790F59A"  
00891 "77C58879BF16A720FA334", // PRN 47  
00892 "E55B5080920F178FA3DDF8259E9DC69921D014969A7256D744B6B55D5D9C90027C2D748AE"  
00893 "72962A945338C153701F3803050BE7791704A0FE0C3A2021F2B3381694C6A9A344EFC9ECA9"  
00894 "8DE06233AED3C0E5C210866B0CF6804D356122AD5B7F3348FB764850488DFA8E37E70D86D"  
00895 "C87DE5150F408C6E8E6BE0DFC2DCB1E77F9B60548B4657A0A8B7FAB7736480FE88A72B89CF"  
00896 "6EFA7B18DBF43B611CA7B61145179FC6293DA3B5373C82DE82EB80ED227757642598B8FC"  
00897 "5928B92A038C28D73B7C146EB4B5C2F4DB9159ED1C5F3C99D73EC91EA22DD78844AA60E8F1"  
00898 "22E6EC73D241CEAF9EB48F5937351B6BB074A43F727BB05578F37C3E962769AFAD89E237BD"  
00899 "B06C74534AF5B3D5CCCE7B73040D589E9F9A0EC0D58A51706425A02B637620ED29C5FB1A7"  
00900 "A12B1B79A0C42C60BC9FB46200C19619F57A7AE12248FED59BD529CBBAE3DBDC8DF43D8F8"  
00901 "42CA54824B5712C0698523C6F2FDD38F93026DE1E474DF762A41EF58245A70A91B6B22A6BF"  
00902 "4EE6CF3DF6384FC2F6CC4A462D025864BEE4BA550DB1BD956E85B18954CC86075118A384BE"  
00903 "EFD66063CCF4BCACC57A890494A8251863FB1E464FD711DCB1C0A5CBE9456198AF4B0CD942"  
00904 "76EEB6C96948B939439D2580F5E1F1A58304B515DF47EF808BCEBDA532C67C18D5BBF7CA07"  
00905 "1C7614681285889DDEEC096AF7C133083CA787D7F3D9A470BC81B151F3FA195680F0679EC6"  
00906 "88ACED82D41E3BAE622B5F3E11A3B892687722CBEA99A0A28CFEC5E9FCD0723EB81A43F6F"  
00907 "66047FD136EB294E705F25A85C95AD8470273DC3B3CFD2294B69099154236EE9588EBA15AD"  
00908 "C0BA2059040C194821BC290FFDADF97F6F5DB9FAA9598A8FB378F3C442009D27A30FCBCE"  
00909 "A4B19D19BCC17000EDEE4", // PRN 48  
00910 "C62BFB7D44BD9DADBE977CFBD96E58F9545D823DB6888B0C6345FA6C21D5AC83FF9BEE1107"  
00911 "2679F00041517757C476A876AA62280F8976DEA8201A7372719C54511143DC6344EC05603D"  
00912 "3E70C0FD5252D53826D84444F7C89B12A5BC439248BF7ABA05CE5575881292428C83A59787"  
00913 "F81BCD2C95270A13AEAA518EDEBDFDE798B28FDA50AEEC08973FE3AC395074A8D33ED0B781"  
00914 "975BBE0DB7CB6D7EA6BECE34BAD17EB538A636C2D174E031DB6D26B4C0633010785953BD7"  
00915 "73266430B12B14D1FA3CA8D0FFE23CD135B9368A86C01ADB25D9A98317F1A1271B7EBACAB6"  
00916 "4097DD40678F157623BE40DF8CF1F888E117CE8476C6A878EC5083B9821EF0A65FA51DBADF"  
00917 "ACC3476CCB2FBF1F5B6012BC539CFAB06C5B90C7D5C3AE701A25CE859B0AEA2BD3150B530F"  
00918 "AD35C506C3249152EABAD7772715528640A6DFA24C19C945A51C4D76B7114116A1D82FC9E0"  
00919 "6E1C72722CD2C92F11D78645737CB634C383148C3B085B6075BCEAD9E68E4636E0383C0A2D"  
00920 "AEF63CBCE8E1E93A70EBB8471D95D2663DA8BA5B4D061FCE0F144FD1BD9D94549986819ABB2"  
00921 "FCA20AE20D6BA2D2F11F0C1CE8D2B35A9A5341E509B9276F59271907063D8056B1EF5793F2"  
00922 "136C32D113F6B1860C9FCB6C6AAEBB7F0FF95519A0D6977F8DAFE137700C19B8006A7A3F65"  
00923 "3BD6DA45380A8A604D317CEBA9C2C265A9DBD6D18E9F8441BC03E4F9BFE04A9DB0CD818ADE"  
00924 "8D831731A9CECF11F440886F09D3863062F4D92BE0142886765F2661E3051C5A67BD3502B2"  
00925 "8F362130E4FB6F0FBF76BADD0A4563187D926E36BC61D88AA5E68A1A127C6B21B6FDF436A"  
00926 "CC6AB5BE5EDBC217287280696169BAF2810EEAFE785AD047DE2A6BF2C060F6FC044F5C337"  
00927 "176953273BE8DE7FE3184", // PRN 49  
00928 "FE62E6E5534E3DB8EC780270A7C3FADC4A612A91B62293B2B89A78CE2196AF2D0C693EA70"  
00929 "A2D61A81D295E87FE8A930659AF517CE6F311EE423B2F1076560CA0B4602B060114FA00C6E"  
00930 "681DF6A9DF7DEEBE9314E1B022BD714D90BABF6A2E1262A6F01BA663CAA0CEFF977A15DD3E"  
00931 "564A8EAB39F72F60FEAD18DEAF2B1702441F262155B3E63A19A9715A76B276A24347C0BDE7"  
00932 "2D74D0EDC29B8C1D8EE902AD3B5295DB074AAEF0995F30BAC3FE4BA8C737E5BC2B9D1FABA5"  
00933 "E05435E47FAB7F960F3A939FAE76B4D525971D18A5DB7F50ECE91B9AB8364E5B6A3EF874B"  
00934 "9616237724B8A473C7D50339D0BBA17D241AE85DA9A1C52F79DCAF5E2DE5A618A3ACB2CFD"  
00935 "CE17C0E603F7DC10B59E05235C802D96B36C924FF54C7C0418A0E76EE77E70D75C1995C0DE"  
00936 "224CB5833067784E872AC3340A2950DAD26F1FC373F71DA4D1DF6069C1C7F5F772393BFF6"  
00937 "F3524530E69857E60AFFFD213DCA2614653C2064DF76234C8B5FA1303BC0CA9C35BD9EF5D8"  
00938 "8E3347720822B3C06DE44509871339133132602D4228AA960800A2F5883CD40327F9C1F823"

```
00939 "BA56ED8ECBDF7CFE0D26A21BFE9E1C6A86298840B6E233CDA4DF652B7F0485946AF349CA0"
00940 "C6738DAB1DF32559D7F623D7CF74F48809B9731A918C21B1F3242F8A4812F1E558F8DEA7E6"
00941 "8162ED00096163C8F6317D6A19326F9606374453F61B7E7816F7DAB45A04C8C687A218425"
00942 "CACA9D94A185BDA4B6C9B7E30EBEAE6E992A6F5590587B06C2B73024E09C46104750B4B446"
00943 "A971B9568C4C29C036401CCC36A6580DDE2E0ACC4509D6D0D4E1E7E6012B01411487673DDE"
00944 "5A134BB226BBE1E508F0A7CE3A60BD85CD528CDC8D2D86E27A27477C83F6E1A9CAC0AB73AB"
00945 "D25D2CA84D26BAFE257E4" // PRN 50
00946 };
00947
00948
00949 constexpr size_t GALILEO_E6_C_PRIMARY_CODE_STR_LENGTH = 1279;
00950 constexpr char GALILEO_E6_C_PRIMARY_CODE[GALILEO_E6_NUMBER_OF_CODES][1280] = {
00951 "F5A3D656F9DAC534C05240934AEDC5F175B4D4873FA63FDBB4DC03D20321BE8E78FD2B6525"
00952 "51FE532214B660B493575402DC6F93F68B3A22B5099899A21DFC979C3A955A68B6C9EAD355"
00953 "009D6C30036FC1C0DA770D6C598CBC95CC2710DD627B5E10CF9764825427FE0387DF7EC6F8"
00954 "2D04F3BA9FB420C3EE731BBB5E39D5B9A52D6E74E1B359D94F31A1A6583CDDE1D34CD5B18D"
00955 "2F66E21E6E969BA981631F0C9BE31C8BBB66BBBE411EE227BFE9142C851A4EFD9D8B4C42A2"
00956 "C2480B47B39FA395D29C329F0016CE57943AFC6E927B439441C4E2C0475AE9C09D566AED39"
00957 "4DB059623D8C1288BE6DF72C69C6AED2AA5F8BD4A978C4A3AB6D64FF8B0FC1AD4956E183C"
00958 "047041310390C0F39F6AC03E20DF72A88121182170310B7B72E742743FC49DFBC34DFBFC38"
00959 "0375457DC4147E78BD655EB3BF5D98A4BF788A8E9AA8DA792359F2E01F6E368886D294958C"
00960 "8A9EF481BB6C7AB4DA9D4534881560900066DE707B81F873D6607DAA5EAC4C4EA9D1B1B81E"
00961 "83A69E7ECE07A3B525EFC8FC1D16B42A3BA6B215E87C1698D15E14A066045C8CE1E49E39A6"
00962 "F54F473F6ED2D8A9E777FC39B2321E71BA999B11BB17D5C62894C5125A687EAA0A4A30817E"
00963 "C1BF7D7A5EC983029218255D4558A6784619444B916238CDF511179464272309E5F1BE2AE9"
00964 "4FE7FB2578415A1AAF7414EA7100F9D7E9C886D148D4DB3E40B8761857F30CBE06E0952EEB"
00965 "D9425671A3381EEF7CD9F43B35E3E8563885332CB1F0DD358EBD237D513E0EA6D878001AB"
00966 "CD3138270F9A84B6D223FAF18B959B1E725F85D4A26A1271217F1791E8CEA41D307F38ACC7"
00967 "16068A1D0CB76F8337D32B9602297C0E23BA115679B219A4D5936F5A1E7A03F2AAA77464AF"
00968 "A89B278C63865F672784C", // PRN 01
00969 "FAF0C60E496650198BFC9A6439703908202698930C129C4F44DFF72FB0BB4CD682B432B372"
00970 "59394EB345B0170CAA2B80CF697DD6A748F16A57443AD90414D29E2F08585261256D601F25"
00971 "C69A1DD6E1E2A84CFEC805E231ABB2636806CBE410E91F77D39BE7D2E2561C14AA6255F70"
00972 "893888BFA75E678B64CF24A6E58D6995BCE88D52EA67D235D142A7087624D70487B09D5DEC"
00973 "BE0394BFAB0AE4BA70C9FE89BFF26FCCD8D0967DB8F8DEE946E351D682EDE22858DC551A0D"
00974 "7DEA29488C68ABDC47681284A1EA38365D7DE5040453208BF0427722B8FBEF5C945DA0053F"
00975 "E68AAEF688E36E839480D88119B98D724791825087E05A460DEF21C8C5606A12CA60DF8FE"
00976 "6CCED0894CC82200350A4EFD1CE37D9E0F638546CA829E1BB73B59B324CDF74395FC2AD279"
00977 "8CB6F4A3E672AF65F107BAC8F03600A6705F13C9E61E249859D1A80AF1C1DC65586FD5DE0A"
00978 "F497DD14530071EDEE0EC495C29160B62B536613CDE573AAEF518086C1FF759F6D140633C7"
00979 "4431305DAB5C78ED02A62780BFF8D0DD8BDB3A8ABBA7476E0233EF37C483AF79BAA7467014"
00980 "E718B032E782F6A0A0A344CE0E9A74C6707218ABF4732912E1FAD286D8451DC6CB4BAB5DA"
00981 "F13F6BC88D784C2F768E77679C97EBAEDBB0FADC2FAFC767B6BF8EE2E75530DAB81752A318"
00982 "D1A20409A6701C7DECAC13E5403EF6B6BC7F08E0E71834AB6AF60F10756F75B12020681F3C"
00983 "5FED958DC68CB94DD76EE356A9755E68C98ABF565A043FB0D5D78DA229EBAB7B8A09334B"
00984 "C7EAA1CCE4B5B3795A0CA50DFDD8FC97E02514EF8B1116F96A0BBC99C6C746E63B229E9DD8"
00985 "F12B8F523E2334E63B497FD9E4559071F9FC5DAA1EDD430184902FDCCE250DC635B27CA59D"
00986 "6C25E97998670D26B8B5C", // PRN 02
00987 "D201BEF00E5E7722A2C298AC291E48BAF9A23010418483F6E934B44AA77B2AC6A63E5431C4"
00988 "C511E6BC05CA757EB3B55EB419BC1CF0B9F36748B966BF628F6A93A25B5B82585919F076CB"
00989 "6A8CE44BDD5F45BFD61887A1D0C166E2E7E7072FCCF9153685F284864B172CD4412256CF3B"
00990 "3235935485D2D7AEAA104F325EEEC7690E9128FA66FAE53F6F909391A1A214F59E8622D110A"
00991 "1B3B8F0F9353360F0D0089F004D3138EBE2242A2F1C3BDE35E11E0927835F64654C9C33EEE"
00992 "D8EEB95BB41E2984506F055240EF1E4BC37AA09957258269229438F1DC2AE9B8D69EDF81D"
00993 "363476077884F8459E0530B154E180D5649F5F0976BB57024630B3E79F75F8D771C6A228A1"
00994 "F54243ECAD36DEA7FF0E8B2C00C51A7C2C57C4E126D1DE562F78729BD48BCA7BEDB42D926"
00995 "655FEB9C9ED27494F116A401EEF84AC1CD5FE0A6307E31CE8A5645BE4976CD69E892E7AF47"
00996 "F879FB74A1C9059EC7DF6583147ECE387C98F1D113056DA88FFA69823F1EC0BA85C15D1926"
00997 "5AE3B4E5FCC492D46FE95A41363EDF1FCE1B5587DCD85A5C38534FC7F6E8FC3792B07F18F5"
00998 "492A1EFE800E75ABDB9FB2D95B7A0725A5DDA376D87195148958B0A8D68CD70C6A3DB61600"
00999 "01E9B905D9AE3F36702C31C599141271BCDCADB774161D89CDF69D357BFA138C0EF8A1DED3"
01000 "47E94964E515AF4110FEE5FB856348981DECA930BDFE097048C1675EF70CD2AE95FFF80869"
01001 "465EE4368F659E7D866CFD48939E10AD23BA969BDCB06B1BF01089C961D69656348ED561"
01002 "1001726AA8F60222A676C45E64985C6D7550AF9A42242E84223EA30E40833C750D6615D200"
01003 "BE90DB979AB964E87C28D4E20B1BD77B483CFFA8667EB25B7689F7F0D6FD360612BD9883C4"
01004 "B124F1614B0524EE8F4AC", // PRN 03
01005 "D4FD152565117A08F0E6EFB5F7F7887F38907B4D5AE8B678EFB899CEB429AD7DF00FB70384"
01006 "0BBEE662284FA7C3F422D20BCF86699386DE3D1FCE841FF98E6D96729C5CAD69AF22A3D053"
01007 "64603D1599FF1FDC5541DB486E9B8616E4F0A6151F9DEB45CE1385822DE89422FE2E0E6045"
01008 "BD974469502FB22895A03162ACA82A59E8193D21CD4A56EF18593B61E26B4FA19DF9659237"
01009 "84C11B28D58FA424A286E74EOCE74BC24B1DBB1494C5C0D0689EF695272FBCF1357E01A147"
01010 "51804238425390E2313CF280D93CBFEED9A9C9E4430F5201FDD172C46AA4F92F4E7CCFFD884"
01011 "B662D579130B5D292D4289CFA6647BFEF52850E3FBB2DCA3EC501B57012A1B1099C25BD1D7"
01012 "51428D52EABF52E41E4BB2788AF7F588EBA91093C89B72F766709351F9BCED9A0A5C00A82C"
01013 "D5B8395A847828682D2AE5A99F5D4199C0DED495452BBBC53D50CEFCBB92F294A1848C1625"
01014 "7A75D4AC456F238E364B28A4C9C175FE091E9FA4A0A48E7B6D82A4D1337A659F9AC752EA7"
01015 "73D09D4532459B3FC322BF89E29E6D8E8ECF2F7A02671FBB69A747163799E475A048EA943"
01016 "59C2646B7A7FF8462BD296A4A534749977A73EABD2D9B92C19D085554E4E91875989943E7B"
01017 "650BD52F279DA6F16B7197FF9F9E973075E456D7BCF81C847C89A60C9F772236E0411FFC62"
01018 "160D283645CE34DD96D143B99F20F6F7BF5331D7167E5429E3B4DB62ECC7C301E6DA239DD40"
01019 "0899B838A1BA42641FE3B5EAEBC09C1D9190241C072FB1F7C6594CD8DA18053C8BF8179A62"
01020 "1CF335477CE97C05A6A9F7D63D9D4651B23318285B3B7341207686AB801BD89DAED3BC07"
01021 "B0F942C2AE2970136F0A73BC69455B4D312015141D20ADA39FA0382C6783181DB4380584FB"
01022 "AB7C4F1ADC15E261A07C4", // PRN 04
01023 "97010A672C963BBD3D3ACAC38BA53CB739375BF1E093B2E64296D0D07476A0105C9DF0204E"
01024 "A54D9143B1775615OA377A4E1F4F2A741B4FAFDB369B44C37F917F94D19EDF8EC3E00B7EE"
01025 "3251756875111AE7F435057D4B7EE6F3B4512F99938F6F13984BC23A086DD96BB4076C815"
```

01026 "AC407D693E6FB1685C6D4B7611F3010CDEF1B2CFEF6FA7744437B8D199545720B19CF7BAD6"  
01027 "A7B12CDDFF2CC45811CE4C7A35CB5E9FCD95C2CD0BF16CB69C95A46B4541E0F7FB4645075FE"  
01028 "4BC1E64865954BFB96187C265196CAD43EC41035774460FAA570186683A33E019719B14329"  
01029 "128C4463FD32E5FAD23CB21D6224ED9634CEB7B0A36F359D3023D1142C10BBD85885329618"  
01030 "E92BDCB786799423F38F7C871C1FB356C5E4008F9535D9C7334FD2D3CC4BCA196451BC47EC"  
01031 "7B6C11283156BE8E53CBC4DA262AC1D19EB20350C5541F9F76AFF6A8F8055FACB6E1DD43F"  
01032 "53AD0895689AD69D2BF12D9A1D1FFB14C7AFFCB0B9F0529122F1D1EC0BBFC582773EB58854"  
01033 "7D68D4E494CC330065CEC477869F15985F8D60D4A5E8BE45F2D156C6353347270EF50EBD24"  
01034 "48D2156250B6A489A8BED543561F099ADD1A205C1F32725DD2A6759C9CBA0215B06F93FF42"  
01035 "3F7C10FD684E7310BAA9BE72BFDE2C2AD9210257E1C16DF9B38D94084F31C8E3D2627C1506"  
01036 "AC041DEA65ED6C4E04CD4533DE37E65577F54E517781669A637E6C4DC98E73EA90D70A6D78"  
01037 "EE8807E629AB100B68F2EC8282F232E8C8905D82FCC6E592C91A3E105C3EAF38EB73E02B3"  
01038 "7ACD8F6447CB22AC99987A8EA7CDFFB0EDA3B96C68DF4A4D3E18391C84C850161F5058802F"  
01039 "2D32821AB7FC6B31C1013E4D1EFEB41234BDA4C84107B3FB5E7CA8D01816B8DE61C43CC1B0"  
01040 "A9EA415DB41D4820D6890", // PRN 05  
01041 "9CDFD69656832C506F718657BA4A8EF7F0AEAE78FBB829A1EF3295DC643FABCAB21344DB1"  
01042 "DD8E1DF73CCAAF198619192AE27B42E470F1C7CF2907D1C8F6F0650F52BC4C0382530D1547"  
01043 "52BEE42D38A03D1CFAB68AE474C22C7A4F4E06F2A3FDAAE81758D66741FFF25B6AFD529CF3"  
01044 "C3F7EF339EFFCD3A3499DF2C85BA742FA348D449C167F7BF00688B11C453EE783623CC8476"  
01045 "2AC049D9929D470037709C71BD6FE5019C9432C8A7CCB29904C1846586D7639B09DEC1F194"  
01046 "9AE5C6D14CE92118750B0FDDCCE61DBA6B1EBD92CA5B8DD456EB39936FFE010B741A22DF"  
01047 "11DC489FB7EA415FB265BD2A7361E7A428E47DE546103CC8E4E7F83A6936BCB1F9074E692D"  
01048 "4C2E1BCE4C6065D407764469A5B8BFD4C43EF9CB843952B5141208147C47ECFD4B5CC6953"  
01049 "E77E0B3498A9EA930B9679AFA3227529BE29A8794AC7A23F6E8D5391B81FCB07F3E3082DAF"  
01050 "B331B5A9A6C5C203BF1402458834D4B12F69F0A3EA8EEA9357A4D2A7FF90BD7ECEB441BAB3"  
01051 "F5F6F150B542EC9DC17028E5C13BCEBAFCB5D2089D8F207CE493F65917022BF08E1D55A299"  
01052 "F420D651610A07690A2511BC48ADA6F68077AB928A3017A01FDE99DE9ED02BDD06EB9C060"  
01053 "02580554C378D701C68FC56F1C47362CF5113D4D728CE9B9E4C77B086258862CE67992B34B"  
01054 "8810A4766B4F26BE34CE35A356E81A828735022E70F3ACE90CD175AEAE9479BCA317B8B629"  
01055 "DDBC67D21CA13A0155E57332460466CCEA034F79EA1A8345850CDEB29AA02701A5FA6ED331"  
01056 "EB96A9DCFE3DABD9009A5DF5D4ADC8535675B073060FA15A424EE7375D039C923C9448DAEB"  
01057 "AB6A881DEA6567FEDB235ECFA494068536F1F41D1906CFF5C343952F8E9141245862609079"  
01058 "14E4C5E4225026034E030", // PRN 06  
01059 "EE21FF4082E6C3345ACE456EC43F40B2A12AE15C5BE1BE2807FB93BF95FC0D5462806256A0"  
01060 "3C1ECCDEFB0686372D1E8CF83FD93BC5A4ABD65171A643C1E2EB2341BDBF5F1920659C23AE"  
01061 "E9641712E4BD7AFDAD4E3E95CF019CE0F43E87B4838DE690D51B9A1CBBE45C8B8BCD5757EB"  
01062 "B51A11A13652A4164269150A901F48232753F9C633D459ABA310880BC30008F5B8FECC0924"  
01063 "BD2677EA5C3073959497B2F9FA34F9A93D2EF528DA3B43E96DF37CE5C0D11C207699246CDC"  
01064 "6D0F18DE1C6437DF6CD080C6E7415920A75E5358B64B7D61327C6BB90918788876CED884C"  
01065 "8706EE4D7DB11E582DAA279C29DC0E828A2ECE0FBD33B63A4DBC07F730500F6D061D76DE3"  
01066 "4796347839A13A286635F44A05C340B3B20D3FBBC48EE3B2E156BED9F93E0AD92C24205A72"  
01067 "E16FB425AE3D8508D4D8EAD20080CB028B4000FCA39FCC13B0630BE3A4E2B2AFD3622CA1D9"  
01068 "8D85BC88EF298B4A5DEAE33992D0BB1630A10D77165E0FDD2E757E7573DE3A17F3AF85C7B45"  
01069 "CADCF9A6D1909737934294485D54A62E8C681E64093562EF0CECA523784E1A46345EDF26C"  
01070 "F69C895F5ED3A1195F00D1DEDE55105666BA9819B7A20559CF624D5A0B167A8D63AB21A63"  
01071 "30F89266A43DC90F712719302BD5E32AEDC6F0A206948B9E37E4B2ACA3542EE36ABD483B01"  
01072 "00921531F2F9A6CD3135B250187C49628D52FE26F13F2A1C4E5ACC734C097D8692DF4F8914"  
01073 "4C32E767AA5BBECCFA3CD0E49A24F0470E482AEBAAAF5FB33B77B52C1579B07A9D22952989A"  
01074 "B1CF9167CE3552F6506C657E93FBA9A9E9C30F9AB133A01C8CB57702B8DA67D64D6BE89ECB"  
01075 "7D75C431AFE28C02FF57FAC04962C412718CF9B8740F42BBF580A7DCF37E72CF43C3958D5F"  
01076 "F7756EE1930A9EFC7A91C", // PRN 07  
01077 "D57C59619A6F398CC2A89D6D54F9E1F846BA7A7DECEA83A16101DB4BB31DBAB22CA53DFBBD"  
01078 "32AAD3FCCC964A4C441B6DCCE145BCB7A59DE9AA123E8D590AF9999C7414333ADC7BF0C2CC"  
01079 "3E3461232AA7C4633B626BDB6EE99EADCCBAF3D2F7D51963A604C0E234B5B1F19F56301C24"  
01080 "65974324FE52F22355B7E8229DEC199012AAEBDCDE7DA613C04C3D978D615C07214E2051A3"  
01081 "D8BE099AEA258AA7765FBD3E9BAB6F572B81F0A6FAE0C2BAF0672052ED8B23AE3EC6946899"  
01082 "683E6F1055BD70BB78E82503A3CE3DB48EA5E47D229E97B0C48CA00C771E377F02CC0834"  
01083 "2EEFAB06F7F723B910B14C6793925C11938756A787AB0A05E123CEDDDA545D70FAB1E78CB4"  
01084 "053AE3ECD7682021F5F9105DB0B3CCPDF0A9C2F4E5486749126BD7C4DE65015F84702AF1D8"  
01085 "A4213E5C7D487379C54FE1E1224C2D576217ACD1490C3E0FC84B24A2008C55F335137195A9"  
01086 "AF9CBA46F695E5F97C7C3B2165B561920C1C488D9B165D1FB726B63758B61B8E93F3E5BE0"  
01087 "97110612DC469FEAFE150E67F594F2B8D5BC1EB4AA67F733FE309D96A14C0716454DE41CD2B"  
01088 "D8668EA6EC1084A3C2A0C761D78DFFBE70884D7D24FE54AE8DAA71C857350395CF11AC3FA4"  
01089 "BB6AB84F58F9856D802633DFE2C67840AB432E00F42007EA46AFC0CB22AF6C97241F8B7D0"  
01090 "5081AA67CA16B1217D467CC0076C8EFE15B9EAC83E08D65E13A6686E8F72988343411432DE"  
01091 "B668A82FDC4B73E2C1CC9086CE058EBEE5726C956B927D45FB646CE54ACAF821E993BD3313"  
01092 "2FC549C620346B23AD0D86422BC0D833F8C866279F12EE7DFD6C6F7E2B088EC6711C2CEF6"  
01093 "EA64FD758345040D55ECB443BACBFD8AF4723A1703A8B1B633DC5176602D1A842574AD30A6"  
01094 "4403C37C03BDD026C6354", // PRN 08  
01095 "829CC3C571A2BB19164855795A60833734868AFF6C47F933119B4192048D740AF94292F56B"  
01096 "22B263CAC2047455A46790417D2F8170D05C57C62CB0B06209B6A12405F7FE9146EC142A4A"  
01097 "B8F7C73768A4E98469B897FA1BADFE07763EB95675EA202BE998BF04B68BC6E3ABC749F60A"  
01098 "A44DB45C609CEFAA011E3748FA1FEBD9A5894C2CD399C644D09ED9E25E8E574056EADC9670"  
01099 "8887685ABAA4CE56B27D7C7CCEA8B28C4D8A9EF758A502747717C7E14B8E4DEE68E2B545F9"  
01100 "BB5BB94E53AE779638EC429BE9A9396666BA6961711DF385FB4E1E9106BEE3F80AEF8F5392"  
01101 "FB31E74E16168FCE9E1969724576B522A4C91620A44F0EFFBE0567E87C7B681791D4E51006"  
01102 "8AB39D7F16D5E78E17EE00C14F722E4CD48AD67EE110401C70BC29F0F61DD1BC91EF6741B"  
01103 "3A77745EDBD43BEC8036B6E4A2CFB84A5C191EC35B561044FDE81A7ECEFC4E64AEDEB9F10"  
01104 "E5303E2A25AEDD8F10B938E8F580F766F73E2A53C04D7C82DAD64BEAF4D8EE3A47943A6CE5"  
01105 "A2A3185D5AFB2C4B829D02F86FD976895056869B28B36851337F36412950D8CD6F957EF06"  
01106 "7BCA0633EAA13ECC7E600E2A1B3C42067F241D79B07D045A3E38990F6B6E0A44F2C10834ED"  
01107 "3A2DE1D172534C31E8ACC6F66D9545906AACD083D8EA0CC30F535883D6B110B1CFEDBED33F"  
01108 "77D84F69F8C92124ADF0B633B62E4BEA52085AF0AC40CF80803A0459E345E6BE44D925F7"  
01109 "CFBF10D7EEC73816C8D87380D93FBE77C300B2368C2AC45BFE5004AC6ABB4E68377661112E"  
01110 "F717EF4F34B6BCC382D0F08F75799B2451038491A339A0C5675DCDFA5044B89EB4A423F1DA"  
01111 "084AFCFC32D9DFA6B7808C6A1D108A33B8FBAB59A758E4E6CAF334971413066908B0B1D56A"  
01112 "373515E8F1EC066E8AB30", // PRN 09

01113 "B2413CBEE7A6CFB5400DF607C16E694DBFF1375B9AB755724EEB0759221F2CD261F15C2BD"  
01114 "D03C8802B58035B4E9F7976E3A31286A9789549297DA272693746E2C3841413F328103F0A0"  
01115 "ACF1AB94F0E35626F11B5D8934BE54DB9524F9B5F9BD4EBB46BC8350E78E4C05885F7DF3CE"  
01116 "D09369D44D4025DA9A861274A10AA161D4A69701F03F1684716C6EA79FB19CAB0DA4368FCE"  
01117 "3BD177D2B9EB4C9729F615AEFE232B7875F2D1125DB8838292CAD93F45303AE0C8BFBEB0050"  
01118 "F32A9F2BD11DA869D84976441ACCB6311DF1AB09143926D138B336B839DDDAFA5FFA32F1EE"  
01119 "65F6F7D2FFA363815C288FBF708F9D81BF4AEE2ADE4FFA5F2FA647EC8DF62D564C41A5EB04"  
01120 "BFAB5803EFE44B041B7A2798AD9EF34204F573D855136442B5AB6A85A402399C1E4EBC0D11"  
01121 "1C9C3A374FEAE92B6ECD22BF21A47F7096C48846FE685236171B4B19C80778F4236B68078C"  
01122 "7CC3CDFEFF7CBA3A7681593F0FADA37D709598AB0A6DEE2DFB0ED8894D2457CC268D1C4D78E"  
01123 "7FA50B99B19AD4FF7B97916A87B39C2694C24C360EEA1BB0920C2DE28AF4D76816E63690E3"  
01124 "8BE3E85BC6047F8AC762D85C0D641F15497E4615B295E314971EC646136D0418B42D0F326F"  
01125 "B798407077627FAD436271917398FBE18FA095BFC7521E361683E464D2EABB64D5C3BA927B"  
01126 "9A10143F16C528C91E50C625807717648B7396EC6BBD820FAC6C3C30A517CA3A730A4C7C38"  
01127 "E9280A27C6496723DAC59D8F02D0519C306069FF2F1F6871892613C11A2C7001EB913341B6"  
01128 "F0A9B2049CF8F156B695C10ABBCBC49527775FC3E824AD7B947F00BECC85AE53C5C51CA50"  
01129 "7046B78E674D7CBB206C09183EFO3DA63D911484B790A5128724557EF769B76167C8126D43"  
01130 "32F1C42EC16457EDD1108", // PRN 10  
01131 "F4C4B94772CE84695325CB3AEFF60BD564147ADEE80541C6FE5F5653DFF51AA05764D6FAC7"  
01132 "B59B0F365EFAFA4CFA15EAE9AC57D932F9B73EB90328F617C9588A8CFBC92805A025EF87E9"  
01133 "2CB06A1DDADF24097395C4D45189D229B793A241C80E4124F9890E018647BEEFA9534B5118"  
01134 "DBEAF65432D66D2A0A9755F5EED57D79930B5112F4823632F375A9B085AFE938D862139EE"  
01135 "3034DF716AAB84D9A233F82561426D33BF33D70C3181477B24BA1C3E70622817414C58BD09"  
01136 "C0CFEE8244677F2B3F5F12BDD5CA3300CF23F854FE08BA140C2E217200D17C4C368CCDD"  
01137 "A00B14B289F91F75F7D44D3F556C26E8C5476340DF71B01A993038DB68AD589ABA593CEF78"  
01138 "852B567AE047B1776386200978E45FB6E7380ACCE6575A12EDC02A6F4070B570B40CD05E3A"  
01139 "16949A681771A349E66B6C8BAE36C24C4203662E1C1C0C4744800D4E0A4356CA79DD087839"  
01140 "D35B3B1684B1FC6748526854CAD7EE517D54B4D682C6951C74ACB98E8E63ACB1567585EDE"  
01141 "5D056A1C375BC82BA4D9E3D1773326A54617B0790E1C687C15BFCCE14C8E7A84670C01FE0D"  
01142 "7CCTDFA1604A464EC552D72AF3BB3FD3F4CFD4A94235E132165D14B1F412331FF47D6F7AE5"  
01143 "B2A429237BD09C2085EFC7C0EBE3B799D78621D8DA4CC7A67966DCEDD37AD84F9E44A775A1"  
01144 "F1CF610077CA993E13117A2B0868A677E4E677F548658D9AA5CFE91221CCC623B08F65F0C7"  
01145 "301A0877E97DC3A6928D8B8E69E702703ED7A760DD68AE184BAF4E9F50AFC38138E52FBE5B"  
01146 "57BFD56230C4F4D14FDF6A95353E26A1366FFA0B41AA3703770D46D73B3EB27603C35E36C"  
01147 "008497C3CB01D8398E5823370D18F40B3BAC32ADED06819135A8BD06755963BC7793E6E0AE"  
01148 "0AF9B60FB948E5103302C", // PRN 11  
01149 "D174C617556835B19D5960E23554CC72DB74D0EA1FAFDF8C0246CFAE6A8FFEE862CB9A5281"  
01150 "EB8F140068D986D133AB150D40326E2FA87BB3450DAE0804A4C62C9492F5F96553D9D43A1F"  
01151 "AE170114E9C445029ED21CFB0C19A9EA0853A79C49379B40265A44E0817212CF5249016A2"  
01152 "7C4C2CCE7EBA1CA9FD688F46C7529ED9706420A3ECCD1B5057606BEB0470A3C763786BD777"  
01153 "1CE7B9FDB5C87C4BA27A3A7E165036EFB7F0695CE4365591206E27EE874282043D513A920B"  
01154 "9438A1F05DC14DF93C5D49D07CD44873807CC57D75382A2C5E6BE193743B17D5B491B9984"  
01155 "EC215EAE7B9426FBAF64D1C6B880635EB5067479C9AA351E8D6CBC98A65A12238EE347173B"  
01156 "FCC47E6D8ED753F4E6942DCF60A0A4C15C21DA17A13FF00A50E175F7935EB113A088E4EF86"  
01157 "E87FA710DA93B99E99ACFD01FEFB68F4154AF85CDF7C66D20355F9ECD51969CFD33ABCE58C"  
01158 "1B1807E04212C4D0209ED76AEF40116F861BB558585E144C9351451CAE605D0B765C87B5AD"  
01159 "51A7A430A0B8D5D05E58976137D5102CF771544AF66201B253D4665F1FF69E775A18D4AB7F"  
01160 "24A000A3E2376594B03CF933FECADF701949BAB66B51606569ECC2EC4F24D2720E5E1DEB68C"  
01161 "37594A1256EE376EACE92A5B728B9A0D1EF50530FDE76D94DCF0F83D393E3318FEFE11E035"  
01162 "23BBBDFCF4560E43B583139DD17269E87EAF57ACA8E202C51FD39DA50E6095DC0F73332CE"  
01163 "D27DDCC09F1A6D1613FCEBA8E94ADEA15C36C14EE6473C6688FFCDAB46F1E2657F3108452"  
01164 "88213398EEF25C72E6FB17E9A375471B96D5137E1246D035B6681371BE08FAACED63444E9E"  
01165 "36BCFF48A69DA97FCAD19F4A575C88FC7FC4ACCA11DF09479BB8488CC91B9E1B0C1D458A6"  
01166 "B30C8369823703C20E29C", // PRN 12  
01167 "C5DF1B2B634C4D2A78DF4B5B3E241E680D6C23AC76D6628F4F54640BF4A02635EA662C21CD"  
01168 "A94970DBBCEC6FECB55991D712F0CE85F1A51211302CC584E53528B385CF87244ABCDF469B"  
01169 "5904163D13053885C3FA5362CEC766EF045E712039DCEA042C0D4E5F40D96F3A5692EA71"  
01170 "0E4BBF55E343F0F91DAA8B57089637F5660E0A7CF3A972FA5EC1F3CB709DC466C9C6947F62"  
01171 "5B176F7001C1F8023813593C71E8525538BE92985638E24F65A76CBA45EA7A4A91A649D012"  
01172 "5BA259FEC3D98FD529720206788AFAE0069BA99F2C99E5936870528E6050B3AA4EC63D152"  
01173 "406FE5FA6B003887FB513FBE49046ACFBEE51B976D19E4155979FC8444A215799774A2EA25"  
01174 "4931F21B2BA11226C589029898B057EE6AB6FE46D835A70382581AD5CBF10BE6BFFFD3E742"  
01175 "2D92688D6B21A1A4F0E6EA33B75DC807ED6A81A1C680A0D404B336CDA9C25842B529D52CD7"  
01176 "C47199CCE9025A3376E7A862920435CA63A9A15F2205BA89F1058E11967546585FFF6D447"  
01177 "E608DDA0714AB130066C6802AA85749F8F0D456266DFB1CA1C14861E789F1CD517C849F6FB"  
01178 "ABCC77775501C4E0E90EE3BF6DA51F6B601040CE72D9F5C5777B2A98F361E587373E3E6512"  
01179 "D754084CFE93AC63350FBE72C7C0BDE2C6F172F9D37035F7547830362461FAA94EE864FF63"  
01180 "AF608670A75D835F34713D23724C538514B5CBEE32F5FA38B5ECDBDC38AA3D76F98E8B198B"  
01181 "FE357FBA9AA44EE2B2A6A6359B433C561FA6085F31C68A597FF3F0264ECFC0F60DA72761"  
01182 "3EC063B7C3CA6E6B44DE29BF78101BCAB6678E1E3EC1F1644768EE11BAA498055AFE6506F3"  
01183 "71328375D7BC6499E68040E254FBB35F13DA867316A310A7A99CE1555BE880EDFB23F611"  
01184 "697268C2A7FE22AA8C484", // PRN 13  
01185 "99553082DACAED3A9D8C2BC62B100A611FB9290BC70B80CCBC3848DB085EB673D54A3DA91D"  
01186 "7AFF3F32104D82C048265976A4D434643895328DEB977B0B443C5B707874FECC0F4F2FCBC9"  
01187 "D11B1533D69F76BB0E0385B2D2C63B6353A7A0C3AFDCF53FE9E114376740BBADC000EE5D91"  
01188 "AA0374537D89C52EC1509AE47872F21288FA9730A287C6DBECEDC910C04488789093830392"  
01189 "E19BD364B128FFBD247246E12325C1205429E6742D0BBD7215CE420145F72CB29BD25A906F"  
01190 "5C798FD4A575E775CB7D3404B7C210B0D52E035841031AA8DC341D8F51CF76B5A0BAB7A2AD"  
01191 "485B2A35E024E257E438E7D9C778D3D6C4776ADE71500581D748F8678682A06A92AF0BF58E"  
01192 "DB721827B4E53CDEFFCA2CC17D891C5CF70423A2351D30D68249275A12704BE1198569C06A9"  
01193 "A80942525F559AA1DA963F488EF43BC365F9851E4BBCAB743BDAD827B7D0651C832B1E46CB"  
01194 "6A3F86E2776C2B891384519BDE2B8718BD4C573214D1A5C612AF7766CF76C62B6614F31F86"  
01195 "843019A87EBA5F1DFB1E9F75EBD93C70211C627030BE342C81D872AA48B819F722D401413F"  
01196 "AD3249D21409E6C75CE3675AF1A58AFD6B6AE89E187E7F996AB04C8806FBD228A1BFAF3C96"  
01197 "8F25DC38F5D5A6BE70E964903CA9E26490493B2CDBCFCA21EBC06EB4D091F68882D5921CF46"  
01198 "2B633DEB2B22F6C0B5BA32BFDE473AAED4438EAB2702A96D9521F2E0F77E6A0572245FA7C1"  
01199 "CE8D208C80631065B742CB5A4960CF26FBC39BE71A3DD8F6917B5E42FE004B240DFF1FBD3B"



01200 "EFDE62D365CD90DE8C8E6EA10C0BB41B2F0E5E727298A51E6E17D7C1DB21463D7A7EE7D31F"  
01201 "6D6FACF6EF6BE5DE2923196102A1E333E66650AC7560BA7BF414FE8DBB6BCBB7D9984FF2BB"  
01202 "C67CFE3E429FBD089B118", // PRN 14  
01203 "CB1C401CAF1A29C20B51EABD89D7D33CC1A87FBDC6585DBE08DB43E71263AB39FABDD384A4"  
01204 "A3FC2123F946DD41DC71A001DA565B7461298E3F7563D8ECE03A88E32E0B160F7FEB1BE347"  
01205 "8EC8A5678EDD796C4189653F60B54E1C6883E8C8AA39E25339C7ABBEF64F44F0A101F1C0B2"  
01206 "B2C9C9D8AAA52D6138A97474C6846B748A00C79F0E6B3199606C96F27EF514D0DF0A7E19FA"  
01207 "B0FB6F2BC285E29BACD339DBC0DB249D7793583CCD255438D6D6AB60F4391E5B00432DA093"  
01208 "7DA4E0C5CB78F2B3260C9E1FD4EDFBCE8B58AD830317492CD55482600E960AB8A817E273D1"  
01209 "40B98AD2A65D2FA49CEA1999ECCED6C959D9F1E2AF88AA2DDD5CBF23F24E97EDEF4848AE69"  
01210 "C8776EAD126005D19A8D5775DEFAD8D05D7264290EE8801DA11CD339BDD19226920DFFFFFDB"  
01211 "CC9AEA51DD307C46727239EA1A75308DB6BD265C3385F86F8A508BFFF721EC83EE506410DF"  
01212 "67A455C394F53CD878E2910DE68FDCCFDDFDD1BACA4C9B0F88F0B48069053449DBA65C88D1"  
01213 "9F8933071BB92BE7A8C90B5C3A273543B037D7376B2E62DFA057255696694E3BC4841D7F60"  
01214 "4D1AD887977EFD806C352A8521C87A170E8B6A34B51B1BD7A8D100AF8B28B0688F236F3E46"  
01215 "CC938DF606F971B0220D1B2DC5732FD84B4653917DFBF880EB6FC4CB5DD96783513EBF955"  
01216 "D6532097DC92A6B1EF5FFC14122C3487B250D4760160DDA65BA139AA6C6B4F153A563215C"  
01217 "075A6FFF369F91D20B19859810D090880485F8A6AA7AEAA92A821D28D30642E9AA09C7797"  
01218 "4EDED3C5ADE25381C17ECF1B083E282FEF30C861506B7570A4898A1BEE9FADB6BD8E5A1D76"  
01219 "4C240065B1E20E12C880A28F40C70360228F609F23752BEE6FA3131568A48EBE3D3A233D49"  
01220 "E038FF9AB4D38F5F1C844", // PRN 15  
01221 "B84F109CD8389BCECD75039A199EC00D2EEE6B25AC377D745F53A5D1ED39FB1FCB4F6F4AFD"  
01222 "777C4A25796488CF472B401D92CC72EA350ABD40F97299A4FF136DE0279E565B4A12EEDA83"  
01223 "7FFF7B19CA1E93F5A9916588244D13C93CA89E6755379E20F625C6100E4F49EE42B3C0CB4"  
01224 "DFFODDD0D48B1A493BBB70E63212CCFAA9AFE196E2439B464D8CBA5C2369DA2EF5CC516828A"  
01225 "BA5D913D8073C62A430766A58EA9D2AAE0FDBDB7F53EB787BCDD5A2D39A0CEBE2CE4821378"  
01226 "4BB8AFA96649F0464449FEF241137454AF150891B2FA6A91F46B6AA89CCA338BF0A24C7D32"  
01227 "C9E65249501FF157C3882B6901A92BF0F8404083D4E303BA691B8C459A20C28C20E3EF52E9"  
01228 "B723E6C66C83C517B3546568C18348B0F6A6280FF96BBE767D26E885122B1969017FFDCDCD"  
01229 "CBF9FEAB2704356A90D45E51F4B831BDBF9876D3DCD4E9ED85518DFE2770ABEE905255839"  
01230 "9FC54CFA8F681468D7E6BC6C86B041A0A076D99C8EE9419E2FE6441B260D4D1ED17283B12B"  
01231 "62A383876A4661713552D56FD894FC6E7DC47BF5186784DF1F94C777B5172767EE0BE98178"  
01232 "334F3CBF35BBF356095BB1E0362D5CB4E98A90DBB9388C5799923F13BAB7463F8056E2721B"  
01233 "D1FFF68C157383D2BFC00DA77BCDB0B551BAD8A7F775DA4318A27D64BD8B83F425D24187DD"  
01234 "0175106997F55DBC63829CD1E735CEE21FB56F5017C305C359339C487B0FB7E58EF0824BF68"  
01235 "5AB6690089EBF304927A4A18601DD705C44FA55A42BC89E12DED3EA505067AC0F8ACB14BA9"  
01236 "60789091564837B1F9FB41C14F2016E6049DE6F6245CED1A3578689CF8EF22096042B489D"  
01237 "7C5893C65C0D2AC101FA9057519B72562E724C2A3A30389E743D3F45272064B06C13D8B131"  
01238 "6A5A660326F34156B8640", // PRN 16  
01239 "B42ADA523F329F9CCA7199214DF87AC1C50118F7E238FBE3CE9CE8095F79E78481602141FB"  
01240 "C3B87CB1F7C68DE2D74F44F4B30B2EE48F218117512185B3F1F4AD2EAFAC164E64B5DB1D"  
01241 "D1D0425A6331638413ABAA7940CCA0E6C0C509B0379CC8A36DDE4AE0D9A66A2475C4E9013E"  
01242 "62DDA0A923FD6427F119E9F82E3E75C580865D23F04D99D2BBBA095898BE9D923E68E80C44"  
01243 "AA0BDF2539B2F55ADD4B7C9279501AB2AF650DB22494441C5C5CFF830E0AEC172971C5DF39"  
01244 "20F49B20A7454CB5427217DBF6165A09A038F22D958015586E4AF9A9ED3BFF2F9C72064BA"  
01245 "8F9AFC92F4DB4480348C578D6D66127C3FA9C1BEBEC210129F18997467E69F3FF46757B0820"  
01246 "D29F714D5E58E67D0349F1E8936EBA52A9B1617AEA064B1CDACA91276C51858FB8789E9E33"  
01247 "F5857A732B20A5FDA984A6BDB86EF7A8E26D56380555E8CDFEFC5264962B4E31DA645EDB62"  
01248 "DED96EED3E495A723DE2F80DDEE78A8A751DC24CF05BDAB15A0E4FD50B311E45EACDA6D1DD"  
01249 "4A2FE6367AFC07BB54D5C040A16EE0B892CAC0CB791EB9CF7CEAD2483B9EDC0E91E2F99573"  
01250 "429855A0AA4435D63F5BB1901924F887C563951D144B75B776E623EC065B77781AB0225875"  
01251 "1BA883BBAFF11974DA1ACAACEA54549800B6CD1784BFD9445439D7F1A832D4278688E77DD7"  
01252 "E3E2A51DA7BEE25BCC3DCDABE90A3E06D6EC0D57F0EBE3C98E751567FF5E1E0E9C3D74DA40"  
01253 "4D9855A0AA4435D63F5BB1901924F887C563951D144B75B776E623EC065B77781AB0225875"  
01254 "1A06624C5C803C6BE60EF6F60D24103AA100CFBE7124CCFAC338421D614AD80BAE9CB2F80E"  
01255 "110E4049085B681DCA3E44C8561E548916F0BE26B967E127F4CE9BCC23BB437BC85FD060BD"  
01256 "AACB1E8F2EE47AD3AFF48", // PRN 17  
01257 "CA255F7D7F404B16F1FC7911712420551AB5B3F9F79BDB69C726E5BE2D99D6BCEDF3BF5319"  
01258 "D2692BA47D5D043020EA4D53D9B0094D4B9B75EBF1534C19399B7F4B61482B63D8CE0BD3EB"  
01259 "523917820D7655808F39D1D33853CCD1DA81E21757969D497A8047A2D7B1E90B969E13CBCE"  
01260 "FFA528619B83698A1E9A61801E430C25B18D3972E103647872C252EF4595DAA9F9D2DF35B2"  
01261 "491E3F8CDCE6E2EA7D0610D09E4D5474CBB17C12F0BFC424424B448CF30A14C80886E97971"  
01262 "61C716B7AF3F72B02CCB7E90C46EE4BEE1F85B2E9C950B969DAB140F3728882D5A162932F9"  
01263 "D41ECFDBA16F652AFFD0CF6CF6891A718E9FC2FE0B5AFE92106985B42CF866EDFFFF26A1512"  
01264 "52B5F5CE381DDF779FFFC75E7813898497DBDDA3555D16809086A0505F831ED47932120AFDA"  
01265 "68D423FF4F6D6DF66FB84BCFF787658C103379DA2F2DBF2868D8C436E3D555D148B6C51BD"  
01266 "E4361E8FEC932CE4C4D204FF8F10F936D1C780D7603C1824C222323E5D32ADACBE5510D6E2"  
01267 "AED520CD11560AEB9367D22D9F29506D407078B9D8AF8BEF3087DA339C9BE1412F3C16B131"  
01268 "81BDD8397D6C84791E9C3CE73AD5A2279515558B7FF69A0CEAD79D3890DE3F4D89ADEE6A64"  
01269 "E601D0B5E45924E08F706642C6EE18A3DEB0044325BEC020B12EBA41EA540976962ACA2ED6"  
01270 "28E917E7F18596BE3CC457DCA41A35A0499423C643BC3C95AB34C4AA2BF65E74079DEA268"  
01271 "1169DC3DE6DED27E7596C26ED0614EB77ABF2219CD294C9C4E80E09914038381D2F8B6D472"  
01272 "B770A0FC860A0CA2B27663A41E6354C45AF8DC90E5D12F28AD91A3200A626CFD837C563FB"  
01273 "FDC4C7B0A8319E8C4A4975CE54DF744412BDA3B1D54C44E59D24CF83EBBC9B6990A4B23A62"  
01274 "729500250184E63E1E084", // PRN 18  
01275 "CC8AE040A281117C472F7C87E108B575563E252A5C96064F9242F3D7F2544FE06DBAA94F68"  
01276 "47BFB89B3B6AA1C57F5A4E59FEDA18E365A24F4D1E3A8776768742BCA81BF7FA596A7BE09C"  
01277 "B190CF9FC5D937FBA0CBB6721B3A8B6860AC3EB8B39A19365DEA72D22DB146563106544DD6"  
01278 "E4EB47C7DC740F2AB1C5E44660D41ABB467237FDAA712BB7EBDEF277161178483A3E1D73C6"  
01279 "8E42113070E92F36C82BE02048B61735CA16AC302A0821BEB0EB4A4C84E425C61A54955674"  
01280 "4942CB0F6055ECCD3D4D56B1610BA7C6621C75AC449AC02DE04A2770E792B65D004AA9CD8D"  
01281 "FA96F12C5BDA8782FB08EC0E3EA957CFBFFF321D7C57F8DD78E4D561279AC33E522E33F517"  
01282 "FEB5BB78EC3BCD325E8C5FD75BB4A7362D9D995AD4B86D3DE969D06A04FD9C99F3DC4DA88"  
01283 "BA4F99D1C37C2E96A302CADC30A7DE63FC1F75853D2E8B15E2396CD6617EA77129D4D63EA1"  
01284 "D103DB0604B0153F234B48EC4F74E312D1582CD88CF6CCB958730106667A490DA0530BB66D"  
01285 "909FDD1FC725D104ECD8FE1E8C3F0ECD9553E353AF1597CC690BFE9E83690EE7DEB4EDB60"  
01286 "6961A1E53716075564251A689549AB1CF26C14375AFC82510F8E5EC467EFDA137D77A3640C"

01287 "87099EEE6052C89356C1A7F3AA2274F6A7E6485C5EF1E91EFF12202DF45C6F3CE0E7C0424C"  
01288 "141FC946B451A472A9139AE0E9EA500FF86F3CBC9EB021706038E708A32CCB11412B421045"  
01289 "737958216135D63B7364181F4CAD70B7C15D4D0DD8DB5BF2E20A5F103855A7AB306F26055C"  
01290 "658308036A72637696987E5D84B8664CF552FE3CDFB86421015170CE7F802DA7CCF8921096"  
01291 "536C3DA9237ECBA43A16A736DE01C595259AA18B858ED081183C726B63FC2D1372FEDA07F4"  
01292 "8C1950FE1CFC155E610E4", // PRN 19  
01293 "B7BEC67BE0FF68762DBA40B31D235BB9361A98F8D2D538FCEFF76E5286EEB677CDAAFCDCEA26"  
01294 "8F0DB6849E140A4B0FECACFD4D02927EB095437C2C6C2B2C065FCCC14CB580A647CF53CDDF0"  
01295 "9FA0DE8E5DB1CD030EBD0CEEBA2FE776649D0C61A690DF9AAE4961268E51B8E5E35F837E"  
01296 "F080BB64B21DF3039B36A621432848408E37C7066463352C21FFB88D06A2F87966FB35B65A"  
01297 "8D91BF29E70B3FD78F67C0998BE8622ED3C48EE6112FE4D6D176B448D5D55C52B60A1FA613"  
01298 "4CABA7E5FBC74CF4BCE85A1B6DE091899B7DDAC66B353DE01507A6B6F69E75A20071E94FD"  
01299 "57F0C26D48EF70F23C735AD1C21CB122709E79BF875E150CD269459EB5B8D328EDF5C34261"  
01300 "5965ADEFD6DDED2AC583B1F7834207949EAAFEED3E1C2CA2BE46F572DD6A344531B90634299A"  
01301 "09929C602A3012ECB0738D36B9BFE3625DEF26E11F3BF179A4ADD324AF1EA860993D4DE881"  
01302 "A4D32B209432DDD7D083AE4A37E3FE25FB7CD5D47430311D37B93D7F3191C91ADC0534E670"  
01303 "F5871163C09ABBCD72C0B8BB8697C62EE9BCD3169066A97FBEDAE909327A35400FA12F8915"  
01304 "E546C1830898EC814D5BCC459B24DF4A0C4E40AF4058241DB9897A3E62142E92A1F81BFBF40"  
01305 "B22ACE4A1F14B745411443B5A092AEE0539374F6DDF2FFF84DE33D76295EF6E2E8DDE63943"  
01306 "5821BCCFC21B39E034EB0152C2B5C5C3AF8F073A4A83556F6492B07C86534175BEDE22F414"  
01307 "A4E77012A46A78F9822A096246AA08880E43839FF5D4274784FF3E6240261A201CFA69A385"  
01308 "944C4E513615BECB948703BA86E36570C45499043A457256B0973268BDF6747F584F768705"  
01309 "6FD7C8060F2DA08440582B6325AA1DEC779381A157D267C163AEDC81194D415B2B041EB151"  
01310 "D7EE95286D3F53FB86130", // PRN 20  
01311 "F4CF29F794CC9CB418F513F3E8D26E34073939CAC74CE0C82C99505DE0BEF392D3F95BF717"  
01312 "6EAD5E31CFFD43977FB31FDC78884447A3B7344DA50CB56B5B47A0BA9791E27A45F41B474"  
01313 "C99F801996FE467FF80CCA7AC3B1B09BE33265BF82C4E88D2F660BB92BD75E34786977CFE8E"  
01314 "8DA9571FB56515ECC6D3240DB2A6FDFDACC2A273AD4BDDFF88BE8509AFCEFF967B9B57CFEB11"  
01315 "5B06B0455A6AEECC65C588964261104103EB7A78A5CE531AF579F8CA534898348042B14A238"  
01316 "C6FDDAF2E5F16FF9B2045252AD0C3DDAA4B8A890D3BB97B963A0C9F8134DFA0A21B6195857"  
01317 "663748BA48C2DBD2AD7E4DC0841E875B36892BF76144A5C242E969C19421C18D8EC219AF20"  
01318 "6A1349AA822E49EADB1E97DCF852C3B3A99AE10A5F090B0976E9DAF4F72C1D43C72F39AD76"  
01319 "715F80CB472B2A15B7FE555C74D6B808C1648D7377BE0E218A3DB00B405D6396088DD457F4"  
01320 "72209E387BB9F8C2A941E0C1C9A28D8B3FA18AE509ABC9F9F2F727C5698EE520AFBD786BF6"  
01321 "668A5A8AEA9F359E27AD48CEE9FAE878F2C91C43CBB927B761BBFB1CA183DA0A2987086140"  
01322 "C8241F5D37A7B304B3677E5D7287CD7C8AA4DBB0DF85AB25EBA59191E5CD22B06E389B2000"  
01323 "2B848BB6BC02CDF28EF6C985D048FE3BC8800269252380093385BAD2801701C9D4436F35A1"  
01324 "627277A6ADD6995E85A4620A940E3F9E56612BF25F3A0E17B5AF7491F475133667643A71F7"  
01325 "2D75613C1546CD722209114545C8DBE2CC0A1A57098F66BA20B3E674521F417D0716EB4656"  
01326 "1561E5C3CBFFCB94A7DCBD15B140ACE1DF47FAE637930FB665520A20323DF7C5812B9308F"  
01327 "9C3106CE1F84B814433A54F50243CAC9A3D24B29CBD981191285E4ED9E238F91B04600B79B"  
01328 "2DF3B6AE4188F7E60302C", // PRN 21  
01329 "D84B4EAE8E6A13C1596B2F0266D32A11425C0BAEDF57E1DD9C081F51A3F5043C49FB2D4DD58"  
01330 "1C8F76DCE746C2D4264371D459C60A0129805C7011CC57B05ABBB743F2EC90F7278A101C48"  
01331 "79D176BBA5D711E32189B11F6CC2419A2BF659FAD2CCD7D67B51566C623979A2EB8075F5E"  
01332 "EFC6E3A18CD9E705E73B948B1551AFC0AE902353A1510E7F3B0A4A446EE19F61DE93AAFC05"  
01333 "53027BE4EBA7C1D9C76575F4BCE1B3C8620FB781CCE777E587E587F26CB62E386EDAE3B0AB"  
01334 "6CAE5BC4F7149F18AB73AEEE44ADB412AF2849AF8C9D4AD6E9851D99FC206DBC2989FB554"  
01335 "072810705B57D3C7529BB38D5006D24014234D285C44F8246262C4FE9AC4FCB3FAFEC40DC7"  
01336 "51BCC9AC3CE82E3C3A69346D08B7AA3902317582E9626402AB299D9DA83D3D43EF13FDB6D5"  
01337 "343C36F04B9250E98481FBD8766B81D0B669CBADD4CED277D69E15A6B1ADC24EFE4F47633F"  
01338 "AC44F8D053B85BFFBA2301A647C753480AE5150ADF6854E79ED71CEA29A87D58E39B22AC5"  
01339 "840F1BC0187635673980989FD08F3EED4538B6ECBBAD4768FE211C9F5176D013545F5013DC"  
01340 "9CCT786FCF32D4DAE3C0D299940A9A3109708277E7F0A93A796D3FD907811DB4E824605600B"  
01341 "B21BB1A53A395F7732947BB0C68506E830819F84245A162151461DD9E2D6C37278413FDD70"  
01342 "B0442179B9CF44A7C360AF5A905E6B911AC253B0A3F9CE36794CCFCD4A3306CB0AC11A8B0B"  
01343 "E9320C1F5ABF18A9FA7221ED0FE3FC69CEBA0D93EBBD7B3BC663C3EDD30A0290F26A369E8"  
01344 "ECD629EEAE51F93F43595B833294427FA0F5D14A4D8E7B116C6B66C0A76612B09A4868A828"  
01345 "66E8FD4C760EEF4146DFB2A235DEA57B39877E84186A0EC98A095E977BD2D2C444BDDA8672"  
01346 "08924B8EDC5D7D7A65A54", // PRN 22  
01347 "8642BB604191809623A899925E0E67D0521EFC720D4C80211E7B7BB4ACF2CE3644922F59A2"  
01348 "6CB2584CB354BFC2AB83790C172FC1F2E8A01470DF0E0542422EBE3512D1A6526A439514AD"  
01349 "7B865168795C99D75D535DFBCE2CB4E83AC2BE79D15844C4514A505CA52324FABAD0A547C16"  
01350 "40683B50EBADC87B051E6A888899945BC15311F80DB36F4E3FDDABA13112F1DB5DF6D5BD66"  
01351 "C8C7C348616EB8579DB85940EC9BD4037B3BD4D40D536489A6107A22E8D22F9883B423FD842D"  
01352 "209FAB394616EBACA54F0056F00FCDBB2CB0906E593EF65BFE954CCABC460ED34F77B9A8D22"  
01353 "F09BD3859F9439575CC0886FB87A6196C506B97FCBD1EFFB5085EA4BC0C5C540E331CA7B4"  
01354 "98206604E3D1497BFC2996DC8A759A87FC6EC739B783827F8B5DC0FAA7968317B5C8D17654"  
01355 "1D64C99A3274E45F80ECBEA25CFB79F0BE881D8A1705406765202F13C6E7A2F2D7CA991D3C"  
01356 "85DC599F8C8E53AFD50CCCB5C1A4649C29CC31A9EBE57B2C346161733C23598277151E91EE"  
01357 "F57B51438E96B827B44F1CEE1AC2EA1FAFA2A011DAF21D76A125EEB85A5DE6DD6FCDF2D935"  
01358 "27CA8E0B53EFE5F73855140321A4E4B23A2C7FE6B2305CA6E33F84A47DFCB13F9945BA6A82"  
01359 "668217EDBC45F3958799573F7BC208799C3E0AB968469A2E264BCD36CBB86EF53AA35C0FDA7"  
01360 "115875AD39BF546830CF62D496968079A7EC5D61834A82D487F43E5C1485985F1C8C6A952"  
01361 "8E8ECC85ED2EE473733C2253510E86A781D9EA85C7FC9B39764016334EFCBE57A599F90860"  
01362 "2482E73AE0914D3362739C384EFE618C05F2899AF101CC695E57B8F55C88C0F8D3F8418423"  
01363 "9EE9AF4E14758BFE5AF7B7C3FC467A47B88BB03A2E2E4B51EA7158703BE3F4E7584B67F48C"  
01364 "9D0A41B5F9AE4D2BCDAC0", // PRN 23  
01365 "B9C6C5105394B0CBDC150C6A6961AE449F625D838891B683D87F05499063D51F9839480B55"  
01366 "9F9998E8286A4D756738B0DB9F31B9DD2ACEE5A5DB5C67A617466FE38DA55912793390B698"  
01367 "17D94E2F53BBE1254CEA7B87C49984384FD69EB341A5BE772EAF5E49929FF9E8096B190D1B"  
01368 "91F9D94EC3D28AA49B4A8C78F5614BB1EB4F42E21238C23A3F574CC5F805D2438AB5DAB4A1"  
01369 "ADC2DEAB9A86299EB8B1D8C51478AFD1EA7ED89A450A88D1AB04D792AC98B36C62D2C803AAC"  
01370 "319E0D2438DFB6495A75689D233C8A4044AA9D90AA5BCBD36A385ED8F79843C25F2855BC46"  
01371 "E7F5D42CD243ECB17EE8DFB388CE95F353FE64E53E6D206C0BC40D40A179CEE7C42F586B92"  
01372 "CF9E19EBEFAA8FCFBE5200FF5452BA22EEB894490BE371E200FF09E9608FE5F131BD6F1D75"  
01373 "B5F9CD91884E780788B3A213827CF1C68D31C642A706C018B00B62F1877E4AB97FE53FCD36"

01374 "473528982460200A33B392FBB67C2C2A036E1F855130A17E20B2506733451A0FA65FF344BE"  
01375 "4ABDE757A7DA2EECFEEB00D849462A3AA88BD2F2995C475E428EFB2F0F216239425247A23"  
01376 "59BCB41DB8C9AA6E9A7B3870ACBF8FCC21CEF23AA12268023F3FC83AF658E08804A7DCABBC"  
01377 "3873D0CBAD54436FA3F69AA993BB176363E08A55A346B8F7B33115BC834F4DD6CCD3E81A6F"  
01378 "140B0D007CD2093585EA67670180AA76DED02CE9C55780E19411F2EB59E1B51F569CFE6009"  
01379 "EFC9347BDB81BDE287DD1650EFBE39FDDBA0AB6270A4D31B3BFBFDCa6A64195798E63EB5F4"  
01380 "13837996C25FB9381EA5B87928B8EED6778487FDBB84CFE82896F139FCE0705E1120A72594"  
01381 "8A5841287C9E9010733142369ECF5877A41ED13989AA4E1732C64D587F2A114A7F3746BC16"  
01382 "96D5EAB4CEEBAC0B022B0", // PRN 24  
01383 "A1F6FA8D82B89F75B980A5ED804DBF43B0065A591EA773674A1A76D8CA272812F10FEE1DC0"  
01384 "B762250696E22D43BA240D170C1578DC85D367DA956358FE2EF23C21D526822F6070789751"  
01385 "41F078D14E8C146F033FB8CDB7E6D3F8EDB59F3B6198DEFE0F32E092AAF2A30E06D90A4114"  
01386 "ACD7B52F951ECE05C143D27E10BF13A446E53F8B73669FCCD7D6808261718A7DCACF96BE92"  
01387 "037B8560D9E6C2F93B4F573E5697FC3C8B3FCCCB6715277D29A78265AFD2BBB3F944166849"  
01388 "C92514456850E44BB206C43422DB510BCB62663E0A4D75858E095364F6AC8A7F853CCD9F3"  
01389 "C26D96DB2AB17F3BE4113B3496F2C13EEFAD5F319927E245545D4773D6EFC61989F2399C9E"  
01390 "66C01EF32466DEB522BAA45AE1FB784C95E2F6D96EB7A74A87619D64698538AF5485282FB9"  
01391 "22D4542E563F9B8D66CDF0F65E575F069E88202AE53945AE5AAE23A06AD33BB23F4278F83"  
01392 "92BC4664AA5C11695F6F51FD0101D3163DC2E895C8CDB479750AB91416D1A3F273E7E16F02"  
01393 "C712A2418EB5FD26CC9AC9FBF07FEEEE065EFD044408010CE24DC2982E7F6027067F47D2565"  
01394 "27F216F18AAE98C5846948B8991282F617E18EFB8157870278C16E3528EB565F814BD79CF3"  
01395 "912602FFC279CB0C7A1EDE1E9DAD9E461A1942FFA23F939D4B3764C9C3AB44E5732C9FC079"  
01396 "A09CC3C0B89538D4064093384F99491115FF494F1D2DE458B2700F267FFD9FED9FDADA0979"  
01397 "A20383A49945556DEC54F3AEC5413830EC09019615AA7D6BB175CA6C8AAC1D0E31C3E307E2"  
01398 "A32D33D9FD734300768DE19F41285886C808AEA633FE4E7B082F889AE6B46E59D6204B41C"  
01399 "A0B810BF8589053D6A44E8A8139329530EC9A396333AE7536BF6311493AD6AB1F530C4B8A"  
01400 "D97D3A36E2CEFAFC5430", // PRN 25  
01401 "DCBE11877CA625BFF2A41723A110FAD9B29444FE3641DD4E95271C2DFF6E40AC05255B7446"  
01402 "1B8080C83D21BEC301F942F845677531434A814898D391646C2037E701D50B019863CEE6B4"  
01403 "295AF42E574B2B78EA12824DF8C8C46805ED14DA4C741DA25653B901946FB3D95B097DC426"  
01404 "B094683B7ABF6AC6C3EF247E87146FFBCF47AD4436BD7C56EBEA788EFE96EB63B47D3BCD3"  
01405 "AC9729CC3592B94FB6DCA42F35BB92C5D51BDB976D5667D6FE10F143D4C5305449D81C43E8"  
01406 "4AA65454EE787B2A03F5D699D1A523A2B0BAB0D724674B8CB5B365A369A23A53F7346357A4"  
01407 "077FE422B202538FC0CC0311CDDC34E104F6738FE80F962EF89CEFF0E08E1EDF89E8F6C093"  
01408 "EA4049436645CE8614EED08E9D1A92B321F785C01103DAC7EC8DBC43FCC47AA9096742CDEE"  
01409 "C4460642D8912CDBA0A834457FCC32759DF7954C299F2F51EA0B5F3855515C1E9776728DC"  
01410 "DFE901A4F831BB45C8C66FBA13B4EC45015AB45E8370013331F26A71165FCD9E9D8C62D43A"  
01411 "FEFB7669519E78510039A2A8CFF2EB9B3B53A3CAFA9818948D9A9EA205BC49638D105B993B"  
01412 "23ABA544B4BAF65C67C72F19DE15F24E4806B4FF275C36C1AD18DD20A38E15DDC2A90564E2"  
01413 "26FA92C5848A0866EABDA77D6397C2BC7E7B6C416E85383C7F0102711E9599F4B0E7C06D8E"  
01414 "EBE9115E450262D387930725817179D87CBD563967E3E48F7FEE39ADB8B0DA6928A4040627"  
01415 "F2E96E71C726C830F0E7800671BA2C0511F757EDEDA7DDB69BB5C68817EBE9D7D4180FC172"  
01416 "D816F3C889A9986B739360C99F9FB8DFCFA6D464298E918A89CB09428E46C6EF1E9CBE619"  
01417 "15AA2B0B29508C8B3004658D09DB17E6345EA20501E4FDC2E27DD8F6B9025B776DB586E5C5A"  
01418 "6B5D5CD64C7539535DF54", // PRN 26  
01419 "9381AD3789E337802B6E8458CC2D725126B80D82366CD6B24DB7C4C561DFF33E612721F448"  
01420 "1ED6619C7A7514CA1647495B41F4E7130BF1DF6F87E0CDD1A0280B63D9791D2AE5208638E6"  
01421 "1FDCF744D916903C6E7DC3AB2F41AB27ECBE4309DBCFCF128BE89B37048D93882537A3AFEF"  
01422 "92F14F22132706FC1787EE4F36AB60D14DE2071FA11817BA9B22AE462593A5CED4C0512C5E"  
01423 "651E9080EC8EB439C8E48BE557C4875C9A9055FD72AF727EEAAD3774B6CB379A283EC7C0BD"  
01424 "F5ED0CA5982739A3A3471015A606578718B0DE21BD323980F65F40B0A4D27DD9FE696E36C"  
01425 "BDDC5350FCFB680DF966AF9EE72C71329387BE2B6CEA4E9ED432F3F53DEEF22DD3E5C55BD"  
01426 "C6B0CB7862CA35C2DAFB52CDF1CE7E63C502A05822E2578BDEC4CA9BE8E542CCA527120CC6"  
01427 "1630886981AD109C84C7C9F190A083BDD14CB7D4ACF8FAA5F2261780EBD89AF20870A7A40"  
01428 "1CD02907708AE12D2931D457975C073D5FB241E58B4BCB6601E908A3417421046942145D5E"  
01429 "1A18D33A92114D24F405F6AAADE9ECE8C238B5F45AD7DB223C8CD1113DD3151701C99A86F9"  
01430 "9BEF9671C1C061F6398FFFFBB90C4385052E9ECB102AE42519E90EA7392A7FC504B6E26AC0E"  
01431 "A7DEA08BE2EF7B85049CB6FFD7D05BF16504846D42F2C33254FD3BA377E4DF223200B5F2F"  
01432 "7B01476BA96778F7F16299CF88D1ACA4F7CEACE14F533AD50EAEBA63D847662082F445CD04"  
01433 "1F7D563EF2175486D3F5BBD6F2B45620A9631C2355DBBF0535CB8FD2E9CC1E7F1E2632E31"  
01434 "39585B6BB27F51C37A57731ACBCD01E87EC284E0EC1FB18ADABADA1275C4A2E45BCD325197"  
01435 "7DFA9901FB524692B80AE7A0DD9B6CA604577620C71A2D0B0B110355C94111554E251658D"  
01436 "460C194D5C3FD982A7FA0", // PRN 27  
01437 "E5487EFD24AD377B33607300FCD602F9FCB49730AC30D58F69A915B7912B7622E6C69A5181"  
01438 "10787CEAD3B48AB2E17AA09CB7B3F21389F977F0E0D6787AA4EA8501154B6878FAD595B7E"  
01439 "A7EF437FE17901A017E3725B989A81FAC1B52736AB672D6C5BF3BBE86193F62919A033CA0E"  
01440 "A4A2C351791E4AE35F96435D0DA571908BD4D63D0665B34A138E319CF65C6A65E1506178F9"  
01441 "77E3E62872EAB30544152F48A4C314B374064BEFF9744EE0A6EA9FE0AAD46C5AE6ACA975B9"  
01442 "DD28C66BA09F70FBE4BAA3C626CA5BE7791447E63BCA07DFD24864D0C36ECC3922761D18E0"  
01443 "574F01B3EADAB4DADCF1CBA8902E405279CF31B23F21503BF58AE67781AB39D7B9C4AF1BC9"  
01444 "2BD5AD7097A43DC7C0C6D7E9F0ED15F7395B3C44E4AC78F776256A8168836C881DD0328EA"  
01445 "E29DF99890551818063B9037652C5847638E111EC0E97219442964B49453F57929061E14A9"  
01446 "C7FFB0DA28706C29339B65DD907CCB39DFE9366FEA29C4E82E827019201ACDB48C1C38D5FF"  
01447 "510F289BB3F2C36880509D8D25D1D3BF956610AFAF82EA89558289E5FA6B137FA6D8479E1E"  
01448 "BB0A73C0C1045FDF81C8FD9DCCC4D2D7769A20B939D5643B6BD7B5039BADC90A6DC96B0977"  
01449 "23CC0B3FECFF448CBFAF438D1385A2EB903F4CB6365898D67D4ECE9DCAC0DA810A772ADB3"  
01450 "77C73575B74A081D5AACC41CC77D9BDF16FC80B55AE9C60416218CF62C01A074482ACC916"  
01451 "E3BF11B72C73B8E8EF6801B6375DD1FD3D5861E4E71BCC0DBAD002A488E43E7A83180AE69A"  
01452 "7F7D24B1285EDFB325E819FA644C49671B8026CAC3D75255B9B068C15A83E55D0C09C2104C"  
01453 "72BA1EC84DF143F5E10B9B40470FB4E57E124ED67BD3B673D6DFA785D49375201033943482"  
01454 "1DADAA22CE633C472185C", // PRN 28  
01455 "AE631B48FC62F9B90929480C17F48E9E5EF285B204909B25F8933876395075BE0A02A78862"  
01456 "9CF657D7117C5C8F778E93A38CD7D419C6CEE706410A33920BDBC8B3F0C0C21B27028B10F5"  
01457 "A8FDB4C35F1F5795451CF52D8928658A0E692BD799A1BC06D0A231A886642483E5D3CBFD45"  
01458 "98C3D4F64A1FF623B159E8EC1BA3B66549347B5DE15CB9BEF1695BD58D20B832D49145EB58"  
01459 "F0141CE22477820807078EE41BBA6803428C60AB1E2B3E1368FF0630610D4604EDE0B23D6"  
01460 "EFE2B4311287CB5292F4A212EF01DD0987884D12F4317F0506A87B8CFC8B6F27F452BFEBE4"

01461 "A51DD69FD0679C10633FA7F19A5558D417893BD6B39D29E19A2302A6EC4A9BE6935D7E8118"  
01462 "721AA090C69C990A1C086A1E33D57833C9B561902B676641632B1325A388B76439EA0EE5F0"  
01463 "1FA53D93CA61670CEFFBD5CDD5D58367BD6DB99C88465FD9DD1CF5A94265EED2C6524BAE036"  
01464 "76B44127CB7E4908697BF1CE93F53B2561978A3ADCAE7AB80CFBEBE0F6E88513A9885CD83"  
01465 "2AC0C3B57F9B5798F672BB60601A871EA105640C09A581C68D5525D4977F8123A275F602CA"  
01466 "40D5CFB050644F09B2207D6887E74C93696CB5531435A0D364FA0649DBD0B54CB730BC31AF4"  
01467 "75C28792CBF67491BD84D63754E37B1EACDCF8EDD095E753266A5824E1AC77C02A4419049"  
01468 "93405FEEA4A177D3A00453456782E8ABFBC5CF47EA237BBD6DC7C5E75F64C3CEB478F250064"  
01469 "36FCB0BD768674257740CC383FA717B77A3B384CD127372435643A8BFEC603A38BD7979A8"  
01470 "FBF5C6614772580C4775A4879BFBAA0F93FE37E77BD9AFE9351B15FA37EB5AAC87356EF4D"  
01471 "3A9992E3E525D9F454B919FF892E22C13B2D93D9BF197F8C6A93375DA7F6E5630A48CEEC74"  
01472 "AA783CAC2270582AA2A98", // PRN 29  
01473 "940AB0FBF5D34D53D00095DECDE97492D6C806AAD743760AF951F9DFBF68B9AC4809CD7D0E"  
01474 "F7998944E5C16D60A54A12093FC3C802A5F67016430635BFC039403E244D59E6C02D5C10E3"  
01475 "ABB97024D451037AA52180A23065AE2E74DF1EB04ABF4548215EFE95397B86D3C8756DD9E8"  
01476 "30F96FD26B5501DC3A53CC1EBF1EB1DFB90D6D41B292A8D62AA5C8608949CF90330986136B"  
01477 "9A5C0C0B2EF661DAFD21104B412C8590621C4C65D6099EE25F6E2156290EE1C679F68ADCF5"  
01478 "E37C0C0D19B78C5E63E0D6469D3A9274A7058304A05B3EF796C1E248ECD7AC85434F02F"  
01479 "8E26CC46AF67832A9A715E75EEF4CCA528A07BD57B6CD51DBD47D357843915C8090273011B"  
01480 "8DD8ACC9D51123E4CFD8291634E4990CA6DE35BDD0051612DBB61C5A24C97146B61911FD4"  
01481 "7E4D54D30BE406D5A89E29BA2B8772D2C4EE9770ADFFFE8B1526193881D5F310435D34A9E3D"  
01482 "36D322FB79032F2171F4EF2CDE53F58F76135F78BE0C2C76BC9B8E7863A4482A6F37FEFF27"  
01483 "E2A61F3686909F171FFB8629F86C5EE353C9F48C6067AE66798B0C50058C8ADDE3AD5E5C736"  
01484 "C8AFDDDFFB67078E526E1BB31C28A5EADDC674E65963A1B57E361B60B964E3A0B07267461A"  
01485 "C738E4A0FD14F07D5C5757532BC2F4699F1F9CA4541B41CC8372A6336D4AFE7052359C750C"  
01486 "E0CB866CF15C01E969F8635E9C73DBD3C77778C424631882B11267663F4DD24CDBA30942C6"  
01487 "57C894A1FA9737C74091ABF858214E36FBA12CA159BC40A87630D5AE51510FF75E99E712C4"  
01488 "DEAB587EA8D1D95A5767205E78923F0A6E08B2CB9C9352A773184FF2653C716A12E6BFA440"  
01489 "4E4533FAEA2F40BFEB6A5FD3D34A7BDADDA5DCB008F081C45A90411F64DFDCC682FAAFFEC2"  
01490 "42B90C75E1332EECF6BF8", // PRN 30  
01491 "AF07710836A5AA122AF190399AEBB0ED49CD908A821EDA0C4ED5D6615A431C42C6A9FACC9B"  
01492 "33891A1C58FC4FD786CF7238B95F3E0512858F9BAF2A30C9EE0A13219A693EDC046050DF63"  
01493 "8D41499E626DED085D9CAFD687290F19D628C29EAC839C65EE008F584780E2A2F99DA2624"  
01494 "3009D599438B05874592DD1BD44E4846BF6ED2C795DD35F4E5DBB1251D63736D5528A86A3"  
01495 "32E1FC959F10F99234627B62FD5F773635D44858040173703E6C22768F16CA7B2BE2FEFACB"  
01496 "F7B4853D71F8A3BC42C0DD0DB67A3354DC747CDD4BD0E5BBA53E72E6EBEBA3AF00583EAB1"  
01497 "ECE8BC49876DD3C1B51E20E954FDD7988FBFB54850B098CFD54A2730F46B80148D18EF2BB"  
01498 "BFF87CE8A2CA0C61AE7DBE15917F506F58D81CB2289448324E5C85FF9FB2EC6DCDC0DD27A9"  
01499 "FE8147792D9403562F171E9C400E253E10D0C004235D3926836DB7C44B57C3F9FC322E6FD8"  
01500 "F22741A9FEC04F8BDE90132099346A65C72034497EC1BA5E4DB0A5E6E92DDC4AE8110A6DBF"  
01501 "7D53DB059609B18D590E83DEEBB1EF8450B5FFA70D2B19A78BA53603AC17C89D39AC8DCAEA"  
01502 "1DED7BDC381F5E928F425CEAB22EB54E1A388550B9B5A663F273A1592347695E2AD4E00CD4"  
01503 "FF240BFEC137AB050C39F59E2359953D185863D5D9E2C9A299317797F95B77772D8CCE3B09"  
01504 "E9508E6806535753B842A4FF30C6B8403F594BD5AE77FD85C6CCF25C82F453E8E5A3FC337A"  
01505 "54F801632E4E3A8C07A6D7C9019746A1005E3BFEAC3E4DA6D5598AAEE0F468E8DB4919CBB8"  
01506 "2895779BCEA4134F13F974719614D00A17E945CAA70E64720E96A9C98A0C37531E36388796"  
01507 "D27A694E8A430BBC546CE164B180BF8A8F84B675EA1B85840694479DC7B1760D89D18BE957"  
01508 "E643FD272AE6700A660C8", // PRN 31  
01509 "AE35D9E0DF692D03A1E7A1EEEDA307513BD85EBFEB24D747342C8E5DDF399DC1E9D52D36DBC"  
01510 "70B22F39D20E4BADD58045171F3A9F372429B7F5162F76823FA7E04FA22CD4054B43B77DF3"  
01511 "4AAA4853DA9F942ABDE11CA36F9400B0ECFF00B5CF86F05C9657275A28407946D02A7576D8"  
01512 "8C5273B2AF09D573866D9A1050922A66D0D8D5D9B37EFD8EAED208410F805E1C31FB76510F"  
01513 "6775D9F2F4C0DB4DB9FADF52342E9D98E61E7762772E6D76B6AB2E729A4F487E1ABB49BD67"  
01514 "5409BBF176C1F399B086EF42F817D99354F68CF4985A8C0ACB7B3FB70F1DE8301F46F702AC"  
01515 "5A58016C61BA34DA5D02CD681327827C18D98F2C80E8BAC04399EDBCAEBE3ED7171391BF53"  
01516 "16E62B972DF64F9BBFC1B58AE0B45C6ED2A59823229FA858EEA2ACC42A6F0D5029176EEA0C"  
01517 "206757DBA17D23411DC2E210AF343875E36CB18211F4BD3E22F2A03F31201C2D9D3255AFF"  
01518 "04DD3D6D46638E7E772D033978E7D5EC2DE7EA1F6E0B1E15C65A7BC31B51D265E9E08F527"  
01519 "E4A5C89B249200C00A1E8A84D0A0A2847DCD531FD52E44402394E2331808DC2E4F6B70A7DA"  
01520 "0DA13B062833F6DD8EF3086F2B6743A41DEA63DE43D94595F7270F670FBADEE74E6036A444"  
01521 "13000C0E9CC2742964270C9F42BAFBA890D287EC5FFBC69F9271022F7E3F251B4568BE358"  
01522 "4AEE97C77014E623232435F4FA98F4407535D984CCFC1F73ACAB60E06BF8870065AB22591B"  
01523 "38E6B146A535B93055BB97C94C9B07E2ABA263C41A90F14B9B7DDE090DD04946886235B0A"  
01524 "4DDA152EBE76444C3DCD80A9607872FDEEC65112DE6B13AA2B86230B31DB50566570115CF9"  
01525 "35C5E47E5872A8E97A0873273F6090A29E30E94B239BAC1FDA20BA5BEAEC54496B0FDA2CEB"  
01526 "6D99E901F169EDA8AB920", // PRN 32  
01527 "A93C92EE0E6313BB969005B1B7B56D1CB117EFA4357F8C5A0DCD7416ADC54DF5C0565DC44F"  
01528 "CDDCF53BE98A2B212177BCF3A0CD287147A2C5A9A6D517A0A3C8E1373A3F5C2B1C677F0141"  
01529 "2DB618F421B51EA54122673ACDC72F8C3BC3CE4D50FFA1BE0A551F93F03AA2F46DEE94068"  
01530 "84E571213A245CC4025F77214CA3149B192E1AFB80D99411590B1FF065C264C338A399B5A5"  
01531 "973CBFB3CE1A8DC0223E2A3A34B02C7FF74E82708AD513862662755999139BB860500AB3E"  
01532 "B4BDBD5F34103B23E654429E0E778E43CB2825F8FB7EAC8AF50EB6B0EA25C8EB576F658CFF"  
01533 "0638F9929E02FDE8CF2CF51EB574698EF58C78B979D8F2B0F6DA53EDC935860156DB22035B"  
01534 "47F307F12D39382534007E6233766872D3B61FA8748E2ECF21A207ABA8D3DC874C24609D28"  
01535 "A876EBF929D459422B1F2702E1DA477AF7B1CC325546C268FD4B7DD683EA2E80728BB40E7D"  
01536 "27CC1F8EC9992E3FBC51FA5740C8EE993A3F209EAD5D076DAD8575E25AA2247C886FCEB12C"  
01537 "7DBEA25CE2208A23EC2EFFB4412FFB9E97530E9EDD04F52E9258DF64B2CC474B41F209D9F6"  
01538 "85B79D63A78A9EFD06F1116C79C9B0D8003588BC0846C59D11BC1524DD1E2B23C17BA1D3A4"  
01539 "2E673DB1D0E514CDE95536F99A067E0E00C1D15458798669A604A5900B7BC43AFF6D7361B9"  
01540 "ADD88B5E6185776148AF75A57E3946D1C0376E072C910F4466F42F8D67BB1C5F27530E2423"  
01541 "4B22F8E1A96A0ECE1F2995A6741C15C91600E86F461BC64F2CCF860E524F1E271EB6D9F29B"  
01542 "9C1DD54F5AC97706B72F2DDF903895E5127C581D648341F86185BE980E6D14ABAFB7AF8674"  
01543 "94942855B31BE2FCC3DE9229EB21B24CD0A97D2576B58C9C80D732B56ADD6507F555E10"  
01544 "07CAE0A537F82742AA898", // PRN 33  
01545 "D32684039605FF18C6881E9A5578FB91DEF05935660F2B8921F45459826EEDC6BE3CB55CB1"  
01546 "48992E0E66DB5EE7D952914C645FAFB3BD033012D9B7F4EAB1C752FCC20A9C9872122DD"  
01547 "725682145714122A5CD2AE6DAB9EE24CBB6FD495A7444E9D89A90721730215ECE633E627E8"



01548 "FFB23901B44DF580E08DD1912B6A1F0DFA05802E144296248A14009E92ED71091CD167E9C1"  
01549 "BE1E67BAAD6ED95780026F491BE86FAE681282B6895F5605559A1520FC5ADDAF49F0E9FEAF"  
01550 "66BA20F06224FF1F8F1912BBC4A2A425DA6DB0B1DD76A7E34905FD25C3639A6465746D73"  
01551 "7877B7C97433F37664B7BBB36BCEB4CA552ECF10D2F0E096454E201C88370A19A372BC7B91"  
01552 "B402FE12F9BF79B479A1178263E90AFC48AFCAE32D604DBFBA15E703D8C0C62C7D0A563392"  
01553 "075EA8179F62DBAC26C045E7509DB7DB70DB5CF924AF5A0A2762ABDF27A6303E312D01275D"  
01554 "9CAA9CD4963C8E06AC56C7137C1A49178C2A69E269040C23880DEB737CE0F63548994B2FC"  
01555 "48C64788533C040FC1C31971703893ED62C393CA4BA6DD8C0D5F99DD8B010D2FB6EAFAC80"  
01556 "FE292E090EDA2B440BCFCECDE414890D284AFE9D3233D7E4B50F4F6E86891F90F54A6BA768"  
01557 "068F91BD5F543EDBEC049B8D825338B7B72B66054361E68FE6BF23CF3844831A9F8C855D25"  
01558 "115C3D6B740B17E63703E133177F825B6703C436CD9AB2B8172A8EB50037E3B839EDD9EE5B"  
01559 "DDD483377EE2933651B0A83D2759634CEE2BFEB86DD31206832F166C61458EE956C739542E"  
01560 "D8B7BC646586C0767365D55E023ED453F2B38D714AB85F92256BD92FF2BF934CF375F3982E"  
01561 "82A79C87F5B04B4C04E1B57E1BBA070F14722EC880FEB5A3AA4E3FC0158DE96D916F9C9763"  
01562 "3BD2742DE8B60E9D8C264", // PRN 34  
01563 "8C3627FAE2FA9E2B3B7E97B0B3C62BFFEF95365AD049BCE5AE8AD9D9E57EBC427E48876632"  
01564 "5D91EE1A3AF9ACBBEBD3875AAADC7561097EFE37D6DFD8B617D2A5876FCF968DBB1FFAC4A"  
01565 "F0113FE801FF722CE1093B8A4178035344E77564835D4C8C48AE00C88FD9B9241E272EDBD"  
01566 "72037C45A4A77EA7670C82B85AA2A8CB7A0B7EDB4F3B39ED7915D89E543486D75C01D8A659"  
01567 "CEB6459F3E3E152656FC730D40B604A9A6C8B02D069C156CEA43224B4F9772A53AAA44B4EB"  
01568 "E3E48A8E222ED3A01AF7254C831145C07D686237574B2CA0B9CB2AFF30AE70AF2E97F10B05"  
01569 "44ADECE62641E404C0E264C4DAA5B917AD687FE91B932F92193E5FC87649CC1D5C7C8FF3A29"  
01570 "26B9255AC9167B6BEF8DBA7795FFFC0E10277B68D5554D36FA842CB2F2F1C5760934558D"  
01571 "403161EA0B5327F2345667D2B3FDD8EE06EE64E9025536D074FC134E2A5251A15F0F7485C"  
01572 "767628531A02C62778BF2AF48EE3DC5092237E46430ABC9FC89F015A249427A5B935C996B2"  
01573 "63E15CFE8438161DEB41B043D3D57C63B176211D3072086CCF27050A6494CB6C3FC2E72EDF"  
01574 "8958B5EE5132DDCC3333C6FFDA08985E378450FECFD242042B55E1F2AD8DE97680B2C6C1"  
01575 "7FBE5261D9310CD9283AB1076BA982B563FCDCB5E922F13659DF1AB2E736D75F01C1A88B11"  
01576 "84C78706473BA1CF18829E5A93E5C04FAEE6D953B1FE1434672A452921B92062F440E31E19"  
01577 "441E838E38E0D3A01AF7254C831145C07D686237574B2CA0B9CB2AFF30AE70AF2E97F10B05"  
01578 "8402C23B523772703C81ACFB8D698F875F1E63B1C7D8F6DA7BD96313AC8EA9DC004D01892F"  
01579 "A683D16FD9735B3E627167DD2DA2F74628555F222626A825DF2B898767968EF4C46074202E"  
01580 "365153035291591EFE810", // PRN 35  
01581 "8AF5247C4E1296ED850CE9D12E5C1AE78D3CDBFF66439B5D9281FED33F3D505DFC1718AD11"  
01582 "70033DFE72FA0B0FA0C54163A58E721118E058C330E76485AD282FB57E7D0BAD6C5F9E4D2D"  
01583 "F4CF27835007230E3D49501C1277FE9BB9F321AD4C292E0C9E3B345E0ABB1FBFB0A81CB406"  
01584 "53AF6E3CB0AD6960A10B5C30EC5B1B48B9B0D5164E1245355F7B1C54B8CBC2B33D2C6FA59"  
01585 "7FAB6AA5006167781FC8B1054E284FC8ACDAEAE92D3CA520753BC6F88F50A0A516518B6182"  
01586 "345E1E7DE9AA215CF4FF299AA44E6A76DCE4B4A2562B961DD118151EBFEF12E3231CF29CC2"  
01587 "EFC0502FA4D70CE6626309941070BD0B91C9D947D794A304027B85AEA9E1A31542B3673289"  
01588 "B112EEA96220422045733CC5E0EB951B14FCB431B13D08C8AB3AE1D500E298D4163F2EFD34"  
01589 "70131A949BD2F28EBB21572EE2CF11B54595383C25518BA4CCCE91651CA41AE5C75AF4C358"  
01590 "9F331EA16B7BCDDCB0A036C581741F773B9E94FF4FEE160A26CAF7107EB0C07B236B11828E"  
01591 "8A509C5207CF58B34ABB6B77D80827FB4CEEEDDC09649A740F316F9206A28C3A0A8E97112"  
01592 "EEEE7C70A9A9EF04563750A68DFF06DE3D953C56677D03648612FF7BBE3BB46B8D08EDAFED"  
01593 "E427AF9E8E087399E410D2F8DB33772F2371C4A9E57049F18CC2564E66E2158EA9D77B2939"  
01594 "1363A86B0A2A476A6603CDC0E5D2C8E589353C893487FFC1A6C1DC22AC5560C65DF4ECC2C9"  
01595 "7EADF3E87BA668BA7208253FB170D5F1B570FD027859DA05A17B739EBA433E17A3A2312C71"  
01596 "529DC518856D687EEC2C245E7CBCE19F697253C9797029FB7E12079DB4715BD6B934DFE10D"  
01597 "CFC083D1BE53F894570E052C1681F44B77FFC4EDB0E66CB970AB141F35D3CFA0484EE2CDF9"  
01598 "5702CB2D9F58FF6325E30", // PRN 36  
01599 "FFF67C13E26F4604239AF32CA58D4EF9007ED0FAFBEF5C5653E8943FB34338096E78C87854"  
01600 "F5217019C43B5B066DFEF9EA6BE0F57EC9E7CAE8CF2CEB9BE04F62506F08A54402274BFD90"  
01601 "ECF1A2E6694E18CA9A1652BD72D224043B50F60D31C762852168ECF0643F817BFFC819BF7"  
01602 "A9AC8F8F0DE24E2F00B81E4BF75B284AC9A08836AB656BBC8E75BF2B1B30A3B278B4270DA14"  
01603 "04CA821E83F3035313AC9BD8488E776C70BAC06B1089C80BCD3FB9967BC471E312CE7CA40B"  
01604 "91C7E4839F63C5A8D17EBFE9355412BE059F8C97E81D2D82B46AB626740C2E30F3DC32CA"  
01605 "561118C4ACD57441514882F28E580D36FD8827BE96633868BF85AD1DC374FC2720B6AD70E"  
01606 "CB0622F95A8212D2D8D5494F2509F7C152DF33466B241FD9F1934C020C4BDF29908B59801F"  
01607 "8D95EDC2E87693917B042E4E8374AFD33E2472CBABEA672175112CEF5BDEBD151DD6C688AB"  
01608 "6004B804C19443E0DBAEAE8AFD4AC26BB51B57F9FD08C6A38801CDA3C5E9387DC3AAD407DA"  
01609 "7367159675733460AE1B22C73346DD412473152D08C90B7561DF94ACB36D5FC67129CA8B46"  
01610 "99F4E175D88763D82D93DE0D7DCC13757C925663921A7FEAC8B6AE4B13A6CFA050CE7FBB2C"  
01611 "A7E741B8C0F3E218CA356DB8E74019246526E4C7B6C819F8845528C3DCB71CE156A98E9861"  
01612 "DC28E939AF89B0BB346B6F0A91B2867A8BF9297135101EC536E54BF05BD379EF8A3E5A8A"  
01613 "523ECA0A1525FFBFE5514BBBC3998B166ECB52F2E7D45B6F65D7CCF95D012E011DB513D5DE6"  
01614 "1545B4E904B148EE82800D6126DD7113CC1C8657026677583A7F041FDA18ABFF9E4D4C0CD"  
01615 "3E082FA0A925DDAC6278E533972392FADF66C25C19992DED31CFD4D216EAF001BFFEBAC525"  
01616 "2C88BFB905D1284BF2314", // PRN 37  
01617 "B0643FB1F06E1D5741F2F9CD1FAAAF3C3C010014E17B6DEBA1B868357E090C186F1A7C34F4"  
01618 "2AF8134DFD4F6BD301DD6174CC59DAC54C98F93ABB7C1B74DA21F28DFFB5715692AABDD8"  
01619 "995266C3F72F52250BA6F48B0D9DD474982D7A6A3215D45E7CFC11CA95DD083F8E8FF6D26B"  
01620 "B6D9D328E491CD6D73EB4E4FC98287BE31122BBD97C90AF9CAC6349A4AEB19890D5E2B09C0"  
01621 "214785F523DE48CC7897B3C1878EFBD9073DDB5F06AE8ED67CD8C0702DC362C73D3634A81F"  
01622 "BE7D458A8C6E71F0260BF4BA6D8E2CD8E265582BA270623212081D284C7BD6985713D80756"  
01623 "FAA8B9BF3D2B6A8F362013EC49538C45221E001EDE5E67D31F9A0B0775EC4B003CC08409"  
01624 "532E405FA41BCD1B76947BFD99CDF6C7FDB07D34BCC23CD67B464A870584458C86C2CD4998"  
01625 "AE7F14BB2B3764CA6228F6F3C26BF17E378A876D41EAA0BF46939BE3001B9A6DFAD5B8CE54"  
01626 "DC4C986B5B1B1704BFA06843D7F7367DCB5E53C7FFE2105703F32FF014F6975A2DD4ED209"  
01627 "9F606AF1BAB0C796ECA0223F42DB8FA5C38D9FC25A8FD3027613E14272632C757A2A255847"  
01628 "D1C39DE2E4236ED1A78C3144DB5DFF324537E696BF0904094CBD6C8D6A6D7E11715947429"  
01629 "FF836F10D1F6E8CA45898C4AD590E13DC1B5B7BE3C419B6C418F6BECD3B130A3884DBAAA48"  
01630 "B53AB54112E8E4E9F17F3448D3ACDF2C2829EC14CF68BC184884AE408743A90C92FEB9F4D0"  
01631 "877931AD2BB848971C061334A1E402809E3ACB289D16C68CEF364A1A75B76E5A8E32F3F44"  
01632 "480345D426A2D809D9771A6FBB5A666E5963806A91564DD62514E444861F0B81FCD65A8A59"  
01633 "A8A63F98E4DFF242838CAF22A6015FCB8AD8164C1C53E7D4F39B2E8D4BED0635FA7514A2D7"  
01634 "73A168D4EFA2895151608", // PRN 38

01635 "BE8B93ED7C85573DB5EDFEAAAB80312E3C3225FF6B65457F9833D7666DA0B8180218459269"  
01636 "22A42D3DBC0E19E6BF4FD81111DA2D159D6EA1CE2DEDAAD2A3DAAFFFD91698AE157CD0CC55"  
01637 "3ECA1F4956E667B3D96EB940D33590EED2B2D600C1CFB3E9E918EC2C90271C5B7842395"  
01638 "D22DEE66B6E16A49F698DABBD8A51210DE9F22EFB2B2CCA93C6592A8D88DA6A42A4A4AB06"  
01639 "2674538A1EF48EFF39749BA22DC8D40F62B670828117FC09E5537303214249199EE117FD69"  
01640 "B94D462A6778B44AD2A35A03A1137D0C0023863BB92FA4587078B232C99EE4278F812AB869"  
01641 "A41F58FD13175D09C34FFF79D08FC9153DC3C896B61899F37E31F4A9C0B0EB9DABD25EA89E"  
01642 "A6BB91389BAC5274300B8E27E7354CB8C3C3872CB7C714735F41F3C2FDDA43739A5C1E2550"  
01643 "59E6A1E44BD6DE31B63F4BE586C3B2BC0A66F93470B159857A370A5BA305584803C0EE8E02"  
01644 "C6A8C237A3A14344EEE4953AFB13DCA779C1893048E8415138C7E64921C4B8DF266019EB6F"  
01645 "B3987731AF75B2F07012E91F666B982B3870F60D963A901C2872CBD029B8F0420663F65CEC"  
01646 "7C573A86FADAD25BDE5BC89A21032B7F164A228B2793096ADEFD1A76209E4556D7AD8C2B41"  
01647 "A06E2E1FB89D841E149D311D8AE0ABAD26927A170A785E25DE9EA3DE6FC51A21201ACB15F4"  
01648 "81BB891D1DBSEC7E6F9AC61D1F58C828ED672FA2810CB3B551DF5DB3846DD1DEDA975A308B7"  
01649 "BB07FD9F8F3F01D95FFBDCE2B45909F61685BF2E0C8C1D73B90ED6415300C23AFEE172DF3C"  
01650 "355ABFAB0CE384929600BAF569E3C9AB65EC12C07A79A22FDB8FFDB2009E3D51C7C676A652"  
01651 "7909E0AF7D9343324C94D361BF6A862C0FC2B4A4E8C7C5C7F36B2F5301C00DCC9635081186"  
01652 "C2BE643ABEE415B90D1C8", // PRN 39  
01653 "EC3B51481481486D8823AB29BA36FAC0BA96AD14982DA36695859DE864C11478A630433FD7"  
01654 "6A4505A2FE3DEFF8027EC93A9CCB3FBFAF4A5B0EAEDEB867280E7482386030068A9EA8CFA4D5"  
01655 "22DF49818C782BEDFD0C5321452E3B6E883A153CFB276CB0327F141500ABC3CFF5C83C19A67"  
01656 "B66701714B4C48A562F296FEC437E2431D98098CE25F636E399B870A59D83229F4753F67D"  
01657 "BB895A3E150551D0578EB4795C4AD5E027317E63AFE431B0CBF3BA94E9835CF072BF91369D"  
01658 "D7414976712F226D9A005F25BF427F8EC67C8E0C82A2060137DB850722A565FC2BF7392C2C"  
01659 "8C687E4B914E2A30CEC484CCE9360A495FE352C21DDDB503E7DA11AA96AFC3722FB83054DE"  
01660 "DDC11959AC20B7905318FA95D080E99D14CD7A6462DE93856E598FE2FB30679A1BF42F173E"  
01661 "08DDF79541B1932B5AD345C974BD025EE5AE9AA62F77C0C2D6DB056FFCB5B5BB91028DB44B"  
01662 "653C8EF3D684E42C81E2B2D84A406A3ECEC4665737244D390937AD551AACBAB4FA3EADF47F"  
01663 "7E39E30F0656430B5219B6078B095468323187C4A9176708A240278869EDBD54BECFAF13BB"  
01664 "FD86367B748B0FEAA48AFF0DB1E2A4D02C4C8145FB259D8411EEEB6CF3A96B85869FADFAB0B"  
01665 "5C2197E1DAFE3CF16EF6A56B057CD5463011B03BB810E26AF4355DB962F510BC71F8602D8A"  
01666 "9DDD4244772DE59C638FC3658C64C600CB69EF3BCA2904C242427DF431D9F47488FF324F38"  
01667 "8B5BEFA5B66A977FD9809E32A9EEF99A944612F1D65DE6794782E278CC8AE9EAC377C5A114"  
01668 "B0150678D40A7DC7ADE021F289A657210AC0C84B5970D56A3629B6BA7B3EA98CC9EB575FD6"  
01669 "FA210DEC335E82B1963FCF0DE32DE0AE12884FEFFADF33F7AF874324A712824A37E147F7BE"  
01670 "C32238C0C45C562AD648AC3756FC21D62BFDE47AF915F51A34202A994E7222954CBB1EC3E"  
01671 "E7F6D691A53BB2E692CD80BEF07BEDF47B02C2826F1373AC50E317BB57183A935121B913D6"  
01672 "A8DA1AA66A749FB25BCE08CEA10932F49077A258EF3C9B24DD8B265BF14E4B1BF072CCB833"  
01673 "FECDB250B400F1D1B6F4BDBE460CAC0BD0009585BDF11609EC8AD8C75A341B5BEE2A55776D"  
01674 "18DDA2AED5CE66AD03386BC86F0369E529203BAD8940F5A96E98198D94BF6B9E6A48F5B7D"  
01675 "3ADA260A19336590BF98861722098A4CDF4CBF35E8C55DC0D8D0465066AF954654E76AEB69"  
01676 "F0EB634DAE10F1D1B6F4BDBE460CAC0BD0009585BDF11609EC8AD8C75A341B5BEE2A55776D"  
01677 "9541E3E1F530ADB22DF21C92DC17F7CD289F5CC24EAD69503425AAD76CC14DF302993AA842"  
01678 "D40121491EC8BC040427D7DB30644881CBE1F3BDB93444B46F42C001C1144D265379B6085F"  
01679 "C1C73ACF04A81A1302F20", // PRN 41  
01680 "8C296D08BA20482B2C7E11D7E6959FCADF027202B6D92C4092940D3FB268653AD99406B5FD"  
01681 "1C920CC3BB11D44F38F23F2B88052C1E48E34FDEBADF8C18D851791EF78EFDCC2A34BDE6810"  
01682 "941D0C0E0146804CC25571261C9483B6887B7769C895B76FDD162A4A99FDEBB9DCD6A04CD7"  
01683 "A6328E7694E3C2080A611665FA2508F64332E5D8AB3357E36753CABF60725FF30C6F6E0E82"  
01684 "E80DF32F6DF9BC11EA9CE6F0851D1B0CBAF0732AE05A307BBB39A56A28B164D6459EC3AEF0"  
01685 "52D517BF6E2E9798FAB52BCE087E42D0D8DC45625A6F7A32A7E67305C8000E5738D2C8B53E"  
01686 "82F10459665AAF2592CDA2D00792532FB24A77802482B4C70B38072E738457A80CE8D7E2BF"  
01687 "D049E4353F929FCE6E6AF3091695080CE93C11C8B3C177C51BB9640519A81EA7A4A1CEDC26"  
01688 "D128575A281DFA3499E5146E0D47CDFA9E888E5168AF35C64501B9AE45DE4A28973098FE8B"  
01689 "6E809D357503CAA5BD373F767B41763F6864F42E8B7B7EE40D88A137AEF2D22B5F95FD973E"  
01690 "EE370FB3CD9338E15547B5A1934D50110CBF3E4B8DC070B323CCFCA86DAA359739AFE"  
01691 "B7C2B0CC98996D202FB51ACDC8379709423BB7683294A2D7AF0DF6B44B8243DA21DC829630"  
01692 "5F89ED52F0EABBCD1DDF19044FC033B5EAA5BCA6FB93EB7ECE4B8934C3A35D0F5380C5B4FE"  
01693 "526CDBA2F473F23A4F32C4C2B4A3F9A47F96FBF0023CB8FF2FA1CCB07717CA449AAB5BD98"  
01694 "9B6CAAA8D5E2046AED4BF81B2600D3F3BD037F096F006389C3BD875B6064CFBF44C2DE09962"  
01695 "D1F30995D32CA0F5474F41BE202AD67AFC1049870FF330C6887E3F765CFA3A85ECF1182C8"  
01696 "67598AA61C6997158A367D94E202709D839960E6311AD687ADAA05A5ADE0088E6B492B0F7F"  
01697 "E053A67CB4DA4F3CEDA88", // PRN 42  
01698 "E40E6EF4455C67D6EF9D6A72858B9E6E08AE3B48EB5F0339A02BACBF979E8266AD92F3D100"  
01699 "2DDE744C69B7C92656B3239925CB7910A50EB4D03DCD28908ED3D7F560AEA47AC4B0EC9D7B"  
01700 "91AD12DC15920032C25CA1247FC7F69B148166B6E61629C3B84687B9045AD65FE45CFBED41"  
01701 "34B68A2CEF9BE972482C0A9442ED0A2D48A55F2DCE46412BE557D793C26C98A7F68CBE30F0"  
01702 "70435F552811CAF34F2C94EA0A46AC4FA41252D1AB745B5647AEC23A70EECFCCF0CB88F9313"  
01703 "8C45140CF23AC030C6571BE79520684B0785CACAD1918ABFAF2FA925D0E194C85BC7D750A1"  
01704 "EC485E51111ECE870361DBE27791A3A2E26C80A7C0B1AAFB0C013C29824E1708D7DEED9E8"  
01705 "4C55BB5DF732F361BB6F3322166854F8ED0910E29FCF51B86549D47CFE2D4B7EB110B306EF"  
01706 "FC076F37FBE276329C70DEB6308D9B959447D8604ABE6F18F653E141142E0C77F08744CD49"  
01707 "C284D2C41410786F892A662821BFA5CB5F0FF00ED7584B576E8D3E0D94C0D1A8B1B532A64"  
01708 "F78BFC8AD8676799E8970349828A780B76A76DE7625E6C76EABE4AF12D7F80E3E43AF3DC2"  
01709 "D7EC10A7AF4EC6C3C158AE63D82FB47EE6D98F9043EA02544825434F4FDAC70C139B73088"  
01710 "026AD9607A3B04FEC00BC9E6B8E30C4A606D470FB0E022200A786133CDB870BADEBBF8812"  
01711 "81C8C02FD8AEBEC229DAC57C9F5B49E7551488968821DA88FB7778F796030CA04E78E53BA"  
01712 "EE835D2D51AC62564E579291F205AE7337774677394A326D1D7C1E1E692B7738ABABD3BBFD"

01722 "B1F0E999098FDF74B3817A19E6BA5CC4A41F6760F2A57CEB0C3F668BA49E0E52065B3C3072"  
01723 "34D5735EAE7FE7EF5ABA539C2178A321ED5A59E7BE313757C51969AF57486FBEF7757C2558"  
01724 "A1599C996C8CB823A580C", // PRN 43  
01725 "8B831E082C933E273C83032428539C331C131145AF171D3CB5C4E46C44E2CA0E185EA893B0"  
01726 "69CABC0DBE995C6A123ACF7D546CF4F0FD15FB0F862B4983845F5CCE43F01EA0B2D6523A8D"  
01727 "01BBEE1956EEDA8102AEFFD2EDAEC51644548744D6B51D76F03EEF031C4B5486CED1C94B06"  
01728 "9F1F73E53B2027D028CCF042FA100CDB6BB8D82A4D9DF60A4E7C70664AD2069E57365B0F28"  
01729 "C793D5B8D3313AD3E582404918F0258FD7D70FE878CE8B4DD4DBD092BC17622C46EDFE8852"  
01730 "B638AD2B60AA4784DCAA7CA2F40069D675AACD131FEB0A2E967769670E66A3825ABC8BA6F"  
01731 "5ACC7EBC505CD2BDBE148F21B0398237076CC593D9BCC457BE7A522E3C5E1B0B6A6EB5A4D4"  
01732 "CF788DA7C640FEAA1B28E555B21FB0DAC6503F70D0137387BAD7AA915C9BC7EA692573FA01"  
01733 "2E4A5FF41215F4244AADC04342917B6AE6BA63B4A46C0B715FAC82CEEDB6228BDA51B4548D"  
01734 "C477EECC3FCE6EECED8B15A458420CE6BA48798D7FAB3295C9AE161D7FA1DED73858A29D69"  
01735 "8C1D4005A75D914659603824C5BA2FB807EB63E60DA4C47D3A43D59FAB931E9BC50FEFFBCC"  
01736 "AE3930778018849C388B405EE902183A69A8FE2645378C0144B05CB45BF88FFCB9FD1D19C0"  
01737 "283EDC6116C66D5A54F6542921D4B423A3CBF2530EF534CE01B735F49D5BCB0634066B0617"  
01738 "EC916E7076CF084B30BA248730645D299C37B8B6B181C67AE492AFAC33241D22336342DEC"  
01739 "1B6F74194CA58C21417AE9D4AEE87FD30E77800DE6697FD1040F9BCCA86C2B838F3973591C6"  
01740 "3F843D875B3A2023EAF38DF399DE1DFB696C8D06EE8D75FEF76956AFB612D5F4CF06DB484E"  
01741 "E4E31D7B3C4B5D371EE92EAD3DC6C1D0067D57B822C9F70E3CA17C065546D678D04828BEB0"  
01742 "76572B582FC6939EC5B0", // PRN 44  
01743 "96FF996AF5BF7DBF540BC4737B5F5C0C6CABAFDD3A821A2DA1F597066FC4DB438D8918310B"  
01744 "3184EAE80F2F383C0B0619A1F1F2F77C7F4DD055363C3795EDB3036F582E813D7F44518E8"  
01745 "E959839ABFFFFF2DE9C657C63B9DEF44D18474BE37AA6FCE42A0D2C8F1BD4DE0A7B674A4B"  
01746 "4AF2E67AD12411B0F6113AC6084B5B1F5260E973CF98C0E96B3FA80A224C0BB3295446D538"  
01747 "4FD94A3BEEAE7CF8AA7A171313550554AB13E1179D6BAF847D0E3D898E66683973871F989A4"  
01748 "0D3353E1D8B3EAECA00345DDC6D5D2987D666624B38B9EB0921253B5CFFC2DC9FA5E226F26"  
01749 "59EC8CD131BF7E4093C81F99EEBFEE27B40319E2D4052502F56E79042EFB042709FD035AD5"  
01750 "8203DE6180757C5B7B0C471AF3612E93EA2442B577039AD52C68E08DD9B377D96ED0A453DC"  
01751 "CB5650392D7210EF24B2BE9E3F3D0CEE96F7EFB18ACC226F022C136FC0FDA126A0E1CF7E6"  
01752 "17C1A8AD9E9110B24A4A188A51C19FB67A514F80A967F847C7A4E4BC1BB0505D9865FEC26F"  
01753 "D91813D10A263421503856AEE4DC1C6CF50544EFF6C685174CF644B729C28A4AAC12B44876"  
01754 "B09C10FC6766313A226AD1CB2881903AC4043DD36E8BC1B302653B0035837AC6D1D349A92"  
01755 "577E4E84ACDB2532BC954CB08ACFFBBD1D28AE65951BC79F78CC755925A565268BE2441B1"  
01756 "D40DA1F450B1EFDD2BCF87F815D8DD1F3B05D8F98A838944B60DE76DCE52C36FE5DDB4F837"  
01757 "149853F7831A536A95EC0BAA9A94C96F86B342DA911FEFBA959DE0E58999DC13C16EB9792"  
01758 "B5B92288FF16148F910A5B06D8D05457B12E15619C025B13E4D93472E66F80AB49A23A2201"  
01759 "FB2817A61273291500D84147BAFC8A3B96BEF241A7B0064F13AD2E0C84C82BC1F3CFCEB8A9"  
01760 "3B927D1B58939AA6EC018", // PRN 45  
01761 "87FC908811EE777B2C509E835CDA0260B65E061E9E508F09B49DBF046446AA33C336FDF490"  
01762 "573B177B6C8C791B3B1CC26A874FD9BF1AC54E0D3841A04AE293B5D9E079EBD219082C402F"  
01763 "56ABDBA94930E91F99273811E819E31EBE31E466AAF25F002BA7A3E09C7FC4A5188FE5616"  
01764 "BC1E1AE54B663B502AD59B67579072DC93F49CBA8351B9E563BD67F56179D8370579A8FC1"  
01765 "4133C4E9437434A5E1D0E53E398EADD2C0BF0EF6542EF6101944F25B96EADD0B3853D1D27A"  
01766 "C64108701E3DCBB0BF7F45B9C173CD4207E267090CF7870427BE248713E06411959D2D27D"  
01767 "B5510376AAE140C5FB79D4AC873E55A3863A0FD183E4FC2D1B5D6A5B942F74A79BB64981F7"  
01768 "189FCDAD06793891B7B97BBA3ABB8BC88976D8902B56CA9D548182C65F33D3C121A80817C"  
01769 "D1DF6B4EBB072F4D0D03443A7EFDAA50812ED4E4AC241EB86596146D3228FE464D849C51CF9"  
01770 "CEDCB851AFFB2C92DB16CEF971B6A352BD1589289C0BD619645A0140879923A9588AFA38DC"  
01771 "097DB0C006D82C0D4E1B6B273E184EC81BD688E832B61D39AADD13C2E27E0D57CA72C1FABA"  
01772 "81CA38936473BC3AC79C6635D265172A26F794A2AA8175DCB91792871E14258717AFEF66E"  
01773 "DCC1AF2E2D3A232DEE44A3CFB24696324E466DCF10575C0F14562DA1AF866ADF26A4D54EAA"  
01774 "888F231327F006716BF1F4F801E09FF7FD4528D2B4FC9247358EA7B9AA2EA2B9E80F5660CF"  
01775 "D198779AD8337949A0BBA1FEF3B998AA45DBAA7D5E6F4BDABD00C86097576E199A9C0E8D9"  
01776 "E91794E3D545AFDA0AB92A62B66429F3668762B10DBF683F77DD2FF99F92B94B3060D8747C"  
01777 "34C581102E21C6596248AA139283377A3ED85DA3C999A6FB029780A9CA1DAED1C9B889EF1A"  
01778 "1441219E3AFED76F1BEA0", // PRN 46  
01779 "ACE6BD607FADA0C25C3F74EF23BAF6E29C70214318035E80B160E5485BD57C691605DE7783"  
01780 "B9EFCF498CE5B4D8EE6619191DA2A15820015DF7F0B8AB05B22562DD437FBE3AD141AE0E6A"  
01781 "E45313FF915A39A49466F11B651CF210143D6FB06BA6E5BF4083671FB3DC9454E4EA38ED98"  
01782 "0C5C2EB65F24FAD970316D5BBF490B2403BFDCA348A7DB1B4AFD9A91B6898381CEEBA5F6E"  
01783 "A1419C19D21E1A64DA2063D824AADF7C0E546D608F7FF19B89BDD9E8808A7A236E98DF0728"  
01784 "BA39498DC6BCFE82C534DD6BF02B303B2DB401DF41C53F4F5321CC28861D22010DF2D1457"  
01785 "F9C594F7AD0E00753CAF0CB6EE6B34DA2364EDDCC2619930C7DCE4C976F50BE69444BB2759"  
01786 "44D17D07061A871CFE0DED63400B514C6457F4119845A19072678017FD6A498CE3D3932CA"  
01787 "C6B63335D87426F1C16FCE0F5861FF0953CA13E9C911A947A2C82231E61012DB7E782BB839"  
01788 "764EE42A94FCC7BCE4BED84D891EE959D9430B963CDD38EC2F63884AA493BE7FEAF47B6252"  
01789 "F2F042531A5FF5347AE9C322C374AEF6A7F4529BCBD085696178D1D2C12C9AB91A2B66E3F0"  
01790 "5A45A1950BBFA797134B4430DC09289B09EB1BD88D05B03D2EC269500532FCEBF7EC7A652E"  
01791 "A9F96AFB622E6D71E568756C2D1700D55238AD36D86A6D92C62C4522F4EB2F2AA0F9FAB8E5"  
01792 "6D4052B13890F0DA9BC17376E248EC86AE21EA8430B110A8A2AA835D7317095CCF2FC00ABD"  
01793 "132A1D2CB7CCB105871E4FCE7F75D8B7CB3E9FE24303DAF57EE80581725B9ADCD9CE09F7CA"  
01794 "50DB4B2B7CF8AFCC7C39958F4A963E92E3E29F4349275C772F68D135AE05CA99019C14166E"  
01795 "D34E0C3598A67C9CD8CA68832D7D7EECA0BE86B4355B8439002D2F7CC7933F07E6A8F0908"  
01796 "CF0F3903996C390142728", // PRN 47  
01797 "9F4DE194FAEB87B4D4934D86F08C2F3506BFC8F0F0AE05E9AB45A17B36AA9C7D5534127B3C"  
01798 "98E60885AC2D00006A8CBA4EAF711CE0C9497B04B539553ED80B24EAC9496BE53725F8EAC"  
01799 "5DB5BDB71664C45D01CD17BE31D039CE8CA16FD9D9D60E9294C8ECFF4975392793C879BB2E"  
01800 "375AC7EFA47B32D6D895E6A3F929E78BE82EFC896E89304FC3F3F0751E22D523D3117AF239"  
01801 "B897B12841429AFDBE8F8D303D50D2A56275CEADAC04CC32A95E9DDB45F4D4C83EA7CA29EF"  
01802 "5E8F7D172D18134E5FE9BC9E442F6D2DD3B01807B994C80697A28CAC0927FE31E62985F629"  
01803 "A965FD14083E72D65835BF6230A660F10D91A04EA9809FF8817129E6503F28596FE9A486D4"  
01804 "86A3B789580E36CF9C6B75BB654874267F50FDA64602EE6986EF10824C90E66585A118F5"  
01805 "C3BD1FF67BBCAD408CF0762D59216884525FCB32863B6A1502302C4F09B3BCB27A9EBA72EA"  
01806 "BE68D86A15CA6FDCE89CD4F20055C322161F1B1F9CFAD8B93A68C42AB19565DABC486B0C"  
01807 "A93215DDC8E128806B4ED06FAA0987A49EFCFC1C23E1BACCB6ACB332D720311B8FED658024"  
01808 "F0CA12EAC658494DA79F9CB56D59F17C927650575562095A99F5A11D13CD1267BE89FBE761"

```
01809 "DF5CF0E9D8C67A9DE21E8BEA108BC3E2E3AA68C7742103B6F975DC232443C34F17D9EAA193"
01810 "9C23514BFF908757849AC3EC6E954FB4991373DF8E7DC6011D16F9C19E4C7021A788C0834D"
01811 "6BC09A560908AF50F71D0496A3DE1EA1DF8F485A3C496B9F4B65A279083F8390966C034829"
01812 "6D05BCFA81D2EF8A2714C4A886D35DF870BA4DCF73BFD8E3EA1DFD4D9E37A0ED8E9032E350"
01813 "947D61B0CA2FE10FC20B8A5AD0E3FD70AE3C8BB91F2E04619ADB3C684E10B9127E5B76B8CD"
01814 "0E9BAD2FD100131E6FCB8", // PRN 48
01815 "953E31E5F9C45E57DB6C54DBEBE650C07F8C14D832BC39D5D860912A7B4EE9F726C0D3FF84"
01816 "2DA45E3E9286321AEB6A45FAB260E1FB3AFB7A4DCBFF3D65491BC61AE83207E1CC56858D13"
01817 "046191A5CBAB51C59F3BB37E281847EE624A8E54DA17273262E790246A54C8873BE252E9FB"
01818 "D7A7ECCD476D62124FD1F28AFE207545525B325DC6F73F25D09E3589BC848E0A803EB10FF5"
01819 "0E81732E5F317CD3AADAF6C123807677B273B6B56955065869078BBE27AA04CEB7ECCA493"
01820 "DC2E9499188C51B16309F014624554EB96E3A69E3E067FA946977379B4BD23A9DD57D5E175"
01821 "FAA891F5FCD72431A3C857A582B233403B0D13C48C502F9E5324DB127DA83F019272A29ACD"
01822 "F325E6AB2908ABA54CD56D1D91188B320A1941720BDC2CF00991C932FC5D7025B69F242F7"
01823 "75A06DD6BD8FFB7615246AC4B7495A350C4BBE9F65F7449969EDCE013BF3F7367647316C2A"
01824 "A55F882BB42C7BD7C43618F78A6409A8659919E3FA20DAA9BC384CC6BEA776A438080DA6DF"
01825 "11AB8A853F06DAAE429883F3C153FEDFDF9D968FF806D9B7CA193F3A72D316E0297A01CF1"
01826 "AB6350D2E807F93EC27E7BD91CEB6CDC44001E092EADF5523BF0D67EFD8D9E5A2F7ECA"
01827 "0B126F4D2AA32A484B5272AD7CE9E9F26CF418E6642927124483043E3C15B7833655C6388A"
01828 "26A10152857221718FB9AFE36331E071C04E498DA3181B5F5BC701CC5C30BB963183F6AC5F"
01829 "26A5CACB32ECB7C8A88FDFCE73823FA6FBD965B2772067CC2B81BE5A35660ACD3F5658ED28"
01830 "A0D225B54392F0D03983148ED738FF3A80E5DEB5D2DF3092C1AAF81272E75190397F9FA66A"
01831 "D3C101F259496F9EA011082D7BE7E1E10F80ECC4C42AF40E0AB498081404E5593A71147038"
01832 "3D8149ACE622AA769A900", // PRN 49
01833 "CE11D9F5C85027678446809BB43074F041568F30717E5B784BD0B64FAAE0610A144496CF4A"
01834 "AC1BEE53CF98AAB2BA75239D228BF946D99CD0899C250CF1850D17ADD418E389F4ACE5A942"
01835 "6A1251074E28C13A818477875BCOFF732517A6CFB18A23F633E9D0F876582907AC8C3403A5"
01836 "7BBAC9706D7BFAABEDF132A3990E6C4089BAD9939F1F8CE910779BE5CF96EA87716BDC45CB"
01837 "5C7BCB5D14123D03BE2E50909C523D10096335770D849D509811EC426D80E1115B3AF367D9"
01838 "41C74D8025D7C7F582AEC5E5E7AD5F38538CFE33387B5D19AB846A7CFA6C6CC4E0AFC054DF"
01839 "EDCDBD0A8B489A082B3B2A064EBE0902AF6BDA5B64310641F2D717D06E1064D2C2C0ABB19C"
01840 "66DFBA807E36ACE0F6BB487C9868C2F354909960BDF7A8B4C42CC935EDFB10C1975F051D40"
01841 "29A76D255A1CBAD199393A8A7BBCF80C25649C22B475244323D5323675D74886B30A90B9B9"
01842 "1939521DE115DD5AD0509363C6057B9DB321C0871D45B9269B048A98148BD91E3976DDB740"
01843 "5F0AA990A8C7987767F76AD1CBA8A8442DF151E28FFE3D4A985BFF6C263108D8873CE28F62"
01844 "DED610ECB1A6C35FE5D66F437B227F184D856EF994099763510425B16234A31B14FC8BD6B"
01845 "C37264F9BAEBF838AC2D03BEA98EA7F9FA0160A21BC211EBFB82A5CC2A70F72FFBED8170AE"
01846 "C8209A6B7ED8CFA7A6FFB9E956239CA50249D14E365EA7166033BC83D1FB6F336A8BC85E0"
01847 "EB6EE26EFD4B8A5028CA26F9CE872C7A3CFD21CACF29E42DE0309211CCC8143C7CB336A450"
01848 "A2B786D526F53265B60339ABC9ECA67FAAFBD3CE1DE3C3867A9D1D758AE3162BC3D464EDD"
01849 "1C77EA1A4B4A3A0DBD80D05E566F97DF2555E427C8466AE006876402B89FBF37CA38ADDBC2"
01850 "083D30D7F2DAEADFE334" // PRN 50
01851 };
01852
01853
01854 constexpr size_t GALILEO_E6_C_SECONDARY_CODE_STR_LENGTH = 25;
01855 constexpr char GALILEO_E6_C_SECONDARY_CODE[GALILEO_E6_NUMBER_OF_CODES][26] = {
01856 "83F6F69D8F6E15411FB8C9B1C", // PRN 01
01857 "66558BD3CE0C7792E83350525", // PRN 02
01858 "59A025A9C1AF0651B779A8381", // PRN 03
01859 "D3A32640782F7B18E4DF754B7", // PRN 04
01860 "B91FCAD7760C218FA59348A93", // PRN 05
01861 "BAC77E933A779140F094FBF98", // PRN 06
01862 "537785DE280927C6B58BA6776", // PRN 07
01863 "EFCAB4B65F38531ECA22257E2", // PRN 08
01864 "79F8CAE838475EA5584BEFC9B", // PRN 09
01865 "CA5170FEA3A810EC606B66494", // PRN 10
01866 "1FC32410652A2C49BD845E567", // PRN 11
01867 "FE0A9A7AFDAC44E42CB95D261", // PRN 12
01868 "B03062DC2B71995D5AD8B7DBE", // PRN 13
01869 "F6C398993F598E2DF4235D3D5", // PRN 14
01870 "1BB2FB8B5BF24395C2EF3C5A1", // PRN 15
01871 "2F920687D238CC7046EF6AFC9", // PRN 16
01872 "34163886FC4ED7F2A92EFDDB8", // PRN 17
01873 "66A872CE47833FB2DFD5625AD", // PRN 18
01874 "99D5A70162C920A4BB9DE1CA8", // PRN 19
01875 "81D71BD6E069A7ACCBEDC66CA", // PRN 20
01876 "A654524074A9E6780DB9D3EC6", // PRN 21
01877 "C3396A101BEDAF623CFC5BB37", // PRN 22
01878 "C3D4AB211DF36F2111F2141CD", // PRN 23
01879 "3DFF25EAE761739265AF145C1", // PRN 24
01880 "994909E075D70CDE389102B5", // PRN 25
01881 "B938535522D119F40C25FDAEC", // PRN 26
01882 "C71AB549C0491537026B390B7", // PRN 27
01883 "0CDB8C9E7B53F55F5B0A0597B", // PRN 28
01884 "61C5FA252F1AF81144766494F", // PRN 29
01885 "626027778FD3C6BB4BAA7A59D", // PRN 30
01886 "E745412FF53DEBD03F1C9A633", // PRN 31
01887 "3592AC083F3175FA724639098", // PRN 32
01888 "52284D941C3DCAF2721DDB1FD", // PRN 33
01889 "73B3D8F0AD55DF4FE814ED890", // PRN 34
01890 "94BF16C83BD7462F6498E0282", // PRN 35
01891 "A8C3DE1AC668089B0B45B3579", // PRN 36
01892 "E23FFC2DD2C14388AD8D6BEC8", // PRN 37
01893 "F2AC871CDF89DDC06B5960D2B", // PRN 38
01894 "06191EC1F622A77A526868BA1", // PRN 39
01895 "22D6E2A768E5F35FFC8E01796", // PRN 40
```

```

01896     "25310A06675EB271F2A09EA1D", // PRN 41
01897     "9F7993C621D4BEC81A0535703", // PRN 42
01898     "D62999EACF1C99083C0B4A417", // PRN 43
01899     "F665A7EA441BAA4EA0D01078C", // PRN 44
01900     "46F3D3043F24CDEABD6F79543", // PRN 45
01901     "E2E3E8254616BD96CEFC6A651A", // PRN 46
01902     "E548231A82F9A01A19DB5E1B2", // PRN 47
01903     "265C7F90A16F49EDE2AA706C8", // PRN 48
01904     "364A3A9EB0F0481DA0199D7EA", // PRN 49
01905     "9810A7A898961263A0F749F56" // PRN 50
01906 };
01907
01908
01909 /** \} */
01910 /** \} */
01911 #endif // GNSS_SDR_GALILEO_E6_H

```

## 11.707 galileo\_ephemeris.h File Reference

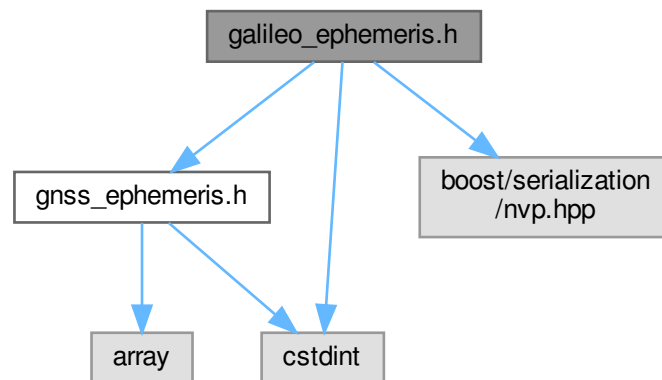
Interface of a Galileo EPHEMERIS storage.

```

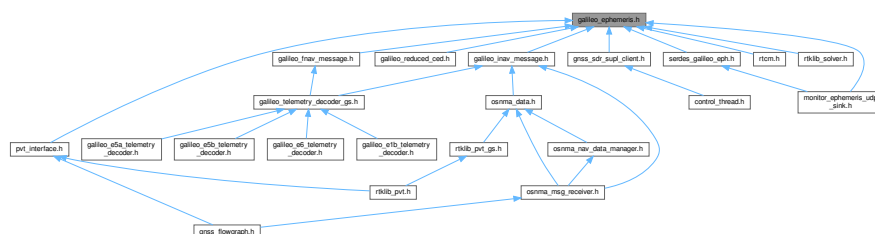
#include "gnss_ephemeris.h"
#include <boost/serialization/nvp.hpp>
#include <cstdint>

```

Include dependency graph for galileo\_ephemeris.h:



This graph shows which files directly or indirectly include this file:



### Classes

- class [Galileo\\_Ephemeris](#)

*This class is a storage and orbital model functions for the Galileo SV ephemeris data as described in Galileo ICD paragraph 5.1.1.*



## 11.707.1 Detailed Description

Interface of a Galileo EPHEMERIS storage.

### Author

Javier Arribas, 2013. jarribas(at)cttc.es,

Mara Branzanti 2013. mara.branzanti(at)gmail.com

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [galileo\\_ephemeris.h](#).

---

## 11.708 galileo\_ephemeris.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file galileo_ephemeris.h
00003  * \brief Interface of a Galileo EPHEMERIS storage
00004  * \author Javier Arribas, 2013. jarribas(at)cttc.es,
00005  * \author Mara Branzanti 2013. mara.branzanti(at)gmail.com
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018
00019 #ifndef GNSS_SDR_GALILEO_EPHEMERIS_H
00020 #define GNSS_SDR_GALILEO_EPHEMERIS_H
00021
00022 #include "gnss_ephemeris.h"
00023 #include <boost/serialization/nvp.hpp>
00024 #include <cstdint>
00025
00026 /** \addtogroup Core
00027  * \{ */
00028 /** \addtogroup System_Parameters
00029  * \{ */
00030
00031
00032  /*!
00033  * \brief This class is a storage and orbital model functions for the Galileo SV
00034  * ephemeris data as described in Galileo ICD paragraph 5.1.1
00035  *
00036  * (See https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.0.pdf )
00037  *
00038  */
00039  class Galileo_Ephemeris : public Gnss_Ephemeris
00040  {
00041  public:
00042      Galileo_Ephemeris()
00043      {
00044          this->System = 'E';
00045      }
00046
00047      double Galileo_System_Time(double week_number, double TOW); //!< Galileo System Time (GST), ICD
        paragraph 5.1.2
00048
00049      int32_t IOD_ephemeris{};
00050      int32_t IOD_nav{};
00051
00052      // SV status
00053      int32_t SISA{}; //!< Signal in space accuracy index
00054      int32_t E5a_HS{}; //!< E5a Signal Health Status
00055      int32_t E5b_HS{}; //!< E5b Signal Health Status
00056      int32_t E1B_HS{}; //!< E1B Signal Health Status
00057      bool E5a_DVS{}; //!< E5a Data Validity Status
00058      bool E5b_DVS{}; //!< E5b Data Validity Status
00059      bool E1B_DVS{}; //!< E1B Data Validity Status
00060      double BGD_E1E5a{}; //!< E1-E5a Broadcast Group Delay [s]
00061      double BGD_E1E5b{}; //!< E1-E5b Broadcast Group Delay [s]
00062
00063      bool flag_all_ephemeris{};
00064

```

```

00065     template <class Archive>
00066
00067     /*!
00068      * \brief Serialize is a boost standard method to be called by the boost XML
00069      * serialization. Here is used to save the ephemeris data on disk file.
00070      */
00071     inline void serialize(Archive& archive, const uint32_t version)
00072     {
00073         if (version)
00074         {
00075             };
00076
00077         archive& BOOST_SERIALIZATION_NVP (PRN);
00078         archive& BOOST_SERIALIZATION_NVP (M_0);
00079         archive& BOOST_SERIALIZATION_NVP (delta_n);
00080         archive& BOOST_SERIALIZATION_NVP (ecc);
00081         archive& BOOST_SERIALIZATION_NVP (sqrtA);
00082         archive& BOOST_SERIALIZATION_NVP (OMEGA_0);
00083         archive& BOOST_SERIALIZATION_NVP (i_0);
00084         archive& BOOST_SERIALIZATION_NVP (omega);
00085         archive& BOOST_SERIALIZATION_NVP (OMEGAdot);
00086         archive& BOOST_SERIALIZATION_NVP (idot);
00087         archive& BOOST_SERIALIZATION_NVP (Cuc);
00088         archive& BOOST_SERIALIZATION_NVP (Cus);
00089         archive& BOOST_SERIALIZATION_NVP (Crc);
00090         archive& BOOST_SERIALIZATION_NVP (Crs);
00091         archive& BOOST_SERIALIZATION_NVP (Cic);
00092         archive& BOOST_SERIALIZATION_NVP (Cis);
00093         archive& BOOST_SERIALIZATION_NVP (toe);
00094         archive& BOOST_SERIALIZATION_NVP (toc);
00095         archive& BOOST_SERIALIZATION_NVP (af0);
00096         archive& BOOST_SERIALIZATION_NVP (af1);
00097         archive& BOOST_SERIALIZATION_NVP (af2);
00098         archive& BOOST_SERIALIZATION_NVP (WN);
00099         archive& BOOST_SERIALIZATION_NVP (tow);
00100         archive& BOOST_SERIALIZATION_NVP (satClkDrift);
00101         archive& BOOST_SERIALIZATION_NVP (dtr);
00102
00103         archive& BOOST_SERIALIZATION_NVP (IOD_ephemeris);
00104         archive& BOOST_SERIALIZATION_NVP (IOD_nav);
00105         archive& BOOST_SERIALIZATION_NVP (SISA);
00106         archive& BOOST_SERIALIZATION_NVP (E5a_HS);
00107         archive& BOOST_SERIALIZATION_NVP (E5b_HS);
00108         archive& BOOST_SERIALIZATION_NVP (E1B_HS);
00109         archive& BOOST_SERIALIZATION_NVP (E5a_DVS);
00110         archive& BOOST_SERIALIZATION_NVP (E5b_DVS);
00111         archive& BOOST_SERIALIZATION_NVP (E1B_DVS);
00112         archive& BOOST_SERIALIZATION_NVP (BGD_E1E5a);
00113         archive& BOOST_SERIALIZATION_NVP (BGD_E1E5b);
00114         archive& BOOST_SERIALIZATION_NVP (flag_all_ephemeris);
00115     }
00116 };
00117
00118
00119 /** \} */
00120 /** \} */
00121 #endif // GNSS_SDR_GALILEO_EPHEMERIS_H

```

## 11.709 Galileo\_FNAV.h File Reference

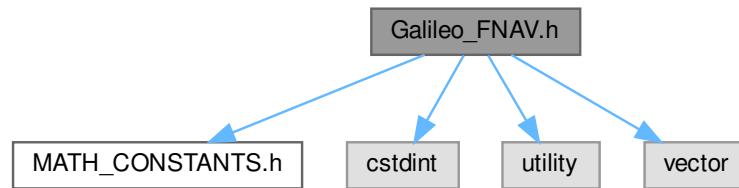
Galileo FNAV message constants.

```

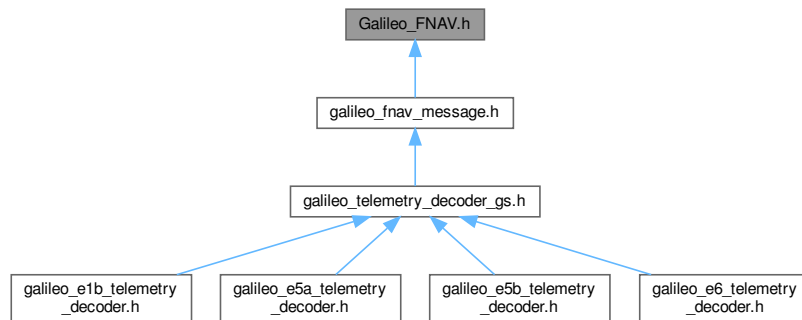
#include "MATH_CONSTANTS.h"
#include <cstdint>
#include <utility>
#include <vector>

```

Include dependency graph for Galileo\_FNAV.h:



This graph shows which files directly or indirectly include this file:



## Functions

- `const std::vector< std::pair< int32_t, int32_t > > > FNAV_PAGE_TYPE_BIT {{{1, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > > FNAV_SV_ID_PRN_1_BIT {{{7, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > > FNAV_IO_DNAV_1_BIT {{{13, 10}}}`
- `const std::vector< std::pair< int32_t, int32_t > > > FNAV_T0C_1_BIT {{{23, 14}}}`
- `const std::vector< std::pair< int32_t, int32_t > > > FNAV_AF0_1_BIT {{{37, 31}}}`
- `const std::vector< std::pair< int32_t, int32_t > > > FNAV_AF1_1_BIT {{{68, 21}}}`
- `const std::vector< std::pair< int32_t, int32_t > > > FNAV_AF2_1_BIT {{{89, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > > FNAV_SISA_1_BIT {{{95, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > > FNAV_AIO_1_BIT {{{103, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > > FNAV_AI1_1_BIT {{{114, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > > FNAV_AI2_1_BIT {{{125, 14}}}`
- `const std::vector< std::pair< int32_t, int32_t > > > FNAV_REGION1_1_BIT {{{139, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > > FNAV_REGION2_1_BIT {{{140, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > > FNAV_REGION3_1_BIT {{{141, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > > FNAV_REGION4_1_BIT {{{142, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > > FNAV_REGION5_1_BIT {{{143, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > > FNAV_BGD_1_BIT {{{144, 10}}}`
- `const std::vector< std::pair< int32_t, int32_t > > > FNAV_E5AHS_1_BIT {{{154, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > > FNAV_WN_1_BIT {{{156, 12}}}`
- `const std::vector< std::pair< int32_t, int32_t > > > FNAV_TOW_1_BIT {{{168, 20}}}`
- `const std::vector< std::pair< int32_t, int32_t > > > FNAV_E5ADVS_1_BIT {{{188, 1}}}`



- `const std::vector< std::pair< int32_t, int32_t > > FNAV_IO_DNAV_2_BIT ({7, 10})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_M0_2_BIT ({17, 32})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_OMEGADOT_2_BIT ({49, 24})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_E_2_BIT ({73, 32})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_A12_2_BIT ({105, 32})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_OMEGA0_2_BIT ({137, 32})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_IDOT_2_BIT ({169, 14})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_WN_2_BIT ({183, 12})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_TOW_2_BIT ({195, 20})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_IO_DNAV_3_BIT ({7, 10})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_I0_3_BIT ({17, 32})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_W_3_BIT ({49, 32})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_DELTAN_3_BIT ({81, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_CUC_3_BIT ({97, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_CUS_3_BIT ({113, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_CRC_3_BIT ({129, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_CRS_3_BIT ({145, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_T0E_3_BIT ({161, 14})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_WN_3_BIT ({175, 12})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_TOW_3_BIT ({187, 20})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_IO_DNAV_4_BIT ({7, 10})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_CIC_4_BIT ({17, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_CIS_4_BIT ({33, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_A0_4_BIT ({49, 32})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_A1_4_BIT ({81, 24})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_DELTATLS_4_BIT ({105, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_T0T_4_BIT ({113, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_W_NOT_4_BIT ({121, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_W_NLSF_4_BIT ({129, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_DN_4_BIT ({137, 3})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_DELTATLSF_4_BIT ({140, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_T0G_4_BIT ({148, 8})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_A0G_4_BIT ({156, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_A1G_4_BIT ({172, 12})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_W_N0G_4_BIT ({184, 6})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_TOW_4_BIT ({190, 20})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_IO_DA_5_BIT ({7, 4})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_W_NA_5_BIT ({11, 2})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_T0A_5_BIT ({13, 10})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_SVI_D1_5_BIT ({23, 6})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_DELTA12_1_5_BIT ({29, 13})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_E_1_5_BIT ({42, 11})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_W_1_5_BIT ({53, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_DELTAI_1_5_BIT ({69, 11})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_OMEGA0_1_5_BIT ({80, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_OMEGADOT_1_5_BIT ({96, 11})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_M0_1_5_BIT ({107, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_AF0_1_5_BIT ({123, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_AF1_1_5_BIT ({139, 13})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_E5AHS_1_5_BIT ({152, 2})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_SVI_D2_5_BIT ({154, 6})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_DELTA12_2_5_BIT ({160, 13})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_E_2_5_BIT ({173, 11})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_W_2_5_BIT ({184, 16})`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_DELTAI_2_5_BIT ({200, 11})`

- `const std::vector< std::pair< int32_t, int32_t > > FNAV_IO_DA_6_BIT {{{7, 4}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_OMEGADOT_2_6_BIT {{{23, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_M0_2_6_BIT {{{34, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_AF0_2_6_BIT {{{50, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_AF1_2_6_BIT {{{66, 13}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_E5AHS_2_6_BIT {{{79, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_SVI_D3_6_BIT {{{81, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_DELTAA12_3_6_BIT {{{87, 13}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_E_3_6_BIT {{{100, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_W_3_6_BIT {{{111, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_DELTAI_3_6_BIT {{{127, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_OMEGA0_3_6_BIT {{{138, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_OMEGADOT_3_6_BIT {{{154, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_M0_3_6_BIT {{{165, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_AF0_3_6_BIT {{{181, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_AF1_3_6_BIT {{{197, 13}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FNAV_E5AHS_3_6_BIT {{{210, 2}}}`

## Variables

- `constexpr int32_t FNAV_T0C_1_LSB = 60`
- `constexpr double FNAV_AF0_1_LSB = TWO_N34`
- `constexpr double FNAV_AF1_1_LSB = TWO_N46`
- `constexpr double FNAV_AF2_1_LSB = TWO_N59`
- `constexpr double FNAV_AI0_1_LSB = TWO_N2`
- `constexpr double FNAV_AI1_1_LSB = TWO_N8`
- `constexpr double FNAV_AI2_1_LSB = TWO_N15`
- `constexpr double FNAV_BGD_1_LSB = TWO_N32`
- `constexpr double FNAV_M0_2_LSB = PI_TWO_N31`
- `constexpr double FNAV_OMEGADOT_2_LSB = PI_TWO_N43`
- `constexpr double FNAV_E_2_LSB = TWO_N33`
- `constexpr double FNAV_A12_2_LSB = TWO_N19`
- `constexpr double FNAV_OMEGA0_2_LSB = PI_TWO_N31`
- `constexpr double FNAV_IDOT_2_LSB = PI_TWO_N43`
- `constexpr double FNAV_I0_3_LSB = PI_TWO_N31`
- `constexpr double FNAV_W_3_LSB = PI_TWO_N31`
- `constexpr double FNAV_DELTAN_3_LSB = PI_TWO_N43`
- `constexpr double FNAV_CUC_3_LSB = TWO_N29`
- `constexpr double FNAV_CUS_3_LSB = TWO_N29`
- `constexpr double FNAV_CRC_3_LSB = TWO_N5`
- `constexpr double FNAV_CRS_3_LSB = TWO_N5`
- `constexpr int32_t FNAV_T0E_3_LSB = 60`
- `constexpr double FNAV_CIC_4_LSB = TWO_N29`
- `constexpr double FNAV_CIS_4_LSB = TWO_N29`
- `constexpr double FNAV_A0_4_LSB = TWO_N30`
- `constexpr double FNAV_A1_4_LSB = TWO_N50`
- `constexpr int32_t FNAV_T0T_4_LSB = 3600`
- `constexpr int32_t FNAV_T0G_4_LSB = 3600`
- `constexpr double FNAV_A0G_4_LSB = TWO_N35`
- `constexpr double FNAV_A1G_4_LSB = TWO_N51`
- `constexpr int32_t FNAV_T0A_5_LSB = 600`
- `constexpr double FNAV_DELTAA12_5_LSB = TWO_N9`
- `constexpr double FNAV_E_5_LSB = TWO_N16`
- `constexpr double FNAV_W_5_LSB = TWO_N15`
- `constexpr double FNAV_DELTAI_5_LSB = TWO_N14`

- constexpr double [FNAV\\_OMEGA0\\_5\\_LSB](#) = [TWO\\_N15](#)
- constexpr double [FNAV\\_OMEGADOT\\_5\\_LSB](#) = [TWO\\_N33](#)
- constexpr double [FNAV\\_M0\\_5\\_LSB](#) = [TWO\\_N15](#)
- constexpr double [FNAV\\_AF0\\_5\\_LSB](#) = [TWO\\_N19](#)
- constexpr double [FNAV\\_AF1\\_5\\_LSB](#) = [TWO\\_N38](#)

### 11.709.1 Detailed Description

Galileo FNAV message constants.

Author

Carles Fernandez, 2020. cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [Galileo\\_FNAV.h](#).

---

## 11.710 Galileo\_FNAV.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file Galileo_FNAV.h
00003  * \brief Galileo FNAV message constants
00004  * \author Carles Fernandez, 2020. cfernandez(at)cttc.es
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_GALILEO_FNAV_H
00019 #define GNSS_SDR_GALILEO_FNAV_H
00020
00021 #include "MATH_CONSTANTS.h"
00022 #include <stdint>
00023 #include <utility>
00024 #include <vector>
00025
00026 /** \addtogroup Core
00027  * \{ */
00028 /** \addtogroup System_Parameters
00029  * \{ */
00030
00031
00032 const std::vector<std::pair<int32_t, int32_t>> FNAV_PAGE_TYPE_BIT({{1, 6}});
00033
00034 /* WORD 1 iono corrections. FNAV (Galileo E5a message)*/
00035 const std::vector<std::pair<int32_t, int32_t>> FNAV_SV_ID_PRN_1_BIT({{7, 6}});
00036 const std::vector<std::pair<int32_t, int32_t>> FNAV_IO_DNAV_1_BIT({{13, 10}});
00037 const std::vector<std::pair<int32_t, int32_t>> FNAV_T0C_1_BIT({{23, 14}});
00038 constexpr int32_t FNAV_T0C_1_LSB = 60;
00039 const std::vector<std::pair<int32_t, int32_t>> FNAV_AF0_1_BIT({{37, 31}});
00040 constexpr double FNAV_AF0_1_LSB = TWO\_N34;
00041 const std::vector<std::pair<int32_t, int32_t>> FNAV_AF1_1_BIT({{68, 21}});
00042 constexpr double FNAV_AF1_1_LSB = TWO\_N46;
00043 const std::vector<std::pair<int32_t, int32_t>> FNAV_AF2_1_BIT({{89, 6}});
00044 constexpr double FNAV_AF2_1_LSB = TWO\_N59;
00045 const std::vector<std::pair<int32_t, int32_t>> FNAV_SISA_1_BIT({{95, 8}});
00046 const std::vector<std::pair<int32_t, int32_t>> FNAV_AI0_1_BIT({{103, 11}});
00047 constexpr double FNAV_AI0_1_LSB = TWO\_N2;
00048 const std::vector<std::pair<int32_t, int32_t>> FNAV_AI1_1_BIT({{114, 11}});
00049 constexpr double FNAV_AI1_1_LSB = TWO\_N8;
00050 const std::vector<std::pair<int32_t, int32_t>> FNAV_AI2_1_BIT({{125, 14}});
00051 constexpr double FNAV_AI2_1_LSB = TWO\_N15;
00052 const std::vector<std::pair<int32_t, int32_t>> FNAV_REGION1_1_BIT({{139, 1}});
00053 const std::vector<std::pair<int32_t, int32_t>> FNAV_REGION2_1_BIT({{140, 1}});
00054 const std::vector<std::pair<int32_t, int32_t>> FNAV_REGION3_1_BIT({{141, 1}});
00055 const std::vector<std::pair<int32_t, int32_t>> FNAV_REGION4_1_BIT({{142, 1}});
00056 const std::vector<std::pair<int32_t, int32_t>> FNAV_REGION5_1_BIT({{143, 1}});
00057 const std::vector<std::pair<int32_t, int32_t>> FNAV_BGD_1_BIT({{144, 10}});
```

```

00058 constexpr double FNAV_BGD_1_LSB = TWO_N32;
00059 const std::vector<std::pair<int32_t, int32_t>> FNAV_E5AHS_1_BIT({{154, 2}});
00060 const std::vector<std::pair<int32_t, int32_t>> FNAV_WN_1_BIT({{156, 12}});
00061 const std::vector<std::pair<int32_t, int32_t>> FNAV_TOW_1_BIT({{168, 20}});
00062 const std::vector<std::pair<int32_t, int32_t>> FNAV_E5ADVS_1_BIT({{188, 1}});
00063
00064 // WORD 2 Ephemeris (1/3)
00065 const std::vector<std::pair<int32_t, int32_t>> FNAV_IO_DNAV_2_BIT({{7, 10}});
00066 const std::vector<std::pair<int32_t, int32_t>> FNAV_M0_2_BIT({{17, 32}});
00067 constexpr double FNAV_M0_2_LSB = PI_TWO_N31;
00068 const std::vector<std::pair<int32_t, int32_t>> FNAV_OMEGADOT_2_BIT({{49, 24}});
00069 constexpr double FNAV_OMEGADOT_2_LSB = PI_TWO_N43;
00070 const std::vector<std::pair<int32_t, int32_t>> FNAV_E_2_BIT({{73, 32}});
00071 constexpr double FNAV_E_2_LSB = TWO_N33;
00072 const std::vector<std::pair<int32_t, int32_t>> FNAV_A12_2_BIT({{105, 32}});
00073 constexpr double FNAV_A12_2_LSB = TWO_N19;
00074 const std::vector<std::pair<int32_t, int32_t>> FNAV_OMEGA0_2_BIT({{137, 32}});
00075 constexpr double FNAV_OMEGA0_2_LSB = PI_TWO_N31;
00076 const std::vector<std::pair<int32_t, int32_t>> FNAV_IDOT_2_BIT({{169, 14}});
00077 constexpr double FNAV_IDOT_2_LSB = PI_TWO_N43;
00078 const std::vector<std::pair<int32_t, int32_t>> FNAV_WN_2_BIT({{183, 12}});
00079 const std::vector<std::pair<int32_t, int32_t>> FNAV_TOW_2_BIT({{195, 20}});
00080
00081 // WORD 3 Ephemeris (2/3)
00082 const std::vector<std::pair<int32_t, int32_t>> FNAV_IO_DNAV_3_BIT({{7, 10}});
00083 const std::vector<std::pair<int32_t, int32_t>> FNAV_IO_3_BIT({{17, 32}});
00084 constexpr double FNAV_IO_3_LSB = PI_TWO_N31;
00085 const std::vector<std::pair<int32_t, int32_t>> FNAV_W_3_BIT({{49, 32}});
00086 constexpr double FNAV_W_3_LSB = PI_TWO_N31;
00087 const std::vector<std::pair<int32_t, int32_t>> FNAV_DELTAN_3_BIT({{81, 16}});
00088 constexpr double FNAV_DELTAN_3_LSB = PI_TWO_N43;
00089 const std::vector<std::pair<int32_t, int32_t>> FNAV_CUC_3_BIT({{97, 16}});
00090 constexpr double FNAV_CUC_3_LSB = TWO_N29;
00091 const std::vector<std::pair<int32_t, int32_t>> FNAV_CUS_3_BIT({{113, 16}});
00092 constexpr double FNAV_CUS_3_LSB = TWO_N29;
00093 const std::vector<std::pair<int32_t, int32_t>> FNAV_CRC_3_BIT({{129, 16}});
00094 constexpr double FNAV_CRC_3_LSB = TWO_N5;
00095 const std::vector<std::pair<int32_t, int32_t>> FNAV_CRS_3_BIT({{145, 16}});
00096 constexpr double FNAV_CRS_3_LSB = TWO_N5;
00097 const std::vector<std::pair<int32_t, int32_t>> FNAV_T0E_3_BIT({{161, 14}});
00098 constexpr int32_t FNAV_T0E_3_LSB = 60;
00099 const std::vector<std::pair<int32_t, int32_t>> FNAV_WN_3_BIT({{175, 12}});
00100 const std::vector<std::pair<int32_t, int32_t>> FNAV_TOW_3_BIT({{187, 20}});
00101
00102 // WORD 4 Ephemeris (3/3)
00103 const std::vector<std::pair<int32_t, int32_t>> FNAV_IO_DNAV_4_BIT({{7, 10}});
00104 const std::vector<std::pair<int32_t, int32_t>> FNAV_CIC_4_BIT({{17, 16}});
00105 constexpr double FNAV_CIC_4_LSB = TWO_N29;
00106 const std::vector<std::pair<int32_t, int32_t>> FNAV_CIS_4_BIT({{33, 16}});
00107 constexpr double FNAV_CIS_4_LSB = TWO_N29;
00108 const std::vector<std::pair<int32_t, int32_t>> FNAV_A0_4_BIT({{49, 32}});
00109 constexpr double FNAV_A0_4_LSB = TWO_N30;
00110 const std::vector<std::pair<int32_t, int32_t>> FNAV_A1_4_BIT({{81, 24}});
00111 constexpr double FNAV_A1_4_LSB = TWO_N50;
00112 const std::vector<std::pair<int32_t, int32_t>> FNAV_DELTATLS_4_BIT({{105, 8}});
00113 const std::vector<std::pair<int32_t, int32_t>> FNAV_TOT_4_BIT({{113, 8}});
00114 constexpr int32_t FNAV_TOT_4_LSB = 3600;
00115 const std::vector<std::pair<int32_t, int32_t>> FNAV_W_NOT_4_BIT({{121, 8}});
00116 const std::vector<std::pair<int32_t, int32_t>> FNAV_W_NLSF_4_BIT({{129, 8}});
00117 const std::vector<std::pair<int32_t, int32_t>> FNAV_DN_4_BIT({{137, 3}});
00118 const std::vector<std::pair<int32_t, int32_t>> FNAV_DELTATLSF_4_BIT({{140, 8}});
00119 const std::vector<std::pair<int32_t, int32_t>> FNAV_TOG_4_BIT({{148, 8}});
00120 constexpr int32_t FNAV_TOG_4_LSB = 3600;
00121 const std::vector<std::pair<int32_t, int32_t>> FNAV_A0G_4_BIT({{156, 16}});
00122 constexpr double FNAV_A0G_4_LSB = TWO_N35;
00123 const std::vector<std::pair<int32_t, int32_t>> FNAV_A1G_4_BIT({{172, 12}});
00124 constexpr double FNAV_A1G_4_LSB = TWO_N51;
00125 const std::vector<std::pair<int32_t, int32_t>> FNAV_W_N0G_4_BIT({{184, 6}});
00126 const std::vector<std::pair<int32_t, int32_t>> FNAV_TOW_4_BIT({{190, 20}});
00127
00128 // WORD 5 Almanac SVID1 SVID2(1/2)
00129 const std::vector<std::pair<int32_t, int32_t>> FNAV_IO_DA_5_BIT({{7, 4}});
00130 const std::vector<std::pair<int32_t, int32_t>> FNAV_W_NA_5_BIT({{11, 2}});
00131 const std::vector<std::pair<int32_t, int32_t>> FNAV_TOA_5_BIT({{13, 10}});
00132 constexpr int32_t FNAV_TOA_5_LSB = 600;
00133 const std::vector<std::pair<int32_t, int32_t>> FNAV_SVI_D1_5_BIT({{23, 6}});
00134 const std::vector<std::pair<int32_t, int32_t>> FNAV_DELTAA12_5_BIT({{29, 13}});
00135 constexpr double FNAV_DELTAA12_5_LSB = TWO_N9;
00136 const std::vector<std::pair<int32_t, int32_t>> FNAV_E_1_5_BIT({{42, 11}});
00137 constexpr double FNAV_E_5_LSB = TWO_N16;
00138 const std::vector<std::pair<int32_t, int32_t>> FNAV_W_1_5_BIT({{53, 16}});
00139 constexpr double FNAV_W_5_LSB = TWO_N15;
00140 const std::vector<std::pair<int32_t, int32_t>> FNAV_DELTAI_1_5_BIT({{69, 11}});
00141 constexpr double FNAV_DELTAI_5_LSB = TWO_N14;
00142 const std::vector<std::pair<int32_t, int32_t>> FNAV_OMEGA0_1_5_BIT({{80, 16}});
00143 constexpr double FNAV_OMEGA0_5_LSB = TWO_N15;
00144 const std::vector<std::pair<int32_t, int32_t>> FNAV_OMEGADOT_1_5_BIT({{96, 11}});

```

```

00145 constexpr double FNAV_OMEGADOT_5_LSB = TWO_N33;
00146 const std::vector<std::pair<int32_t, int32_t> FNAV_M0_1_5_BIT({{107, 16}});
00147 constexpr double FNAV_M0_5_LSB = TWO_N15;
00148 const std::vector<std::pair<int32_t, int32_t> FNAV_AF0_1_5_BIT({{123, 16}});
00149 constexpr double FNAV_AF0_5_LSB = TWO_N19;
00150 const std::vector<std::pair<int32_t, int32_t> FNAV_AF1_1_5_BIT({{139, 13}});
00151 constexpr double FNAV_AF1_5_LSB = TWO_N38;
00152 const std::vector<std::pair<int32_t, int32_t> FNAV_E5AHS_1_5_BIT({{152, 2}});
00153 const std::vector<std::pair<int32_t, int32_t> FNAV_SVI_D2_5_BIT({{154, 6}});
00154 const std::vector<std::pair<int32_t, int32_t> FNAV_DELTAA12_2_5_BIT({{160, 13}});
00155 const std::vector<std::pair<int32_t, int32_t> FNAV_E_2_5_BIT({{173, 11}});
00156 const std::vector<std::pair<int32_t, int32_t> FNAV_W_2_5_BIT({{184, 16}});
00157 const std::vector<std::pair<int32_t, int32_t> FNAV_DELTAI_2_5_BIT({{200, 11}});
00158 // const std::vector<std::pair<int, int> FNAV_Omega012_2_5_bit({{210, 4}});
00159
00160 // WORD 6 Almanac SVID2(1/2) SVID3
00161 const std::vector<std::pair<int32_t, int32_t> FNAV_IO_DA_6_BIT({{7, 4}});
00162 // const std::vector<std::pair<int, int> FNAV_Omega022_2_6_bit({{10, 12}});
00163 const std::vector<std::pair<int32_t, int32_t> FNAV_OMEGADOT_2_6_BIT({{23, 11}});
00164 const std::vector<std::pair<int32_t, int32_t> FNAV_M0_2_6_BIT({{34, 16}});
00165 const std::vector<std::pair<int32_t, int32_t> FNAV_AF0_2_6_BIT({{50, 16}});
00166 const std::vector<std::pair<int32_t, int32_t> FNAV_AF1_2_6_BIT({{66, 13}});
00167 const std::vector<std::pair<int32_t, int32_t> FNAV_E5AHS_2_6_BIT({{79, 2}});
00168 const std::vector<std::pair<int32_t, int32_t> FNAV_SVI_D3_6_BIT({{81, 6}});
00169 const std::vector<std::pair<int32_t, int32_t> FNAV_DELTAA12_3_6_BIT({{87, 13}});
00170 const std::vector<std::pair<int32_t, int32_t> FNAV_E_3_6_BIT({{100, 11}});
00171 const std::vector<std::pair<int32_t, int32_t> FNAV_W_3_6_BIT({{111, 16}});
00172 const std::vector<std::pair<int32_t, int32_t> FNAV_DELTAI_3_6_BIT({{127, 11}});
00173 const std::vector<std::pair<int32_t, int32_t> FNAV_OMEGA0_3_6_BIT({{138, 16}});
00174 const std::vector<std::pair<int32_t, int32_t> FNAV_OMEGADOT_3_6_BIT({{154, 11}});
00175 const std::vector<std::pair<int32_t, int32_t> FNAV_M0_3_6_BIT({{165, 16}});
00176 const std::vector<std::pair<int32_t, int32_t> FNAV_AF0_3_6_BIT({{181, 16}});
00177 const std::vector<std::pair<int32_t, int32_t> FNAV_AF1_3_6_BIT({{197, 13}});
00178 const std::vector<std::pair<int32_t, int32_t> FNAV_E5AHS_3_6_BIT({{210, 2}});
00179
00180
00181 /** \} */
00182 /** \} */
00183 #endif // GNSS_SDR_GALILEO_FNAV_H

```

## 11.711 galileo\_fnav\_message.h File Reference

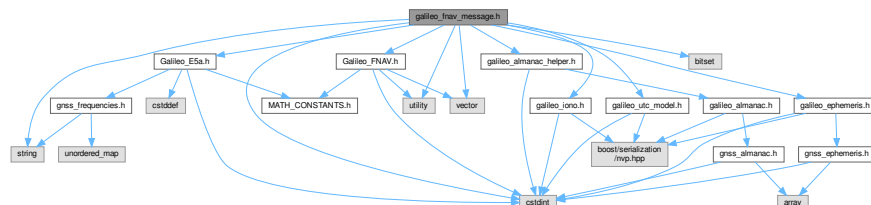
Implementation of a Galileo F/NAV Data message as described in Galileo OS SIS ICD Issue 1.2 (Nov. 2015)

```

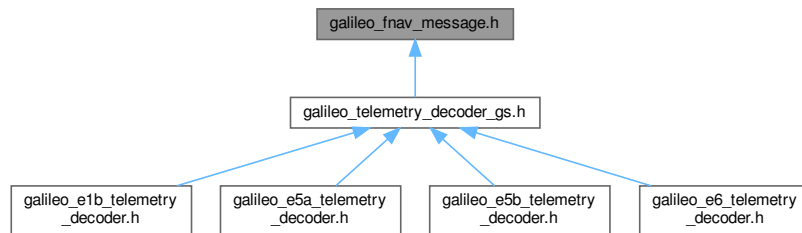
#include "Galileo_E5a.h"
#include "Galileo_FNAV.h"
#include "galileo_almanac_helper.h"
#include "galileo_ephemeris.h"
#include "galileo_iono.h"
#include "galileo_utc_model.h"
#include <bitset>
#include <stdint>
#include <string>
#include <utility>
#include <vector>

```

Include dependency graph for galileo\_fnav\_message.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Galileo\\_Fnav\\_Message](#)

*This class handles the Galileo F/NAV Data message, as described in the Galileo Open Service Signal in Space Interface Control Document (OS SIS ICD), Issue 2.0 (Jan. 2021). See [https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo\\_OS\\_SIS\\_ICD\\_v2.0.pdf](https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.0.pdf).*

### 11.711.1 Detailed Description

Implementation of a Galileo F/NAV Data message as described in Galileo OS SIS ICD Issue 1.2 (Nov. 2015)

#### Author

Marc Sales, 2014. marcsales92(at)gmail.com \based on work from:

- Javier Arribas, 2011. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [galileo\\_fnav\\_message.h](#).

---

### 11.712 galileo\_fnav\_message.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file galileo_fnav_message.h
00003  * \brief Implementation of a Galileo F/NAV Data message
00004  *        as described in Galileo OS SIS ICD Issue 1.2 (Nov. 2015)
00005  * \author Marc Sales, 2014. marcsales92(at)gmail.com
00006  * \based on work from:
00007  * <ul>
00008  * <li> Javier Arribas, 2011. jarribas(at)cttc.es
00009  * </li>
00010  *
00011  *
00012  * -----
00013  *
00014  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00015  * This file is part of GNSS-SDR.
00016  *
00017  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00018  * SPDX-License-Identifier: GPL-3.0-or-later
00019  *
00020  * -----
00021  */
00022
00023 #ifndef GNSS_SDR_GALILEO_FNAV_MESSAGE_H
00024 #define GNSS_SDR_GALILEO_FNAV_MESSAGE_H
00025
00026
00027 #include "Galileo_E5a.h"
00028 #include "Galileo_FNAV.h"
00029 #include "galileo_almanac_helper.h"
00030 #include "galileo_ephemeris.h"
  
```

```
00031 #include "galileo_iono.h"
00032 #include "galileo_utc_model.h"
00033 #include <bitset>
00034 #include <stdint>
00035 #include <string>
00036 #include <utility>
00037 #include <vector>
00038
00039 /** \addtogroup Core
00040  * \{ */
00041 /** \addtogroup System_Parameters
00042  * \{ */
00043
00044
00045 /*!
00046  * \brief This class handles the Galileo F/NAV Data message, as described in the
00047  * Galileo Open Service Signal in Space Interface Control Document (OS SIS ICD), Issue 2.0 (Jan.
00048 2021).
00049  * See https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo\_OS\_SIS\_ICD\_v2.0.pdf
00049 */
00050 class Galileo_Fnav_Message
00051 {
00052 public:
00053     Galileo_Fnav_Message() = default;
00054
00055     void split_page(const std::string& page_string);
00056     bool have_new_ephemeris();
00057     bool have_new_iono_and_GST();
00058     bool have_new_utc_model();
00059     bool have_new_almanac();
00060     Galileo_Ephemeris get_ephemeris() const;
00061     Galileo_Iono get_iono() const;
00062     Galileo_Utc_Model get_utc_model() const;
00063     Galileo_Almanac_Helper get_almanac() const;
00064
00065     inline int32_t get_TOW1() const
00066     {
00067         return FNAV_TOW_1;
00068     }
00069
00070     inline int32_t get_TOW2() const
00071     {
00072         return FNAV_TOW_2;
00073     }
00074
00075     inline int32_t get_TOW3() const
00076     {
00077         return FNAV_TOW_3;
00078     }
00079
00080     inline int32_t get_TOW4() const
00081     {
00082         return FNAV_TOW_4;
00083     }
00084
00085     inline bool get_flag_CRC_test() const
00086     {
00087         return flag_CRC_test;
00088     }
00089
00090     inline bool get_flag_TOW_set() const
00091     {
00092         return flag_TOW_set;
00093     }
00094
00095     inline void set_flag_TOW_set(bool flag_tow)
00096     {
00097         flag_TOW_set = flag_tow;
00098     }
00099
00100     inline bool is_TOW1_set() const
00101     {
00102         return flag_TOW_1;
00103     }
00104
00105     inline void set_TOW1_flag(bool flag_tow1)
00106     {
00107         flag_TOW_1 = flag_tow1;
00108     }
00109
00110     inline bool is_TOW2_set() const
00111     {
00112         return flag_TOW_2;
00113     }
00114
00115     inline void set_TOW2_flag(bool flag_tow2)
00116     {
```

```

00117         flag_TOW_2 = flag_tow2;
00118     }
00119
00120     inline bool is_TOW3_set() const
00121     {
00122         return flag_TOW_3;
00123     }
00124
00125     inline void set_TOW3_flag(bool flag_tow3)
00126     {
00127         flag_TOW_3 = flag_tow3;
00128     }
00129
00130     inline bool is_TOW4_set() const
00131     {
00132         return flag_TOW_4;
00133     }
00134
00135     inline void set_TOW4_flag(bool flag_tow4)
00136     {
00137         flag_TOW_4 = flag_tow4;
00138     }
00139
00140 private:
00141     bool CRC_test(const std::bitset<GALILEO_FNAV_DATA_FRAME_BITS>& bits, uint32_t checksum) const;
00142     void decode_page(const std::string& data);
00143     uint64_t read_navigation_unsigned(const std::bitset<GALILEO_FNAV_DATA_FRAME_BITS>& bits, const
std::vector<std::pair<int32_t, int32_t>& parameter) const;
00144     int64_t read_navigation_signed(const std::bitset<GALILEO_FNAV_DATA_FRAME_BITS>& bits, const
std::vector<std::pair<int32_t, int32_t>& parameter) const;
00145
00146     std::string omega0_1{};
00147     // std::string omega0_2{};
00148     // bool omega_flag{};
00149
00150     int32_t IOD_ephemeris{};
00151
00152     int32_t page_type{};
00153     // WORD 1 SVID, Clock correction, SISA, Ionospheric correction, BGD, GST, Signal
00154     // health and Data validity status
00155     int32_t FNAV_SV_ID_PRN_1{};
00156     int32_t FNAV_IODnav_1{};
00157     int32_t FNAV_t0c_1{};
00158     double FNAV_af0_1{};
00159     double FNAV_af1_1{};
00160     double FNAV_af2_1{};
00161     int32_t FNAV_SISA_1{};
00162     double FNAV_ai0_1{};
00163     double FNAV_ai1_1{};
00164     double FNAV_ai2_1{};
00165     double FNAV_BGD_1{};
00166     int32_t FNAV_E5ahs_1{};
00167     int32_t FNAV_WN_1{};
00168     int32_t FNAV_TOW_1{};
00169     bool FNAV_region1_1{};
00170     bool FNAV_region2_1{};
00171     bool FNAV_region3_1{};
00172     bool FNAV_region4_1{};
00173     bool FNAV_region5_1{};
00174     bool FNAV_E5advs_1{};
00175
00176     // WORD 2 Ephemeris (1/3) and GST
00177     int32_t FNAV_IODnav_2{};
00178     double FNAV_M0_2{};
00179     double FNAV_omegadot_2{};
00180     double FNAV_e_2{};
00181     double FNAV_a12_2{};
00182     double FNAV_omega0_2{};
00183     double FNAV_idot_2{};
00184     int32_t FNAV_WN_2{};
00185     int32_t FNAV_TOW_2{};
00186
00187     // WORD 3 Ephemeris (2/3) and GST
00188     int32_t FNAV_IODnav_3{};
00189     double FNAV_i0_3{};
00190     double FNAV_w_3{};
00191     double FNAV_deltan_3{};
00192     double FNAV_Cuc_3{};
00193     double FNAV_Cus_3{};
00194     double FNAV_Crc_3{};
00195     double FNAV_Crs_3{};
00196     int32_t FNAV_t0e_3{};
00197     int32_t FNAV_WN_3{};
00198     int32_t FNAV_TOW_3{};
00199
00200     // WORD 4 Ephemeris (3/3), GST-UTC conversion, GST-GPS conversion and TOW.
00201     // Note that the clock is repeated in this page type

```



```

00202     int32_t FNAV_IODnav_4{};
00203     double FNAV_Cic_4{};
00204     double FNAV_Cis_4{};
00205     double FNAV_A0_4{};
00206     double FNAV_A1_4{};
00207     int32_t FNAV_deltatls_4{};
00208     int32_t FNAV_t0t_4{};
00209     int32_t FNAV_WN0t_4{};
00210     int32_t FNAV_WN1sf_4{};
00211     int32_t FNAV_DN_4{};
00212     int32_t FNAV_deltatlsf_4{};
00213     int32_t FNAV_t0g_4{};
00214     double FNAV_A0g_4{};
00215     double FNAV_A1g_4{};
00216     int32_t FNAV_WN0g_4{};
00217     int32_t FNAV_TOW_4{};
00218
00219     // WORD 5 Almanac (SVID1 and SVID2(1/2)), Week Number and almanac reference time
00220     int32_t FNAV_IODa_5{};
00221     int32_t FNAV_WNa_5{};
00222     int32_t FNAV_t0a_5{};
00223     int32_t FNAV_SVID1_5{};
00224     double FNAV_Deltaa12_1_5{};
00225     double FNAV_e_1_5{};
00226     double FNAV_w_1_5{};
00227     double FNAV_deltai_1_5{};
00228     double FNAV_Omega0_1_5{};
00229     double FNAV_Omegadot_1_5{};
00230     double FNAV_M0_1_5{};
00231     double FNAV_af0_1_5{};
00232     double FNAV_af1_1_5{};
00233     uint32_t FNAV_E5ahs_1_5{};
00234     int32_t FNAV_SVID2_5{};
00235     double FNAV_Deltaa12_2_5{};
00236     double FNAV_e_2_5{};
00237     double FNAV_w_2_5{};
00238     double FNAV_deltai_2_5{};
00239
00240     // WORD 6 Almanac (SVID2(2/2) and SVID3)
00241     int32_t FNAV_IODa_6{};
00242     double FNAV_Omega0_2_6{};
00243     double FNAV_Omegadot_2_6{};
00244     double FNAV_M0_2_6{};
00245     double FNAV_af0_2_6{};
00246     double FNAV_af1_2_6{};
00247     int32_t FNAV_E5ahs_2_6{};
00248     int32_t FNAV_SVID3_6{};
00249     double FNAV_Deltaa12_3_6{};
00250     double FNAV_e_3_6{};
00251     double FNAV_w_3_6{};
00252     double FNAV_deltai_3_6{};
00253     double FNAV_Omega0_3_6{};
00254     double FNAV_Omegadot_3_6{};
00255     double FNAV_M0_3_6{};
00256     double FNAV_af0_3_6{};
00257     double FNAV_af1_3_6{};
00258     int32_t FNAV_E5ahs_3_6{};
00259
00260     bool flag_CRC_test{};
00261     bool flag_all_ephemeris{}; // Flag indicating that all words containing ephemeris have been
received
00262     bool flag_ephemeris_1{}; // Flag indicating that ephemeris 1/3 (word 2) have been received
00263     bool flag_ephemeris_2{}; // Flag indicating that ephemeris 2/3 (word 3) have been received
00264     bool flag_ephemeris_3{}; // Flag indicating that ephemeris 3/3 (word 4) have been received
00265
00266     bool flag_iono_and_GST{}; // Flag indicating that ionospheric and GST parameters (word 1) have
been received
00267     bool flag_TOW_1{};
00268     bool flag_TOW_2{};
00269     bool flag_TOW_3{};
00270     bool flag_TOW_4{};
00271     bool flag_TOW_set{}; // it is true when page 1,2,3 or 4 arrives
00272     bool flag_utc_model{}; // Flag indicating that utc model parameters (word 4) have been received
00273
00274     bool flag_all_almanac{}; // Flag indicating that all Almanac data have been received
00275     bool flag_almanac_1{}; // Flag indicating that almanac 1/2 (word 5) have been received
00276     bool flag_almanac_2{}; // Flag indicating that almanac 2/2 (word 6) have been received
00277 };
00278
00279
00280 /** \} */
00281 /** \} */
00282 #endif // GNSS_SDR_GALILEO_FNAV_MESSAGE_H

```

## 11.713 galileo\_has\_data.h File Reference

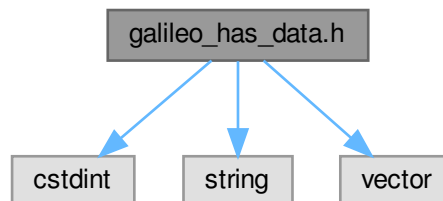
Class for Galileo HAS message type 1 data storage.

```
#include <cstdint>
```

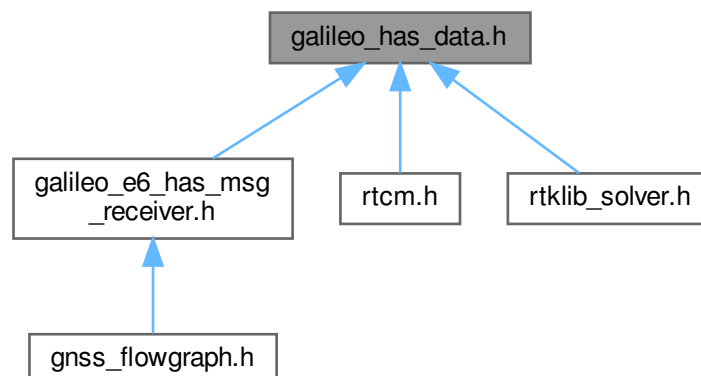
```
#include <string>
```

```
#include <vector>
```

Include dependency graph for galileo\_has\_data.h:



This graph shows which files directly or indirectly include this file:



### Classes

- struct [mt1\\_header](#)
- class [Galileo\\_HAS\\_data](#)

*This class is a storage for Galileo HAS message type 1, as defined in Galileo High Accuracy Service Signal-In-Space Interface Control Document (HAS SIS ICD) Issue 1.0, May 2022. See [https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo\\_HAS\\_SIS\\_ICD\\_v1.0.pdf](https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_HAS_SIS_ICD_v1.0.pdf).*

### 11.713.1 Detailed Description

Class for Galileo HAS message type 1 data storage.

## Author

Carles Fernandez-Prades, 2020-2022 cfernandez(at)cttc.es

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [galileo\\_has\\_data.h](#).

## 11.714 galileo\_has\_data.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file galileo_has_data.h
00003  * \brief Class for Galileo HAS message type 1 data storage
00004  * \author Carles Fernandez-Prades, 2020-2022 cfernandez(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_GALILEO_HAS_DATA_H
00019 #define GNSS_SDR_GALILEO_HAS_DATA_H
00020
00021 #include <stdint>
00022 #include <string>
00023 #include <vector>
00024
00025 /** \addtogroup Core
00026  * \{ */
00027 /** \addtogroup System_Parameters
00028  * \{ */
00029
00030 struct mt1_header
00031 {
00032     uint16_t toh;
00033     uint8_t mask_id;
00034     uint8_t iod_set_id;
00035     uint8_t reserved;
00036     bool mask_flag;
00037     bool orbit_correction_flag;
00038     bool clock_fullset_flag;
00039     bool clock_subset_flag;
00040     bool code_bias_flag;
00041     bool phase_bias_flag;
00042 };
00043
00044 /*!
00045 * \brief This class is a storage for Galileo HAS message type 1, as defined in
00046 * Galileo High Accuracy Service Signal-In-Space Interface Control Document
00047 * (HAS SIS ICD) Issue 1.0, May 2022.
00048 * See https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_HAS_SIS_ICD_v1.0.pdf
00049 */
00050 class Galileo_HAS_data
00051 {
00052 public:
00053     Galileo_HAS_data() = default;
00054
00055     std::vector<std::string> get_signals_in_mask(uint8_t nsys) const;          //!< Get a
00056     vector of Nsys std::string with signals in mask for system nsys, with 0 <= nsys < Nsys
00057     std::vector<std::string> get_signals_in_mask(const std::string& system) const;  //!< Get a
00058     vector of Nsys std::string with signals in mask for system ("GPS"/"Galileo")
00059     std::vector<std::string> get_systems_string() const;                      //!< Get Nsys
00060     system name strings
00061     std::vector<std::string> get_systems_subset_string() const;              //!< Get Nsat
00062     system name strings present in clock corrections subset
00063     std::vector<std::vector<float>> get_code_bias_m() const;                 //!< Get Nsat
00064     x Ncodes code biases in [m]
00065     std::vector<std::vector<float>> get_phase_bias_cycle() const;            //!< Get Nsat
00066     x Nphases phase biases in [cycles]
00067     std::vector<std::vector<float>> get_delta_clock_subset_correction_m() const;  //!< Get
00068     Nsys_sub vectors with Nsat_sub delta clock C0 corrections in [m]
00069     std::vector<float> get_delta_radial_m() const;                          //!< Get Nsat
00070     delta radial corrections in [m]
00071     std::vector<float> get_delta_radial_m(uint8_t nsys) const;              //!< Get
00072     delta radial corrections in [m] for system nsys, with 0 <= nsys < Nsys

```

```

00064     std::vector<float> get_delta_in_track_m() const;                //!< Get Nsat
delta in-track corrections in [m]
00065     std::vector<float> get_delta_in_track_m(uint8_t nsys) const;    //!< Get
delta in-track corrections in [m] for system nsys, with 0 <= nsys < Nsys
00066     std::vector<float> get_delta_cross_track_m() const;            //!< Get Nsat
delta cross-track corrections in [m]
00067     std::vector<float> get_delta_cross_track_m(uint8_t nsys) const; //!< Get
delta cross-track corrections in [m] for system nsys, with 0 <= nsys < Nsys
00068     std::vector<float> get_delta_clock_correction_m() const;        //!< Get Nsat
delta clock C0 corrections in [m]
00069     std::vector<float> get_delta_clock_correction_m(uint8_t nsys) const; //!< Get
delta clock C0 corrections in [m] for system nsys, with 0 <= nsys < Nsys
00070     std::vector<float> get_delta_clock_subset_correction_m(uint8_t nsys) const; //!< Get
delta clock C0 subset corrections in [m] for system nsys, with 0 <= nsys < Nsys
00071     std::vector<int> get_PRNs_in_mask(uint8_t nsys) const;          //!< Get PRNs
in mask for system nsys, with 0 <= nsys < Nsys
00072     std::vector<int> get_PRNs_in_mask(const std::string& system) const; //!< Get PRNs
in mask for system ("GPS"/"Galileo")
00073     std::vector<int> get_PRNs_in_submask(uint8_t nsys) const;       //!< Get PRNs
in submask for system nsys, with 0 <= nsys < Nsys
00074     std::vector<uint16_t> get_gnss_iod(uint8_t nsys) const;        //!< Get GNSS
IODs for for system nsys, with 0 <= nsys < Nsys
00075     std::vector<uint8_t> get_num_satellites() const;               //!< Get Nsys
number of satellites
00076     std::vector<uint8_t> get_num_subset_satellites() const;        //!< Get
Nsys_sub number of satellites
00077     float get_code_bias_m(const std::string& signal, int PRN) const; //!< Get code
bias in [m] for a given signal and PRN satellite
00078     float get_phase_bias_cycle(const std::string& signal, int PRN) const; //!< Get
phase bias in [cycles] for a given signal and PRN satellite
00079     float get_delta_radial_m(const std::string& system, int prn) const; //!< Get
orbital radial correction in [m] for a given system ("GPS"/"Galileo") and PRN
00080     float get_delta_in_track_m(const std::string& system, int prn) const; //!< Get
orbital in_track correction in [m] for a given system ("GPS"/"Galileo") and PRN
00081     float get_delta_cross_track_m(const std::string& system, int prn) const; //!< Get
orbital cross_track correction in [m] for a given system ("GPS"/"Galileo") and PRN
00082     float get_clock_correction_mult_m(const std::string& system, int prn) const; //!< Get
clock correction in [m], already multiplied by its Delta Clock Multiplier, for a given system
("GPS"/"Galileo") and PRN
00083     float get_clock_subset_correction_mult_m(const std::string& system, int prn) const; //!< Get
clock correction subset in [m], already multiplied by its Delta Clock Multiplier
00084     uint16_t get_nsat() const;                                     //!< Get
total number of satellites with corrections
00085     uint16_t get_nsat_sub() const;                                 //!< Get
number of satellites in clock subset corrections
00086     uint16_t get_validity_interval_s(uint8_t validity_interval_index) const; //!< Get
validity interval in [s] from the validity_interval_index
00087     uint16_t get_gnss_iod(const std::string& system, int prn) const; //!< Get GNSS
IOD from a given system ("GPS"/"Galileo") and PRN
00088     uint8_t get_gnss_id(int nsat) const;                          //!< Get GNSS
ID from the nsat satellite
00089
00090     // Mask
00091     std::vector<uint8_t> gnss_id_mask;                            //!< GNSS ID. See HAS SIS ICD 1.0 Section
5.2.1.1
00092     std::vector<uint64_t> satellite_mask;                          //!< SatM - Satellite Mask. See HAS SIS
ICD 1.0 Section 5.2.1.2
00093     std::vector<uint16_t> signal_mask;                             //!< SigM - Signal Mask. See HAS SIS ICD
1.0 Section 5.2.1.3
00094     std::vector<bool> cell_mask_availability_flag;                //!< CMAF - Cell Mask Availability Flag.
See HAS SIS ICD 1.0 Section 5.2.1.4
00095     std::vector<std::vector<std::vector<bool>>> cell_mask;          //!< CM - Cell Mask. See HAS SIS ICD 1.0
Section 5.2.1.5
00096     std::vector<uint8_t> nav_message;                             //!< NM - Navigation Message Index. See
HAS SIS ICD 1.0 Section 5.2.1.6
00097
00098     // Orbit corrections
00099     std::vector<uint16_t> gnss_iod;                                //!< IODref - Reference Issue of Data. See HAS SIS ICD
1.0 Table 26
00100     std::vector<int16_t> delta_radial;                             //!< DR - Delta Radial Correction. See HAS SIS ICD 1.0
Table 25
00101     std::vector<int16_t> delta_in_track;                          //!< DIT - Delta In-Track Correction. See HAS SIS ICD 1.0
Table 25
00102     std::vector<int16_t> delta_cross_track;                      //!< DCT - Delta Cross Correction. See HAS SIS ICD 1.0
Table 25
00103
00104     // Clock full-set corrections
00105     std::vector<uint8_t> delta_clock_multiplier;                 //!< DCM - Delta Clock Multipliers. See HAS SIS ICD
1.0 Section 5.2.3.1
00106     std::vector<int16_t> delta_clock_correction;                 //!< DCC - Delta Clock Corrections. See HAS SIS ICD
1.0 Section 5.2.3.2
00107
00108     // Clock subset corrections
00109     std::vector<uint8_t> gnss_id_clock_subset;                  //!< GNSS ID. Specific
GNSS to which the corrections refer. See HAS SIS ICD 1.0 Section 5.2.1.1
00110     std::vector<uint8_t> delta_clock_multiplier_clock_subset;    //!< DCM. Multiplier for
all Delta Clock corrections. See HAS SIS ICD 1.0 Section 5.2.3.1

```

```

00111     std::vector<uint64_t> satellite_submask;                                //!< SatMsub - Satellite
Subset Mask. See HAS SIS ICD 1.0 Section 5.2.4.1
00112     std::vector<std::vector<int16_t>> delta_clock_correction_clock_subset; //!< DCCsub - Delta Clock
Subset Corrections. See HAS SIS ICD 1.0 Section 5.2.4.1
00113
00114     // Code bias
00115     std::vector<std::vector<int16_t>> code_bias; //!< CB - Code bias for the m-th signal of the n-th
SV. See HAS SIS ICD 1.0 Section 5.2.5
00116
00117     // Phase bias
00118     std::vector<std::vector<int16_t>> phase_bias;                                //!< PB - Phase bias for the m-th
signal of the n-th SV. See HAS SIS ICD 1.0 Section 5.2.6
00119     std::vector<std::vector<uint8_t>> phase_discontinuity_indicator; //!< PDI - Phase Discontinuity
Indicator. See HAS SIS ICD 1.0 Section 5.2.6.
00120
00121     uint32_t tow; //!< Time of Week
00122
00123     mt1_header header; //!< MT1 Header parameters. See HAS SIS ICD 1.0 Section 5.1.1
00124     uint8_t has_status; //!< HASS - HAS Status (from HAS page header). See HAS SIS ICD 1.0 Section
3.1.1
00125     uint8_t message_id; //!< MID - Message ID (from HAS page header). See HAS SIS ICD 1.0 Section 3.1
00126
00127     uint8_t Nsys; //!< Number of GNSS for which corrections are provided. See HAS SIS ICD 1.0
Section 5.2.1
00128     uint8_t Nsys_sub; //!< Number of GNSS for which corrections are provided in clock subset
corrections. See HAS SIS ICD 1.0 Section 5.2.2.1
00129
00130     uint8_t validity_interval_index_orbit_corrections; //!< VI - Validity Interval Index for
Orbit corrections. See HAS SIS ICD 1.0 Section 5.2.2.1
00131     uint8_t validity_interval_index_clock_fullset_corrections; //!< VI - Validity Interval Index for
Clock full-set corrections. See HAS SIS ICD 1.0 Section 5.2.2.1
00132     uint8_t validity_interval_index_clock_subset_corrections; //!< VI - Validity Interval Index for
Clock subset corrections. See HAS SIS ICD 1.0 Section 5.2.2.1
00133     uint8_t validity_interval_index_code_bias_corrections; //!< VI - Validity Interval Index for
Code bias. See HAS SIS ICD 1.0 Section 5.2.2.1
00134     uint8_t validity_interval_index_phase_bias_corrections; //!< VI - Validity Interval Index for
Phase bias. See HAS SIS ICD 1.0 Section 5.2.2.1
00135 };
00136
00137
00138 /** \} */
00139 /** \} */
00140 #endif // GNSS_SDR_GALILEO_HAS_DATA_H

```

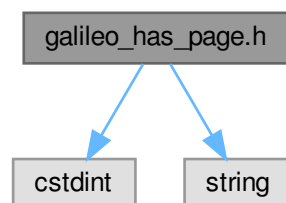
## 11.715 galileo\_has\_page.h File Reference

Class for Galileo HAS message page storage.

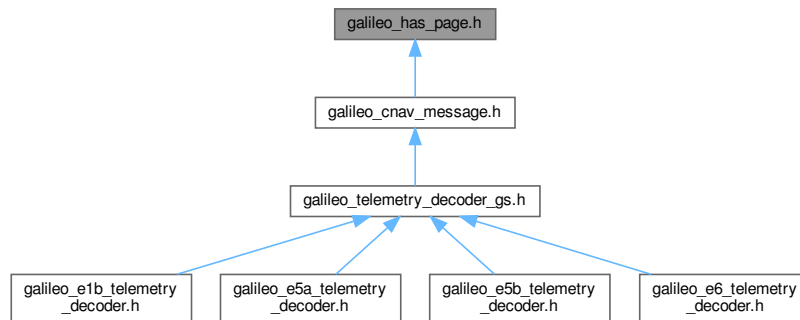
```
#include <cstdint>
```

```
#include <string>
```

Include dependency graph for galileo\_has\_page.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Galileo\\_HAS\\_page](#)

*This class is a storage for Galileo HAS message page, as defined in Galileo High Accuracy Service Signal-In-Space Interface Control Document (HAS SIS ICD) Issue 1.0, May 2022.*

## 11.715.1 Detailed Description

Class for Galileo HAS message page storage.

### Author

Carles Fernandez-Prades, 2021 cfernandez(at)cttc.es

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [galileo\\_has\\_page.h](#).

## 11.716 galileo\_has\_page.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file galileo_has_page.h
00003  * \brief Class for Galileo HAS message page storage
00004  * \author Carles Fernandez-Prades, 2021 cfernandez(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2022 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_GALILEO_HAS_PAGE_H
00019 #define GNSS_SDR_GALILEO_HAS_PAGE_H
00020
00021 #include <stdint>
00022 #include <string>
00023
00024 /** \addtogroup Core
00025  * \{ */
00026 /** \addtogroup System_Parameters
00027  * \{ */
00028
00029
00030  /*!
```

```

00031  * \brief This class is a storage for Galileo HAS message page, as defined in
00032  * Galileo High Accuracy Service Signal-In-Space Interface Control Document
00033  * (HAS SIS ICD) Issue 1.0, May 2022
00034  */
00035  class Galileo_HAS_page
00036  {
00037  public:
00038      Galileo_HAS_page() = default;
00039
00040      std::string has_message_string; //!< HAS message content
00041      uint64_t time_stamp{};          //!< HAS page time stamp, in [s]
00042      uint32_t tow{};                 //!< HAS page time of week, in [s]
00043
00044      // HAS page header
00045      uint8_t has_status{};           //!< HAS status
00046      uint8_t reserved{};             //!< HAS reserved field
00047      uint8_t message_type{};         //!< HAS message type (MT)
00048      uint8_t message_id{};           //!< HAS message ID (MID)
00049      uint8_t message_size{};         //!< HAS message size (MS)
00050      uint8_t message_page_id{};      //!< HAS message page ID (PID)
00051  };
00052
00053  /** \} */
00054  /** \} */
00055  /** \} */
00056  #endif // GNSS_SDR_GALILEO_HAS_PAGE_H

```

## 11.717 Galileo\_INAV.h File Reference

Galileo INAV message constants.

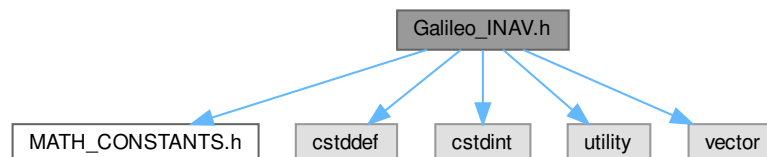
```
#include "MATH_CONSTANTS.h"
```

```
#include <cstdint>
```

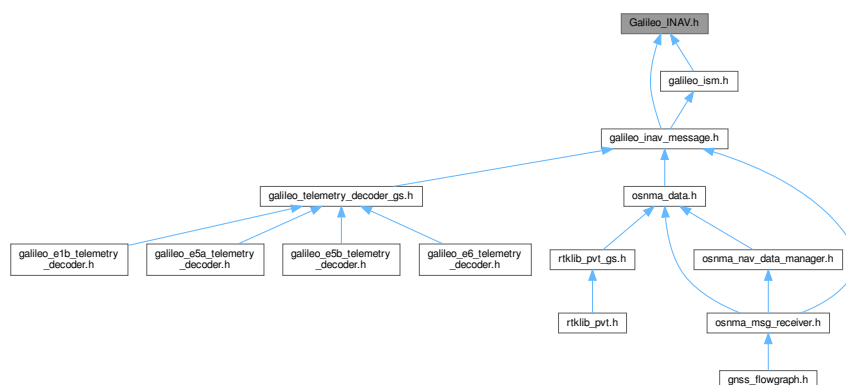
```
#include <utility>
```

```
#include <vector>
```

Include dependency graph for Galileo\_INAV.h:



This graph shows which files directly or indirectly include this file:



## Functions

- `const std::vector< std::pair< int32_t, int32_t > > TYPE {{{1, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > PAGE_TYPE_BIT {{{1, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > IOD_NAV_1_BIT {{{7, 10}}}`
- `const std::vector< std::pair< int32_t, int32_t > > T0_E_1_BIT {{{17, 14}}}`
- `const std::vector< std::pair< int32_t, int32_t > > M0_1_BIT {{{31, 32}}}`
- `const std::vector< std::pair< int32_t, int32_t > > E_1_BIT {{{63, 32}}}`
- `const std::vector< std::pair< int32_t, int32_t > > A_1_BIT {{{95, 32}}}`
- `const std::vector< std::pair< int32_t, int32_t > > IOD_NAV_2_BIT {{{7, 10}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA_0_2_BIT {{{17, 32}}}`
- `const std::vector< std::pair< int32_t, int32_t > > I_0_2_BIT {{{49, 32}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA_2_BIT {{{81, 32}}}`
- `const std::vector< std::pair< int32_t, int32_t > > I_DOT_2_BIT {{{113, 14}}}`
- `const std::vector< std::pair< int32_t, int32_t > > IOD_NAV_3_BIT {{{7, 10}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA_DOT_3_BIT {{{17, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DELTA_N_3_BIT {{{41, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > C_UC_3_BIT {{{57, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > C_US_3_BIT {{{73, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > C_RC_3_BIT {{{89, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > C_RS_3_BIT {{{105, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > SISA_3_BIT {{{121, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > IOD_NAV_4_BIT {{{7, 10}}}`
- `const std::vector< std::pair< int32_t, int32_t > > SV_ID_PRN_4_BIT {{{17, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > C_IC_4_BIT {{{23, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > C_IS_4_BIT {{{39, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > T0C_4_BIT {{{55, 14}}}`
- `const std::vector< std::pair< int32_t, int32_t > > AF0_4_BIT {{{69, 31}}}`
- `const std::vector< std::pair< int32_t, int32_t > > AF1_4_BIT {{{100, 21}}}`
- `const std::vector< std::pair< int32_t, int32_t > > AF2_4_BIT {{{121, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > SPARE_4_BIT {{{127, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > AI0_5_BIT {{{7, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > AI1_5_BIT {{{18, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > AI2_5_BIT {{{29, 14}}}`
- `const std::vector< std::pair< int32_t, int32_t > > REGION1_5_BIT {{{43, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > REGION2_5_BIT {{{44, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > REGION3_5_BIT {{{45, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > REGION4_5_BIT {{{46, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > REGION5_5_BIT {{{47, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > BGD_E1_E5A_5_BIT {{{48, 10}}}`
- `const std::vector< std::pair< int32_t, int32_t > > BGD_E1_E5B_5_BIT {{{58, 10}}}`
- `const std::vector< std::pair< int32_t, int32_t > > E5B_HS_5_BIT {{{68, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > E1_B_HS_5_BIT {{{70, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > E5B_DVS_5_BIT {{{72, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > E1_B_DVS_5_BIT {{{73, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > WN_5_BIT {{{74, 12}}}`
- `const std::vector< std::pair< int32_t, int32_t > > TOW_5_BIT {{{86, 20}}}`
- `const std::vector< std::pair< int32_t, int32_t > > SPARE_5_BIT {{{106, 23}}}`
- `const std::vector< std::pair< int32_t, int32_t > > A0_6_BIT {{{7, 32}}}`
- `const std::vector< std::pair< int32_t, int32_t > > A1_6_BIT {{{39, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DELTA_T_LS_6_BIT {{{63, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > T0T_6_BIT {{{71, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > W_NOT_6_BIT {{{79, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > WN_LSF_6_BIT {{{87, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DN_6_BIT {{{95, 3}}}`



- `const std::vector< std::pair< int32_t, int32_t > > DELTA_T_LSF_6_BIT {{{98, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > TOW_6_BIT {{{106, 20}}}`
- `const std::vector< std::pair< int32_t, int32_t > > IOD_A_7_BIT {{{7, 4}}}`
- `const std::vector< std::pair< int32_t, int32_t > > WN_A_7_BIT {{{11, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > T0A_7_BIT {{{13, 10}}}`
- `const std::vector< std::pair< int32_t, int32_t > > SVI_D1_7_BIT {{{23, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DELTA_A_7_BIT {{{29, 13}}}`
- `const std::vector< std::pair< int32_t, int32_t > > E_7_BIT {{{42, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA_7_BIT {{{53, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DELTA_I_7_BIT {{{69, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA0_7_BIT {{{80, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA_DOT_7_BIT {{{96, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > M0_7_BIT {{{107, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > IOD_A_8_BIT {{{7, 4}}}`
- `const std::vector< std::pair< int32_t, int32_t > > AF0_8_BIT {{{11, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > AF1_8_BIT {{{27, 13}}}`
- `const std::vector< std::pair< int32_t, int32_t > > E5B_HS_8_BIT {{{40, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > E1_B_HS_8_BIT {{{42, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > SVI_D2_8_BIT {{{44, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DELTA_A_8_BIT {{{50, 13}}}`
- `const std::vector< std::pair< int32_t, int32_t > > E_8_BIT {{{63, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA_8_BIT {{{74, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DELTA_I_8_BIT {{{90, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA0_8_BIT {{{101, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA_DOT_8_BIT {{{117, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > IOD_A_9_BIT {{{7, 4}}}`
- `const std::vector< std::pair< int32_t, int32_t > > WN_A_9_BIT {{{11, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > T0A_9_BIT {{{13, 10}}}`
- `const std::vector< std::pair< int32_t, int32_t > > M0_9_BIT {{{23, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > AF0_9_BIT {{{39, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > AF1_9_BIT {{{55, 13}}}`
- `const std::vector< std::pair< int32_t, int32_t > > E5B_HS_9_BIT {{{68, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > E1_B_HS_9_BIT {{{70, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > SVI_D3_9_BIT {{{72, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DELTA_A_9_BIT {{{78, 13}}}`
- `const std::vector< std::pair< int32_t, int32_t > > E_9_BIT {{{91, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA_9_BIT {{{102, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DELTA_I_9_BIT {{{118, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > IOD_A_10_BIT {{{7, 4}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA0_10_BIT {{{11, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA_DOT_10_BIT {{{27, 11}}}`
- `const std::vector< std::pair< int32_t, int32_t > > M0_10_BIT {{{38, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > AF0_10_BIT {{{54, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > AF1_10_BIT {{{70, 13}}}`
- `const std::vector< std::pair< int32_t, int32_t > > E5B_HS_10_BIT {{{83, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > E1_B_HS_10_BIT {{{85, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > A_0_G_10_BIT {{{87, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > A_1_G_10_BIT {{{103, 12}}}`
- `const std::vector< std::pair< int32_t, int32_t > > T_0_G_10_BIT {{{115, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > WN_0_G_10_BIT {{{123, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CED_DeltaAred_BIT {{{7, 5}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CED_exred_BIT {{{12, 13}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CED_eyred_BIT {{{25, 13}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CED_DeltaI0red_BIT {{{38, 17}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CED_Omega0red_BIT {{{55, 23}}}`

- `const std::vector< std::pair< int32_t, int32_t > > CED_lambda0red_BIT {{{78, 23}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CED_af0red_BIT {{{101, 22}}}`
- `const std::vector< std::pair< int32_t, int32_t > > CED_af1red_BIT {{{123, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > RS_IODNAV_LSBS {{{15, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ISM_CONSTELLATION_ID_BIT {{{7, 3}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ISM_SERVICE_LEVEL_ID_BIT {{{10, 3}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ISM_WN_BIT {{{13, 12}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ISM_T0_BIT {{{25, 9}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ISM_MASK_MSB_BIT {{{34, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ISM_MASK_BIT {{{35, 32}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ISM_PCONST_BIT {{{67, 4}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ISM_PSAT_BIT {{{71, 4}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ISM_URA_BIT {{{75, 4}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ISM_URE_BIT {{{79, 4}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ISM_BNOM_BIT {{{83, 4}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ISM_TVALIDITY_BIT {{{87, 4}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ISM_CRC_BIT {{{97, 32}}}`
- `const std::vector< std::pair< int32_t, int32_t > > TIME_0_BIT {{{7, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > WN_0_BIT {{{97, 12}}}`
- `const std::vector< std::pair< int32_t, int32_t > > TOW_0_BIT {{{109, 20}}}`

## Variables

- `constexpr double GALILEO_INAV_PAGE_PART_WITH_PREABLE_SECONDS = 2.04`  
*Page Duration + (Galileo I/NAV Preamble bits)\*(Galileo E5b-I tiered Code Period(seconds))*
- `constexpr uint32_t GALILEO_INAV_PAGE_SYMBOLS = 500`  
*The complete Galileo INAV page length.*
- `constexpr int32_t GALILEO_INAV_PREAMBLE_LENGTH_BITS = 10`
- `constexpr int32_t GALILEO_INAV_PREAMBLE_PERIOD_SYMBOLS = 250`
- `constexpr int32_t GALILEO_INAV_PAGE_PART_SYMBOLS = 250`  
*Each Galileo INAV pages are composed of two parts (even and odd) each of 250 symbols, including preamble. See Galileo ICD 4.3.2.*
- `constexpr int32_t GALILEO_INAV_PAGE_PART_SECONDS = 1`
- `constexpr int32_t GALILEO_INAV_PAGE_PART_MS = 1000`
- `constexpr int32_t GALILEO_INAV_PAGE_SECONDS = 2`
- `constexpr int32_t GALILEO_INAV_INTERLEAVER_ROWS = 8`
- `constexpr int32_t GALILEO_INAV_INTERLEAVER_COLS = 30`
- `constexpr int32_t GALILEO_TELEMETRY_RATE_BITS_SECOND = 250`
- `constexpr int32_t GALILEO_PAGE_TYPE_BITS = 6`
- `constexpr int32_t GALILEO_DATA_JK_BITS = 128`
- `constexpr int32_t GALILEO_DATA_FRAME_BITS = 196`
- `constexpr int32_t GALILEO_DATA_FRAME_BYTES = 25`
- `constexpr char GALILEO_INAV_PREAMBLE [11] = "0101100000"`
- `constexpr int32_t T0E_1_LSB = 60`
- `constexpr double M0_1_LSB = PI_TWO_N31`
- `constexpr double E_1_LSB = TWO_N33`
- `constexpr double A_1_LSB_GAL = TWO_N19`
- `constexpr double OMEGA_0_2_LSB = PI_TWO_N31`
- `constexpr double I_0_2_LSB = PI_TWO_N31`
- `constexpr double OMEGA_2_LSB = PI_TWO_N31`
- `constexpr double I_DOT_2_LSB = PI_TWO_N43`
- `constexpr double OMEGA_DOT_3_LSB = PI_TWO_N43`
- `constexpr double DELTA_N_3_LSB = PI_TWO_N43`
- `constexpr double C_UC_3_LSB = TWO_N29`
- `constexpr double C_US_3_LSB = TWO_N29`

- constexpr double [C\\_RC\\_3\\_LSB](#) = TWO\_N5
- constexpr double [C\\_RS\\_3\\_LSB](#) = TWO\_N5
- constexpr double [C\\_IC\\_4\\_LSB](#) = TWO\_N29
- constexpr double [C\\_IS\\_4\\_LSB](#) = TWO\_N29
- constexpr int32\_t [T0C\\_4\\_LSB](#) = 60
- constexpr double [AF0\\_4\\_LSB](#) = TWO\_N34
- constexpr double [AF1\\_4\\_LSB](#) = TWO\_N46
- constexpr double [AF2\\_4\\_LSB](#) = TWO\_N59
- constexpr double [AI0\\_5\\_LSB](#) = TWO\_N2
- constexpr double [AI1\\_5\\_LSB](#) = TWO\_N8
- constexpr double [AI2\\_5\\_LSB](#) = TWO\_N15
- constexpr double [BGD\\_E1\\_E5A\\_5\\_LSB](#) = TWO\_N32
- constexpr double [BGD\\_E1\\_E5B\\_5\\_LSB](#) = TWO\_N32
- constexpr double [A0\\_6\\_LSB](#) = TWO\_N30
- constexpr double [A1\\_6\\_LSB](#) = TWO\_N50
- constexpr int32\_t [T0T\\_6\\_LSB](#) = 3600
- constexpr int32\_t [T0A\\_7\\_LSB](#) = 600
- constexpr double [DELTA\\_A\\_7\\_LSB](#) = TWO\_N9
- constexpr double [E\\_7\\_LSB](#) = TWO\_N16
- constexpr double [OMEGA\\_7\\_LSB](#) = TWO\_N15
- constexpr double [DELTA\\_I\\_7\\_LSB](#) = TWO\_N14
- constexpr double [OMEGA0\\_7\\_LSB](#) = TWO\_N15
- constexpr double [OMEGA\\_DOT\\_7\\_LSB](#) = TWO\_N33
- constexpr double [M0\\_7\\_LSB](#) = TWO\_N15
- constexpr double [AF0\\_8\\_LSB](#) = TWO\_N19
- constexpr double [AF1\\_8\\_LSB](#) = TWO\_N38
- constexpr double [DELTA\\_A\\_8\\_LSB](#) = TWO\_N9
- constexpr double [E\\_8\\_LSB](#) = TWO\_N16
- constexpr double [OMEGA\\_8\\_LSB](#) = TWO\_N15
- constexpr double [DELTA\\_I\\_8\\_LSB](#) = TWO\_N14
- constexpr double [OMEGA0\\_8\\_LSB](#) = TWO\_N15
- constexpr double [OMEGA\\_DOT\\_8\\_LSB](#) = TWO\_N33
- constexpr int32\_t [T0A\\_9\\_LSB](#) = 600
- constexpr double [M0\\_9\\_LSB](#) = TWO\_N15
- constexpr double [AF0\\_9\\_LSB](#) = TWO\_N19
- constexpr double [AF1\\_9\\_LSB](#) = TWO\_N38
- constexpr double [DELTA\\_A\\_9\\_LSB](#) = TWO\_N9
- constexpr double [E\\_9\\_LSB](#) = TWO\_N16
- constexpr double [OMEGA\\_9\\_LSB](#) = TWO\_N15
- constexpr double [DELTA\\_I\\_9\\_LSB](#) = TWO\_N14
- constexpr double [OMEGA0\\_10\\_LSB](#) = TWO\_N15
- constexpr double [OMEGA\\_DOT\\_10\\_LSB](#) = TWO\_N33
- constexpr double [M0\\_10\\_LSB](#) = TWO\_N15
- constexpr double [AF0\\_10\\_LSB](#) = TWO\_N19
- constexpr double [AF1\\_10\\_LSB](#) = TWO\_N38
- constexpr double [A\\_0G\\_10\\_LSB](#) = TWO\_N35
- constexpr double [A\\_1G\\_10\\_LSB](#) = TWO\_N51
- constexpr int32\_t [T\\_0\\_G\\_10\\_LSB](#) = 3600
- constexpr double [CED\\_DeltaAred\\_LSB](#) = TWO\_P8
- constexpr double [CED\\_exred\\_LSB](#) = TWO\_N22
- constexpr double [CED\\_eyred\\_LSB](#) = TWO\_N22
- constexpr double [CED\\_DeltaI0red\\_LSB](#) = TWO\_N22
- constexpr double [CED\\_Omega0red\\_LSB](#) = TWO\_N22
- constexpr double [CED\\_lambda0red\\_LSB](#) = TWO\_N22
- constexpr double [CED\\_af0red\\_LSB](#) = TWO\_N26

- constexpr double [CED\\_af1red\\_LSB](#) = TWO\_N35
- constexpr size\_t [INAV\\_RS\\_SUBVECTOR\\_LENGTH](#) = 15
- constexpr size\_t [INAV\\_RS\\_PARITY\\_VECTOR\\_LENGTH](#) = 60
- constexpr size\_t [INAV\\_RS\\_INFO\\_VECTOR\\_LENGTH](#) = 58
- constexpr size\_t [INAV\\_RS\\_BUFFER\\_LENGTH](#) = 118
- constexpr int32\_t [BITS\\_IN\\_OCTET](#) = 8
- constexpr int32\_t [FIRST\\_RS\\_BIT](#) = 7
- constexpr int32\_t [FIRST\\_RS\\_BIT\\_AFTER\\_IODNAV](#) = 17
- constexpr int32\_t [GALILEO\\_ISM\\_CRC\\_DATA\\_BITS](#) = 96
- constexpr int32\_t [GALILEO\\_ISM\\_CRC\\_DATA\\_BYTES](#) = 12
- constexpr char [GALILEO\\_INAV\\_PLAIN\\_SSP1](#) [9] = "00000100"
- constexpr char [GALILEO\\_INAV\\_PLAIN\\_SSP2](#) [9] = "00101011"
- constexpr char [GALILEO\\_INAV\\_PLAIN\\_SSP3](#) [9] = "00101111"
- constexpr char [GALILEO\\_INAV\\_ENCODED\\_SSP1](#) [17] = "1110100100100101"
- constexpr char [GALILEO\\_INAV\\_ENCODED\\_SSP2](#) [17] = "0110110001001110"
- constexpr char [GALILEO\\_INAV\\_ENCODED\\_SSP3](#) [17] = "1101000000111110"

### 11.717.1 Detailed Description

Galileo INAV message constants.

Author

Carles Fernandez, 2020. [cfernandez\(at\)cttc.es](mailto:cfernandez(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [Galileo\\_INAV.h](#).

---

## 11.718 Galileo\_INAV.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file Galileo_INAV.h
00003  * \brief Galileo INAV message constants
00004  * \author Carles Fernandez, 2020. cfernandez(at)cttc.es
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_GALILEO_INAV_H
00019 #define GNSS_SDR_GALILEO_INAV_H
00020
00021 #include "MATH_CONSTANTS.h"
00022 #include <stdint>
00023 #include <stdint>
00024 #include <utility>
00025 #include <vector>
00026
00027 /** \addtogroup Core
00028  * \{ */
00029 /** \addtogroup System_Parameters
00030  * \{ */
00031
00032
00033 // Galileo INAV message structure
00034 constexpr double GALILEO_INAV_PAGE_PART_WITH_PREAMBLE_SECONDS = 2.04; //!< Page Duration + (Galileo
I/NAV Preamble bits)*(Galileo E5b-I tiered Code Period(seconds))
00035 constexpr uint32_t GALILEO_INAV_PAGE_SYMBOLS = 500; //!< The complete Galileo INAV
page length
00036 constexpr int32_t GALILEO_INAV_PREAMBLE_LENGTH_BITS = 10;
00037 constexpr int32_t GALILEO_INAV_PREAMBLE_PERIOD_SYMBOLS = 250;

```

```

00038 constexpr int32_t GALILEO_INAV_PAGE_PART_SYMBOLS = 250; //!< Each Galileo INAV pages are composed of
two parts (even and odd) each of 250 symbols, including preamble. See Galileo ICD 4.3.2
00039 constexpr int32_t GALILEO_INAV_PAGE_PART_SECONDS = 1; // a page part last 1 sec
00040 constexpr int32_t GALILEO_INAV_PAGE_PART_MS = 1000; // a page part last 1 sec
00041 constexpr int32_t GALILEO_INAV_PAGE_SECONDS = 2; // a full page last 2 sec
00042 constexpr int32_t GALILEO_INAV_INTERLEAVER_ROWS = 8;
00043 constexpr int32_t GALILEO_INAV_INTERLEAVER_COLS = 30;
00044 constexpr int32_t GALILEO_TELEMETRY_RATE_BITS_SECOND = 250; // bps
00045 constexpr int32_t GALILEO_PAGE_TYPE_BITS = 6;
00046 constexpr int32_t GALILEO_DATA_JK_BITS = 128;
00047 constexpr int32_t GALILEO_DATA_FRAME_BITS = 196;
00048 constexpr int32_t GALILEO_DATA_FRAME_BYTES = 25;
00049 constexpr char GALILEO_INAV_PREAMBLE[11] = "0101100000";
00050
00051 const std::vector<std::pair<int32_t, int32_t> TYPE({{1, 6}});
00052 const std::vector<std::pair<int32_t, int32_t> PAGE_TYPE_BIT({{1, 6}});
00053
00054 /* Page 1 - Word type 1: Ephemeris (1/4) */
00055 const std::vector<std::pair<int32_t, int32_t> IOD_NAV_1_BIT({{7, 10}});
00056 const std::vector<std::pair<int32_t, int32_t> T0_E_1_BIT({{17, 14}});
00057 constexpr int32_t T0E_1_LSB = 60;
00058 const std::vector<std::pair<int32_t, int32_t> M0_1_BIT({{31, 32}});
00059 constexpr double M0_1_LSB = PI_TWO_N31;
00060 const std::vector<std::pair<int32_t, int32_t> E_1_BIT({{63, 32}});
00061 constexpr double E_1_LSB = TWO_N33;
00062 const std::vector<std::pair<int32_t, int32_t> A_1_BIT({{95, 32}});
00063 constexpr double A_1_LSB_GAL = TWO_N19;
00064 // last two bits are reserved
00065
00066
00067 /* Page 2 - Word type 2: Ephemeris (2/4) */
00068 const std::vector<std::pair<int32_t, int32_t> IOD_NAV_2_BIT({{7, 10}});
00069 const std::vector<std::pair<int32_t, int32_t> OMEGA_0_2_BIT({{17, 32}});
00070 constexpr double OMEGA_0_2_LSB = PI_TWO_N31;
00071 const std::vector<std::pair<int32_t, int32_t> I_0_2_BIT({{49, 32}});
00072 constexpr double I_0_2_LSB = PI_TWO_N31;
00073 const std::vector<std::pair<int32_t, int32_t> OMEGA_2_BIT({{81, 32}});
00074 constexpr double OMEGA_2_LSB = PI_TWO_N31;
00075 const std::vector<std::pair<int32_t, int32_t> I_DOT_2_BIT({{113, 14}});
00076 constexpr double I_DOT_2_LSB = PI_TWO_N43;
00077 // last two bits are reserved
00078
00079 /* Word type 3: Ephemeris (3/4) and SISA */
00080 const std::vector<std::pair<int32_t, int32_t> IOD_NAV_3_BIT({{7, 10}});
00081 const std::vector<std::pair<int32_t, int32_t> OMEGA_DOT_3_BIT({{17, 24}});
00082 constexpr double OMEGA_DOT_3_LSB = PI_TWO_N43;
00083 const std::vector<std::pair<int32_t, int32_t> DELTA_N_3_BIT({{41, 16}});
00084 constexpr double DELTA_N_3_LSB = PI_TWO_N43;
00085 const std::vector<std::pair<int32_t, int32_t> C_UC_3_BIT({{57, 16}});
00086 constexpr double C_UC_3_LSB = TWO_N29;
00087 const std::vector<std::pair<int32_t, int32_t> C_US_3_BIT({{73, 16}});
00088 constexpr double C_US_3_LSB = TWO_N29;
00089 const std::vector<std::pair<int32_t, int32_t> C_RC_3_BIT({{89, 16}});
00090 constexpr double C_RC_3_LSB = TWO_N5;
00091 const std::vector<std::pair<int32_t, int32_t> C_RS_3_BIT({{105, 16}});
00092 constexpr double C_RS_3_LSB = TWO_N5;
00093 const std::vector<std::pair<int32_t, int32_t> SISA_3_BIT({{121, 8}});
00094
00095
00096 /* Word type 4: Ephemeris (4/4) and Clock correction parameters */
00097 const std::vector<std::pair<int32_t, int32_t> IOD_NAV_4_BIT({{7, 10}});
00098 const std::vector<std::pair<int32_t, int32_t> SV_ID_PRN_4_BIT({{17, 6}});
00099 const std::vector<std::pair<int32_t, int32_t> C_IC_4_BIT({{23, 16}});
00100 constexpr double C_IC_4_LSB = TWO_N29;
00101 const std::vector<std::pair<int32_t, int32_t> C_IS_4_BIT({{39, 16}});
00102 constexpr double C_IS_4_LSB = TWO_N29;
00103 const std::vector<std::pair<int32_t, int32_t> T0C_4_BIT({{55, 14}}); //
00104 constexpr int32_t T0C_4_LSB = 60;
00105 const std::vector<std::pair<int32_t, int32_t> AF0_4_BIT({{69, 31}}); //
00106 constexpr double AF0_4_LSB = TWO_N34;
00107 const std::vector<std::pair<int32_t, int32_t> AF1_4_BIT({{100, 21}}); //
00108 constexpr double AF1_4_LSB = TWO_N46;
00109 const std::vector<std::pair<int32_t, int32_t> AF2_4_BIT({{121, 6}});
00110 constexpr double AF2_4_LSB = TWO_N59;
00111 const std::vector<std::pair<int32_t, int32_t> SPARE_4_BIT({{127, 2}});
00112 // last two bits are reserved
00113
00114 /* Word type 5: Ionospheric correction, BGD, signal health and data validity status and GST */
00115 /* Ionospheric correction */
00116 /* Az */
00117 const std::vector<std::pair<int32_t, int32_t> AI0_5_BIT({{7, 11}}); //
00118 constexpr double AI0_5_LSB = TWO_N2;
00119 const std::vector<std::pair<int32_t, int32_t> AI1_5_BIT({{18, 11}}); //
00120 constexpr double AI1_5_LSB = TWO_N8;
00121 const std::vector<std::pair<int32_t, int32_t> AI2_5_BIT({{29, 14}}); //
00122 constexpr double AI2_5_LSB = TWO_N15;
00123 /* Ionospheric disturbance flag */

```

```

00124 const std::vector<std::pair<int32_t, int32_t>> REGION1_5_BIT({{43, 1}}); //
00125 const std::vector<std::pair<int32_t, int32_t>> REGION2_5_BIT({{44, 1}}); //
00126 const std::vector<std::pair<int32_t, int32_t>> REGION3_5_BIT({{45, 1}}); //
00127 const std::vector<std::pair<int32_t, int32_t>> REGION4_5_BIT({{46, 1}}); //
00128 const std::vector<std::pair<int32_t, int32_t>> REGION5_5_BIT({{47, 1}}); //
00129 const std::vector<std::pair<int32_t, int32_t>> BGD_E1_E5A_5_BIT({{48, 10}}); //
00130 constexpr double BGD_E1_E5A_5_LSB = TWO_N32;
00131 const std::vector<std::pair<int32_t, int32_t>> BGD_E1_E5B_5_BIT({{58, 10}}); //
00132 constexpr double BGD_E1_E5B_5_LSB = TWO_N32;
00133 const std::vector<std::pair<int32_t, int32_t>> E5B_HS_5_BIT({{68, 2}}); //
00134 const std::vector<std::pair<int32_t, int32_t>> E1_B_HS_5_BIT({{70, 2}}); //
00135 const std::vector<std::pair<int32_t, int32_t>> E5B_DVS_5_BIT({{72, 1}}); //
00136 const std::vector<std::pair<int32_t, int32_t>> E1_B_DVS_5_BIT({{73, 1}}); //
00137 /* GST */
00138 const std::vector<std::pair<int32_t, int32_t>> WN_5_BIT({{74, 12}});
00139 const std::vector<std::pair<int32_t, int32_t>> TOW_5_BIT({{86, 20}});
00140 const std::vector<std::pair<int32_t, int32_t>> SPARE_5_BIT({{106, 23}});
00141
00142
00143 /* Page 6 */
00144 const std::vector<std::pair<int32_t, int32_t>> A0_6_BIT({{7, 32}});
00145 constexpr double A0_6_LSB = TWO_N30;
00146 const std::vector<std::pair<int32_t, int32_t>> A1_6_BIT({{39, 24}});
00147 constexpr double A1_6_LSB = TWO_N50;
00148 const std::vector<std::pair<int32_t, int32_t>> DELTA_T_LS_6_BIT({{63, 8}});
00149 const std::vector<std::pair<int32_t, int32_t>> T0T_6_BIT({{71, 8}});
00150 constexpr int32_t T0T_6_LSB = 3600;
00151 const std::vector<std::pair<int32_t, int32_t>> W_NOT_6_BIT({{79, 8}});
00152 const std::vector<std::pair<int32_t, int32_t>> WN_LSF_6_BIT({{87, 8}});
00153 const std::vector<std::pair<int32_t, int32_t>> DN_6_BIT({{95, 3}});
00154 const std::vector<std::pair<int32_t, int32_t>> DELTA_T_LSF_6_BIT({{98, 8}});
00155 const std::vector<std::pair<int32_t, int32_t>> TOW_6_BIT({{106, 20}});
00156
00157
00158 /* Page 7 */
00159 const std::vector<std::pair<int32_t, int32_t>> IOD_A_7_BIT({{7, 4}});
00160 const std::vector<std::pair<int32_t, int32_t>> WN_A_7_BIT({{11, 2}});
00161 const std::vector<std::pair<int32_t, int32_t>> T0A_7_BIT({{13, 10}});
00162 constexpr int32_t T0A_7_LSB = 600;
00163 const std::vector<std::pair<int32_t, int32_t>> SVI_D1_7_BIT({{23, 6}});
00164 const std::vector<std::pair<int32_t, int32_t>> DELTA_A_7_BIT({{29, 13}});
00165 constexpr double DELTA_A_7_LSB = TWO_N9;
00166 const std::vector<std::pair<int32_t, int32_t>> E_7_BIT({{42, 11}});
00167 constexpr double E_7_LSB = TWO_N16;
00168 const std::vector<std::pair<int32_t, int32_t>> OMEGA_7_BIT({{53, 16}});
00169 constexpr double OMEGA_7_LSB = TWO_N15;
00170 const std::vector<std::pair<int32_t, int32_t>> DELTA_I_7_BIT({{69, 11}});
00171 constexpr double DELTA_I_7_LSB = TWO_N14;
00172 const std::vector<std::pair<int32_t, int32_t>> OMEGA0_7_BIT({{80, 16}});
00173 constexpr double OMEGA0_7_LSB = TWO_N15;
00174 const std::vector<std::pair<int32_t, int32_t>> OMEGA_DOT_7_BIT({{96, 11}});
00175 constexpr double OMEGA_DOT_7_LSB = TWO_N33;
00176 const std::vector<std::pair<int32_t, int32_t>> M0_7_BIT({{107, 16}});
00177 constexpr double M0_7_LSB = TWO_N15;
00178
00179
00180 /* Page 8 */
00181 const std::vector<std::pair<int32_t, int32_t>> IOD_A_8_BIT({{7, 4}});
00182 const std::vector<std::pair<int32_t, int32_t>> AF0_8_BIT({{11, 16}});
00183 constexpr double AF0_8_LSB = TWO_N19;
00184 const std::vector<std::pair<int32_t, int32_t>> AF1_8_BIT({{27, 13}});
00185 constexpr double AF1_8_LSB = TWO_N38;
00186 const std::vector<std::pair<int32_t, int32_t>> E5B_HS_8_BIT({{40, 2}});
00187 const std::vector<std::pair<int32_t, int32_t>> E1_B_HS_8_BIT({{42, 2}});
00188 const std::vector<std::pair<int32_t, int32_t>> SVI_D2_8_BIT({{44, 6}});
00189 const std::vector<std::pair<int32_t, int32_t>> DELTA_A_8_BIT({{50, 13}});
00190 constexpr double DELTA_A_8_LSB = TWO_N9;
00191 const std::vector<std::pair<int32_t, int32_t>> E_8_BIT({{63, 11}});
00192 constexpr double E_8_LSB = TWO_N16;
00193 const std::vector<std::pair<int32_t, int32_t>> OMEGA_8_BIT({{74, 16}});
00194 constexpr double OMEGA_8_LSB = TWO_N15;
00195 const std::vector<std::pair<int32_t, int32_t>> DELTA_I_8_BIT({{90, 11}});
00196 constexpr double DELTA_I_8_LSB = TWO_N14;
00197 const std::vector<std::pair<int32_t, int32_t>> OMEGA0_8_BIT({{101, 16}});
00198 constexpr double OMEGA0_8_LSB = TWO_N15;
00199 const std::vector<std::pair<int32_t, int32_t>> OMEGA_DOT_8_BIT({{117, 11}});
00200 constexpr double OMEGA_DOT_8_LSB = TWO_N33;
00201
00202
00203 /* Page 9 */
00204 const std::vector<std::pair<int32_t, int32_t>> IOD_A_9_BIT({{7, 4}});
00205 const std::vector<std::pair<int32_t, int32_t>> WN_A_9_BIT({{11, 2}});
00206 const std::vector<std::pair<int32_t, int32_t>> T0A_9_BIT({{13, 10}});
00207 constexpr int32_t T0A_9_LSB = 600;
00208 const std::vector<std::pair<int32_t, int32_t>> M0_9_BIT({{23, 16}});
00209 constexpr double M0_9_LSB = TWO_N15;
00210 const std::vector<std::pair<int32_t, int32_t>> AF0_9_BIT({{39, 16}});

```



```

00211 constexpr double AF0_9_LSB = TWO_N19;
00212 const std::vector<std::pair<int32_t, int32_t> AF1_9_BIT({{55, 13}});
00213 constexpr double AF1_9_LSB = TWO_N38;
00214 const std::vector<std::pair<int32_t, int32_t> E5B_HS_9_BIT({{68, 2}});
00215 const std::vector<std::pair<int32_t, int32_t> E1_B_HS_9_BIT({{70, 2}});
00216 const std::vector<std::pair<int32_t, int32_t> SVI_D3_9_BIT({{72, 6}});
00217 const std::vector<std::pair<int32_t, int32_t> DELTA_A_9_BIT({{78, 13}});
00218 constexpr double DELTA_A_9_LSB = TWO_N9;
00219 const std::vector<std::pair<int32_t, int32_t> E_9_BIT({{91, 11}});
00220 constexpr double E_9_LSB = TWO_N16;
00221 const std::vector<std::pair<int32_t, int32_t> OMEGA_9_BIT({{102, 16}});
00222 constexpr double OMEGA_9_LSB = TWO_N15;
00223 const std::vector<std::pair<int32_t, int32_t> DELTA_I_9_BIT({{118, 11}});
00224 constexpr double DELTA_I_9_LSB = TWO_N14;
00225
00226
00227 /* Page 10 */
00228 const std::vector<std::pair<int32_t, int32_t> IOD_A_10_BIT({{7, 4}});
00229 const std::vector<std::pair<int32_t, int32_t> OMEGA0_10_BIT({{11, 16}});
00230 constexpr double OMEGA0_10_LSB = TWO_N15;
00231 const std::vector<std::pair<int32_t, int32_t> OMEGA_DOT_10_BIT({{27, 11}});
00232 constexpr double OMEGA_DOT_10_LSB = TWO_N33;
00233 const std::vector<std::pair<int32_t, int32_t> M0_10_BIT({{38, 16}});
00234 constexpr double M0_10_LSB = TWO_N15;
00235 const std::vector<std::pair<int32_t, int32_t> AF0_10_BIT({{54, 16}});
00236 constexpr double AF0_10_LSB = TWO_N19;
00237 const std::vector<std::pair<int32_t, int32_t> AF1_10_BIT({{70, 13}});
00238 constexpr double AF1_10_LSB = TWO_N38;
00239 const std::vector<std::pair<int32_t, int32_t> E5B_HS_10_BIT({{83, 2}});
00240 const std::vector<std::pair<int32_t, int32_t> E1_B_HS_10_BIT({{85, 2}});
00241 const std::vector<std::pair<int32_t, int32_t> A_0_G_10_BIT({{87, 16}});
00242 constexpr double A_0G_10_LSB = TWO_N35;
00243 const std::vector<std::pair<int32_t, int32_t> A_1_G_10_BIT({{103, 12}});
00244 constexpr double A_1G_10_LSB = TWO_N51;
00245 const std::vector<std::pair<int32_t, int32_t> T_0_G_10_BIT({{115, 8}});
00246 constexpr int32_t T_0_G_10_LSB = 3600;
00247 const std::vector<std::pair<int32_t, int32_t> WN_0_G_10_BIT({{123, 6}});
00248
00249 /* Page 16 */
00250 constexpr double CED_DeltaAred_LSB = TWO_P8;
00251 const std::vector<std::pair<int32_t, int32_t> CED_DeltaAred_BIT({{7, 5}});
00252 constexpr double CED_exred_LSB = TWO_N22;
00253 const std::vector<std::pair<int32_t, int32_t> CED_exred_BIT({{12, 13}});
00254 constexpr double CED_eyred_LSB = TWO_N22;
00255 const std::vector<std::pair<int32_t, int32_t> CED_eyred_BIT({{25, 13}});
00256 constexpr double CED_DeltaI0red_LSB = TWO_N22;
00257 const std::vector<std::pair<int32_t, int32_t> CED_DeltaI0red_BIT({{38, 17}});
00258 constexpr double CED_Omega0red_LSB = TWO_N22;
00259 const std::vector<std::pair<int32_t, int32_t> CED_Omega0red_BIT({{55, 23}});
00260 constexpr double CED_lambda0red_LSB = TWO_N22;
00261 const std::vector<std::pair<int32_t, int32_t> CED_lambda0red_BIT({{78, 23}});
00262 constexpr double CED_af0red_LSB = TWO_N26;
00263 const std::vector<std::pair<int32_t, int32_t> CED_af0red_BIT({{101, 22}});
00264 constexpr double CED_aflred_LSB = TWO_N35;
00265 const std::vector<std::pair<int32_t, int32_t> CED_aflred_BIT({{123, 6}});
00266
00267 /* Pages 17, 18, 19, 20 */
00268 const std::vector<std::pair<int32_t, int32_t> RS_IODNAV_LSBS({{15, 2}});
00269 constexpr size_t INAV_RS_SUBVECTOR_LENGTH = 15;
00270 constexpr size_t INAV_RS_PARITY_VECTOR_LENGTH = 60;
00271 constexpr size_t INAV_RS_INFO_VECTOR_LENGTH = 58;
00272 constexpr size_t INAV_RS_BUFFER_LENGTH = 118;
00273 constexpr int32_t BITS_IN_OCTET = 8;
00274 constexpr int32_t FIRST_RS_BIT = 7;
00275 constexpr int32_t FIRST_RS_BIT_AFTER_IODNAV = 17;
00276
00277 /* Page 22 */
00278 const std::vector<std::pair<int32_t, int32_t> ISM_CONSTELLATION_ID_BIT({{7, 3}});
00279 const std::vector<std::pair<int32_t, int32_t> ISM_SERVICE_LEVEL_ID_BIT({{10, 3}});
00280 const std::vector<std::pair<int32_t, int32_t> ISM_WN_BIT({{13, 12}});
00281 const std::vector<std::pair<int32_t, int32_t> ISM_T0_BIT({{25, 9}});
00282 const std::vector<std::pair<int32_t, int32_t> ISM_MASK_MSB_BIT({{34, 1}});
00283 const std::vector<std::pair<int32_t, int32_t> ISM_MASK_BIT({{35, 32}});
00284 const std::vector<std::pair<int32_t, int32_t> ISM_PCONST_BIT({{67, 4}});
00285 const std::vector<std::pair<int32_t, int32_t> ISM_PSAT_BIT({{71, 4}});
00286 const std::vector<std::pair<int32_t, int32_t> ISM_URA_BIT({{75, 4}});
00287 const std::vector<std::pair<int32_t, int32_t> ISM_URE_BIT({{79, 4}});
00288 const std::vector<std::pair<int32_t, int32_t> ISM_BNOM_BIT({{83, 4}});
00289 const std::vector<std::pair<int32_t, int32_t> ISM_TVALIDITY_BIT({{87, 4}});
00290 const std::vector<std::pair<int32_t, int32_t> ISM_CRC_BIT({{97, 32}});
00291 constexpr int32_t GALILEO_ISM_CRC_DATA_BITS = 96;
00292 constexpr int32_t GALILEO_ISM_CRC_DATA_BYTES = 12;
00293
00294 /* Page 0 */
00295 const std::vector<std::pair<int32_t, int32_t> TIME_0_BIT({{7, 2}});
00296 const std::vector<std::pair<int32_t, int32_t> WN_0_BIT({{97, 12}});
00297 const std::vector<std::pair<int32_t, int32_t> TOW_0_BIT({{109, 20}});

```

```

00298
00299 /* Secondary Synchronization Patterns */
00300 constexpr char GALILEO_INAV_PLAIN_SSP1[9] = "00000100";
00301 constexpr char GALILEO_INAV_PLAIN_SSP2[9] = "00101011";
00302 constexpr char GALILEO_INAV_PLAIN_SSP3[9] = "00101111";
00303 constexpr char GALILEO_INAV_ENCODED_SSP1[17] = "1110100100100101";
00304 constexpr char GALILEO_INAV_ENCODED_SSP2[17] = "0110110001001110";
00305 constexpr char GALILEO_INAV_ENCODED_SSP3[17] = "1101000000111110";
00306
00307 /** \} */
00308 /** \} */
00309 #endif // GNSS_SDR_GALILEO_INAV_H

```

## 11.719 galileo\_inav\_message.h File Reference

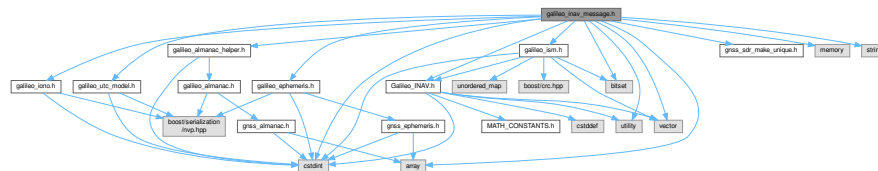
Implementation of a Galileo I/NAV Data message as described in Galileo OS SIS ICD Issue 2.0 (Jan. 2021)

```

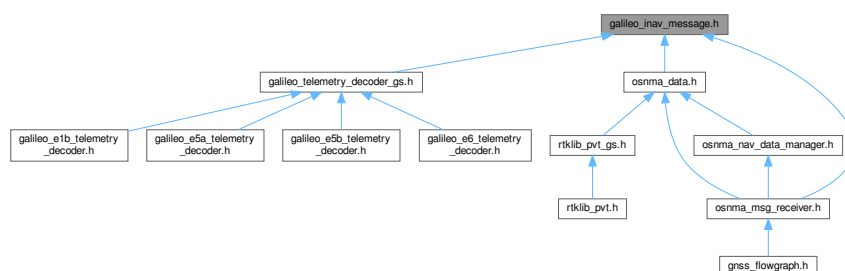
#include "Galileo_INAV.h"
#include "galileo_almanac_helper.h"
#include "galileo_ephemeris.h"
#include "galileo_iono.h"
#include "galileo_ism.h"
#include "galileo_utc_model.h"
#include "gnss_sdr_make_unique.h"
#include <array>
#include <bitset>
#include <cstdint>
#include <memory>
#include <string>
#include <utility>
#include <vector>

```

Include dependency graph for galileo\_inav\_message.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [OSNMA\\_msg](#)  
This class fills the [OSNMA\\_msg](#) structure with the data received from the telemetry blocks.
- class [Galileo\\_Inav\\_Message](#)



*This class handles the Galileo I/NAV Data message, as described in the Galileo Open Service Signal in Space Interface Control Document (OS SIS ICD), Issue 2.0 (Jan. 2021). See [https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo\\_OS\\_SIS\\_ICD\\_v2.0.pdf](https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.0.pdf).*

### 11.719.1 Detailed Description

Implementation of a Galileo I/NAV Data message as described in Galileo OS SIS ICD Issue 2.0 (Jan. 2021)

#### Author

Mara Branzanti 2013. mara.branzanti(at)gmail.com

Javier Arribas, 2013. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [galileo\\_inav\\_message.h](#).

---

## 11.720 galileo\_inav\_message.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file galileo_inav_message.h
00003  * \brief Implementation of a Galileo I/NAV Data message
00004  *        as described in Galileo OS SIS ICD Issue 2.0 (Jan. 2021)
00005  * \author Mara Branzanti 2013. mara.branzanti(at)gmail.com
00006  * \author Javier Arribas, 2013. jarribas(at)cttc.es
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019 #ifndef GNSS_SDR_GALILEO_INAV_MESSAGE_H
00020 #define GNSS_SDR_GALILEO_INAV_MESSAGE_H
00021
00022 #include "Galileo_INAV.h"
00023 #include "galileo_almanac_helper.h"
00024 #include "galileo_ephemeris.h"
00025 #include "galileo_iono.h"
00026 #include "galileo_ism.h"
00027 #include "galileo_utc_model.h"
00028 #include "gnss_sdr_make_unique.h" // for std::unique_ptr in C++11
00029 #include <array>
00030 #include <bitset>
00031 #include <cstdint>
00032 #include <memory>
00033 #include <string>
00034 #include <utility>
00035 #include <vector>
00036
00037 class ReedSolomon; // Forward declaration of the ReedSolomon class
00038
00039 /** \addtogroup Core
00040  *  \{ */
00041 /** \addtogroup System_Parameters
00042  *  \{ */
00043 /*!
00044  * \brief This class fills the OSNMA_msg structure with the data received from the telemetry blocks.
00045  */
00046 class OSNMA_msg
00047 {
00048 public:
00049     OSNMA_msg() = default;
00050     std::array<uint32_t, 15> mack{};
00051     std::array<uint8_t, 15> hkroot{};
00052     uint32_t PRN{}; // PRN_a authentication data PRN
00053     uint32_t WN_sf0{}; // Week number at the start of OSNMA subframe
00054     uint32_t TOW_sf0{}; // TOW at the start of OSNMA subframe
00055 };
00056
00057 /*!
00058  * \brief This class handles the Galileo I/NAV Data message, as described in the

```

```

00059  * Galileo Open Service Signal in Space Interface Control Document (OS SIS ICD), Issue 2.0 (Jan.
00060  2021).
00061  * See https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.0.pdf
00062  */
00063  class Galileo_Inav_Message
00064  {
00065  public:
00066      Galileo_Inav_Message();
00067      ~Galileo_Inav_Message();
00068
00069      /*
00070      * \brief Takes in input a page (Odd or Even) of 120 bit, split it according ICD 4.3.2.3 and join
00071      Data_k with Data_j
00072      */
00073      void split_page(std::string page_string, int32_t flag_even_word);
00074
00075      /*
00076      * \brief Returns true if new Ephemeris has arrived. The flag is set to false when the function is
00077      executed
00078      */
00079      bool have_new_ephemeris();
00080
00081      /*
00082      * \brief Returns true if new Iono model has arrived. The flag is set to false when the function
00083      is executed
00084      */
00085      bool have_new_iono_and_GST();
00086
00087      /*
00088      * \brief Returns true if new UTC model has arrived. The flag is set to false when the function is
00089      executed
00090      */
00091      bool have_new_utc_model();
00092
00093      /*
00094      * \brief Returns true if new UTC model has arrived. The flag is set to false when the function is
00095      executed
00096      */
00097      bool have_new_almanac();
00098
00099      /*
00100      * \brief Returns true if new Reduced CED parameters have arrived. The flag is set to false when
00101      the function is executed
00102      */
00103      bool have_new_reduced_ced();
00104
00105      /*
00106      * \brief Returns true if new ISM data have arrived. The flag is set to false when the function is
00107      executed
00108      */
00109      bool have_new_ism();
00110
00111      /*
00112      * \brief Returns true if new NMA data have arrived. The flag is set to false when the function is
00113      executed
00114      */
00115      bool have_new_nma();
00116
00117      /*
00118      * \brief Returns a Galileo_Ephemeris object filled with the latest navigation data received
00119      */
00120      Galileo_Ephemeris get_ephemeris() const;
00121
00122      /*
00123      * \brief Returns a Galileo_Iono object filled with the latest navigation data received
00124      */
00125      Galileo_Iono get_iono() const;
00126
00127      /*
00128      * \brief Returns a Galileo_Utc_Model object filled with the latest navigation data received
00129      */
00130      Galileo_Utc_Model get_utc_model() const;
00131
00132      /*
00133      * \brief Returns a Galileo_Almanac_Helper object filled with the latest navigation data received
00134      */
00135      Galileo_Almanac_Helper get_almanac() const;
00136
00137      /*
00138      * \brief Returns a Galileo_Ephemeris object filled with the latest reduced CED received
00139      */
00140      Galileo_Ephemeris get_reduced_ced() const;
00141
00142      /*
00143      * \brief Returns a Galileo_ISMs object filled with the latest ISM data received
00144      */

```

```

00137     Galileo_ISM get_galileo_ism() const;
00138
00139     /*
00140     * \brief Returns an OSNMA_msg object filled with the latest NMA message received. Resets msg
00141     buffer.
00142     */
00143     OSNMA_msg get_osnma_msg();
00144
00145     /*
00146     * @brief Retrieves the OSNMA ADKD 4 NAV bits. Resets the string.
00147     */
00148     std::string get_osnma_adkd_4_nav_bits();
00149
00150     /*
00151     * @brief Resets the OSNMA ADKD 4 NAV bits.
00152     */
00153     void reset_osnma_nav_bits_adkd4();
00154
00155     /*
00156     * @brief Retrieves the OSNMA ADKD 0/12 NAV bits. Resets the string.
00157     */
00158     std::string get_osnma_adkd_0_12_nav_bits();
00159
00160     /*
00161     * @brief Resets the OSNMA ADKD 0/12 NAV bits.
00162     */
00163     void reset_osnma_nav_bits_adkd0_12();
00164
00165     inline bool get_flag_CRC_test() const
00166     {
00167         return flag_CRC_test;
00168     }
00169
00170     inline bool get_flag_TOW_set() const
00171     {
00172         return flag_TOW_set;
00173     }
00174
00175     inline void set_flag_TOW_set(bool flag_tow)
00176     {
00177         flag_TOW_set = flag_tow;
00178     }
00179
00180     inline int32_t get_Galileo_week() const
00181     {
00182         return WN_0;
00183     }
00184
00185     inline int32_t get_TOW5() const
00186     {
00187         return TOW_5;
00188     }
00189
00190     inline int32_t get_TOW6() const
00191     {
00192         return TOW_6;
00193     }
00194
00195     inline bool is_TOW5_set() const
00196     {
00197         return flag_TOW_5;
00198     }
00199
00200     inline void set_TOW5_flag(bool flag_tow5)
00201     {
00202         flag_TOW_5 = flag_tow5;
00203     }
00204
00205     inline bool is_TOW6_set() const
00206     {
00207         return flag_TOW_6;
00208     }
00209
00210     inline void set_TOW6_flag(bool flag_tow6)
00211     {
00212         flag_TOW_6 = flag_tow6;
00213     }
00214
00215     inline int32_t get_TOW0() const
00216     {
00217         return TOW_0;
00218     }
00219
00220     inline bool is_TOW0_set() const
00221     {
00222         return flag_TOW_0;
00223     }

```

```

00223
00224     inline void set_TOW0_flag(bool flag_tow0)
00225     {
00226         flag_TOW_0 = flag_tow0;
00227     }
00228
00229     inline bool get_flag_GGTO() const
00230     {
00231         return (flag_GGTO_1 == true and flag_GGTO_2 == true and flag_GGTO_3 == true and flag_GGTO_4 ==
true);
00232     }
00233
00234     inline double get_A0G() const
00235     {
00236         return A_0G_10;
00237     }
00238
00239     inline double get_A1G() const
00240     {
00241         return A_1G_10;
00242     }
00243
00244     inline double get_t0G() const
00245     {
00246         return t_0G_10;
00247     }
00248
00249     inline double get_WN0G() const
00250     {
00251         return WN_0G_10;
00252     }
00253
00254     /*
00255     * \brief Initialize PRN field so we do not need to wait for page 4.
00256     */
00257     inline void init_PRN(uint32_t prn)
00258     {
00259         SV_ID_PRN_4 = prn;
00260         nma_msg.PRN = prn;
00261         nma_msg.mack = std::array<uint32_t, 15>{};
00262         nma_msg.hkroot = std::array<uint8_t, 15>{};
00263         page_position_in_inav_subframe = 255;
00264         nma_position_filled = std::array<int8_t, 15>{};
00265     }
00266
00267     /*
00268     * \brief Enable Reed-Solomon in Galileo E1B
00269     */
00270     inline void enable_reed_solomon()
00271     {
00272         enable_rs = true;
00273     }
00274
00275 private:
00276     bool CRC_test(const std::bitset<GALILEO_DATA_FRAME_BITS>& bits, uint32_t checksum) const;
00277     bool read_navigation_bool(const std::bitset<GALILEO_DATA_JK_BITS>& bits, const
std::vector<std::pair<int32_t, int32_t>& parameter) const;
00278     uint64_t read_navigation_unsigned(const std::bitset<GALILEO_DATA_JK_BITS>& bits, const
std::vector<std::pair<int32_t, int32_t>& parameter) const;
00279     uint64_t read_page_type_unsigned(const std::bitset<GALILEO_PAGE_TYPE_BITS>& bits, const
std::vector<std::pair<int32_t, int32_t>& parameter) const;
00280     int64_t read_navigation_signed(const std::bitset<GALILEO_DATA_JK_BITS>& bits, const
std::vector<std::pair<int32_t, int32_t>& parameter) const;
00281     uint8_t read_octet_unsigned(const std::bitset<GALILEO_DATA_JK_BITS>& bits, const
std::vector<std::pair<int32_t, int32_t>& parameter) const;
00282     void read_page_1(const std::bitset<GALILEO_DATA_JK_BITS>& data_bits);
00283     void read_page_2(const std::bitset<GALILEO_DATA_JK_BITS>& data_bits);
00284     void read_page_3(const std::bitset<GALILEO_DATA_JK_BITS>& data_bits);
00285     void read_page_4(const std::bitset<GALILEO_DATA_JK_BITS>& data_bits);
00286     std::bitset<GALILEO_DATA_JK_BITS> regenerate_page_1(const std::vector<uint8_t>& decoded) const;
00287     std::bitset<GALILEO_DATA_JK_BITS> regenerate_page_2(const std::vector<uint8_t>& decoded) const;
00288     std::bitset<GALILEO_DATA_JK_BITS> regenerate_page_3(const std::vector<uint8_t>& decoded) const;
00289     std::bitset<GALILEO_DATA_JK_BITS> regenerate_page_4(const std::vector<uint8_t>& decoded) const;
00290
00291     Galileo_ISM gal_ism{};
00292     std::string page_Even{};
00293
00294     std::vector<uint8_t> rs_buffer; // Reed-Solomon buffer
00295     std::unique_ptr<ReedSolomon> rs; // The Reed-Solomon decoder
00296     std::vector<int> inav_rs_pages; // Pages 1,2,3,4,17,18,19,20. Holds 1 if the page has arrived, 0
otherwise.
00297
00298     int32_t page_jk_decoder(const char* data_jk);
00299     int32_t IOD_ephemeris{};
00300
00301     // Word type 1: Ephemeris (1/4)
00302     int32_t IOD_nav_1{}; // IOD_nav page 1

```

```

00303     int32_t t0e_1{};           // Ephemeris reference time [s]
00304     double M0_1{};           // Mean anomaly at reference time [rad]
00305     double e_1{};            // Eccentricity
00306     double A_1{};            // Square root of the semi-major axis [meters^1/2]
00307
00308     // Word type 2: Ephemeris (2/4)
00309     int32_t IOD_nav_2{};      // IOD_nav page 2
00310     double OMEGA_0_2{};      // Longitude of ascending node of orbital plane at weekly epoch [rad]
00311     double i_0_2{};          // Inclination angle at reference time [rad]
00312     double omega_2{};         // Argument of perigee [rad]
00313     double iDot_2{};          // Rate of inclination angle [rad/sec]
00314
00315     // Word type 3: Ephemeris (3/4) and SISA
00316     int32_t IOD_nav_3{};
00317     int32_t SISA_3{};
00318     double OMEGA_dot_3{};     // Rate of right ascension [rad/sec]
00319     double delta_n_3{};       // Mean motion difference from computed value [rad/sec]
00320     double C_uc_3{};          // Amplitude of the cosine harmonic correction term to the argument of
latitude [radians]
00321     double C_us_3{};          // Amplitude of the sine harmonic correction term to the argument of
latitude [radians]
00322     double C_rc_3{};          // Amplitude of the cosine harmonic correction term to the orbit radius
[meters]
00323     double C_rs_3{};          // Amplitude of the sine harmonic correction term to the orbit radius
[meters]
00324
00325     // Word type 4: Ephemeris (4/4) and Clock correction parameters*/
00326     int32_t IOD_nav_4{};      //
00327     int32_t SV_ID_PRN_4{};    //
00328     double C_ic_4{};          // Amplitude of the cosine harmonic correction term to the angle of
inclination [radians]
00329     double C_is_4{};          // Amplitude of the sine harmonic correction term to the angle of
inclination [radians]
00330
00331     // Clock correction parameters
00332     int32_t t0c_4{};          // Clock correction data reference Time of Week [sec]
00333     double af0_4{};           // SV clock bias correction coefficient [s]
00334     double af1_4{};           // SV clock drift correction coefficient [s/s]
00335     double af2_4{};           // clock drift rate correction coefficient [s/s^2]
00336     double spare_4{};
00337
00338     // Word type 5: Ionospheric correction, BGD, signal health and data validity status and GST*/
00339     // Ionospheric correction
00340     double ai0_5{};           // Effective Ionisation Level 1st order parameter [sfu]
00341     double ai1_5{};           // Effective Ionisation Level 2st order parameter [sfu/degree]
00342     double ai2_5{};           // Effective Ionisation Level 3st order parameter [sfu/degree]
00343     double BGD_E1E5a_5{};     // E1-E5a Broadcast Group Delay [s]
00344     double BGD_E1E5b_5{};     // E1-E5b Broadcast Group Delay [s]
00345     int32_t E5b_HS_5{};       // E5b Signal Health Status
00346     int32_t E1B_HS_5{};       // E1B Signal Health Status
00347
00348     // Ionospheric disturbance flag
00349     bool Region1_flag_5{};     // Ionospheric Disturbance Flag for region 1
00350     bool Region2_flag_5{};     // Ionospheric Disturbance Flag for region 2
00351     bool Region3_flag_5{};     // Ionospheric Disturbance Flag for region 3
00352     bool Region4_flag_5{};     // Ionospheric Disturbance Flag for region 4
00353     bool Region5_flag_5{};     // Ionospheric Disturbance Flag for region 5
00354     bool E5b_DVS_5{};          // E5b Data Validity Status
00355     bool E1B_DVS_5{};          // E1B Data Validity Status
00356
00357     // GST
00358     int32_t WN_5{};
00359     int32_t TOW_5{};
00360     double spare_5{};
00361
00362     // Word type 6: GST-UTC conversion parameters
00363     double A0_6{};
00364     double A1_6{};
00365     int32_t Delta_tLS_6{};
00366     int32_t t0t_6{};
00367     int32_t WNt_6{};
00368     int32_t WN_LSF_6{};
00369     int32_t DN_6{};
00370     int32_t Delta_tLSF_6{};
00371     int32_t TOW_6{};
00372
00373     // Word type 7: Almanac for SVID1 (1/2), almanac reference time and almanac reference week number
00374     int32_t IOD_a_7{};
00375     int32_t WN_a_7{};
00376     int32_t t0a_7{};
00377     int32_t SVID1_7{};
00378     double DELTA_A_7{};
00379     double e_7{};
00380     double omega_7{};
00381     double delta_i_7{};
00382     double Omega0_7{};
00383     double Omega_dot_7{};

```

```

00384     double M0_7{};
00385
00386     // Word type 8: Almanac for SVID1 (2/2) and SVID2 (1/2)
00387     int32_t IOD_a_8{};
00388     int32_t E5b_HS_8{};
00389     int32_t E1B_HS_8{};
00390     int32_t SVID2_8{};
00391     double af0_8{};
00392     double af1_8{};
00393     double DELTA_A_8{};
00394     double e_8{};
00395     double omega_8{};
00396     double delta_i_8{};
00397     double Omega0_8{};
00398     double Omega_dot_8{};
00399
00400     // Word type 9: Almanac for SVID2 (2/2) and SVID3 (1/2)
00401     int32_t IOD_a_9{};
00402     int32_t WN_a_9{};
00403     int32_t t0a_9{};
00404     double M0_9{};
00405     double af0_9{};
00406     double af1_9{};
00407     int32_t E5b_HS_9{};
00408     int32_t E1B_HS_9{};
00409     int32_t SVID3_9{};
00410     double DELTA_A_9{};
00411     double e_9{};
00412     double omega_9{};
00413     double delta_i_9{};
00414
00415     // Word type 10: Almanac for SVID3 (2/2) and GST-GPS conversion parameters
00416     int32_t IOD_a_10{};
00417     double Omega0_10{};
00418     double Omega_dot_10{};
00419     double M0_10{};
00420     double af0_10{};
00421     double af1_10{};
00422     int32_t E5b_HS_10{};
00423     int32_t E1B_HS_10{};
00424
00425     // GST-GPS conversion
00426     double A_0G_10{}; // Constant term of the offset Delta t systems
00427     double A_1G_10{}; // Rate of change of the offset Delta t systems
00428     int32_t t_0G_10{}; // Reference time for Galileo/GPS Time Offset (GGTO) data
00429     int32_t WN_0G_10{}; // Week Number of Galileo/GPS Time Offset (GGTO) reference
00430
00431     // Word type 0: I/NAV Spare Word
00432     int32_t Time_0{};
00433     int32_t WN_0{};
00434     int32_t TOW_0{};
00435
00436     // Word type 16: Reduced Clock and Ephemeris Data (CED) parameters
00437     double ced_DeltaAred{};
00438     double ced_exred{};
00439     double ced_eyred{};
00440     double ced_DeltaI0red{};
00441     double ced_Omega0red{};
00442     double ced_lambda0red{};
00443     double ced_af0red{};
00444     double ced_af1red{};
00445
00446     double Galileo_satClkDrift{};
00447
00448     int32_t current_IODnav{};
00449
00450     // OSNMA
00451     uint32_t mack_sis{};
00452     uint8_t hkroot_sis{};
00453     uint8_t page_position_in_inav_subframe{255};
00454     std::array<int8_t, 15> nma_position_filled{};
00455     OSNMA_msg nma_msg{};
00456     std::string nav_bits_adkd_4{};
00457     std::string nav_bits_word_6{};
00458     std::string nav_bits_word_10{};
00459     std::string nav_bits_adkd_0_12{};
00460     std::string nav_bits_word_1{};
00461     std::string nav_bits_word_2{};
00462     std::string nav_bits_word_3{};
00463     std::string nav_bits_word_4{};
00464     std::string nav_bits_word_5{};
00465
00466     uint8_t IODnav_LSB17{};
00467     uint8_t IODnav_LSB18{};
00468     uint8_t IODnav_LSB19{};
00469     uint8_t IODnav_LSB20{};
00470

```

```

00471     uint8_t ism_constellation_id{};
00472     uint8_t ism_service_level_id{};
00473
00474     bool flag_CRC_test{};
00475     bool flag_all_ephemeris{}; // Flag indicating that all words containing ephemeris have been
received
00476     bool flag_ephemeris_1{}; // Flag indicating that ephemeris 1/4 (word 1) have been received
00477     bool flag_ephemeris_2{}; // Flag indicating that ephemeris 2/4 (word 2) have been received
00478     bool flag_ephemeris_3{}; // Flag indicating that ephemeris 3/4 (word 3) have been received
00479     bool flag_ephemeris_4{}; // Flag indicating that ephemeris 4/4 (word 4) have been received
00480
00481     bool flag_iono_and_GST{}; // Flag indicating that ionospheric and GST parameters (word 5) have
been received
00482     bool flag_TOW_5{};
00483     bool flag_TOW_6{};
00484     bool flag_TOW_0{};
00485     bool flag_TOW_set{}; // it is true when page 5 or page 6 arrives
00486     bool flag_utc_model{}; // Flag indicating that utc model parameters (word 6) have been received
00487
00488     bool flag_all_almanac{}; // Flag indicating that all Almanac data have been received
00489     bool flag_almanac_1{}; // Flag indicating that almanac 1/4 (word 7) have been received
00490     bool flag_almanac_2{}; // Flag indicating that almanac 2/4 (word 8) have been received
00491     bool flag_almanac_3{}; // Flag indicating that almanac 3/4 (word 9) have been received
00492     bool flag_almanac_4{}; // Flag indicating that almanac 4/4 (word 10) have been received
00493
00494     bool flag_GGTO_1{};
00495     bool flag_GGTO_2{};
00496     bool flag_GGTO_3{};
00497     bool flag_GGTO_4{};
00498
00499     bool flag_CED{};
00500     bool enable_rs{};
00501     bool have_ISM{};
00502 };
00503
00504
00505 /** \} */
00506 /** \} */
00507 #endif // GNSS_SDR_GALILEO_INAV_MESSAGE_H

```

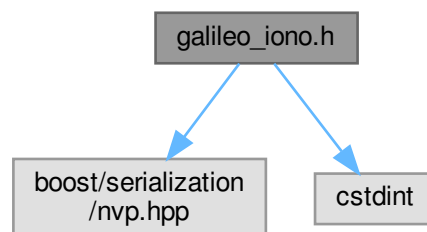
## 11.721 galileo\_iono.h File Reference

Interface of a Galileo Ionospheric Model storage.

```
#include <boost/serialization/nvp.hpp>
```

```
#include <cstdint>
```

Include dependency graph for galileo\_iono.h:







```

00035  * See https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.0.pdf
00036  */
00037  class Galileo_Iono
00038  {
00039  public:
00040      /*!
00041       * Default constructor
00042       */
00043      Galileo_Iono() = default;
00044
00045      // Ionospheric correction
00046      double ai0{}; //!< Effective Ionisation Level 1st order parameter [sfu]
00047      double ai1{}; //!< Effective Ionisation Level 2st order parameter [sfu/degree]
00048      double ai2{}; //!< Effective Ionisation Level 3st order parameter [sfu/degree]
00049
00050      // from page 5 (UTC) to have a timestamp
00051      int32_t tow{}; //!< UTC data reference Time of Week [s]
00052      int32_t WN{};  //!< UTC data reference Week number [week]
00053
00054      // Ionospheric disturbance flag
00055      bool Region1_flag{}; //!< Ionospheric Disturbance Flag for region 1
00056      bool Region2_flag{}; //!< Ionospheric Disturbance Flag for region 2
00057      bool Region3_flag{}; //!< Ionospheric Disturbance Flag for region 3
00058      bool Region4_flag{}; //!< Ionospheric Disturbance Flag for region 4
00059      bool Region5_flag{}; //!< Ionospheric Disturbance Flag for region 5
00060
00061      template <class Archive>
00062
00063      /*!
00064       * \brief Serialize is a boost standard method to be called by the boost XML serialization.
00065       * Here is used to save the iono data on disk file.
00066       */
00067      inline void serialize(Archive& archive, const unsigned int version)
00068      {
00069          if (version)
00070          {
00071              };
00072          archive& BOOST_SERIALIZATION_NVP(ai0);
00073          archive& BOOST_SERIALIZATION_NVP(ai1);
00074          archive& BOOST_SERIALIZATION_NVP(ai2);
00075          archive& BOOST_SERIALIZATION_NVP(tow);
00076          archive& BOOST_SERIALIZATION_NVP(WN);
00077          archive& BOOST_SERIALIZATION_NVP(Region1_flag);
00078          archive& BOOST_SERIALIZATION_NVP(Region2_flag);
00079          archive& BOOST_SERIALIZATION_NVP(Region3_flag);
00080          archive& BOOST_SERIALIZATION_NVP(Region4_flag);
00081          archive& BOOST_SERIALIZATION_NVP(Region5_flag);
00082      }
00083  };
00084
00085
00086  /** @} */
00087  /** @} */
00088  #endif // GNSS_SDR_GALILEO_IONO_H

```

## 11.723 galileo\_ism.h File Reference

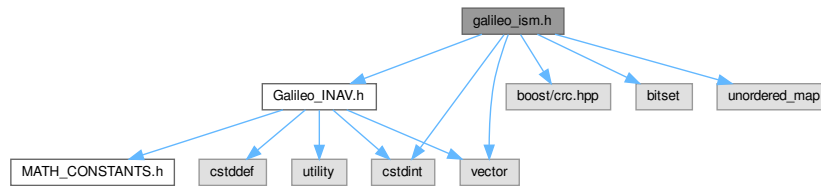
Interface of a Galileo Integrity Support Message.

```

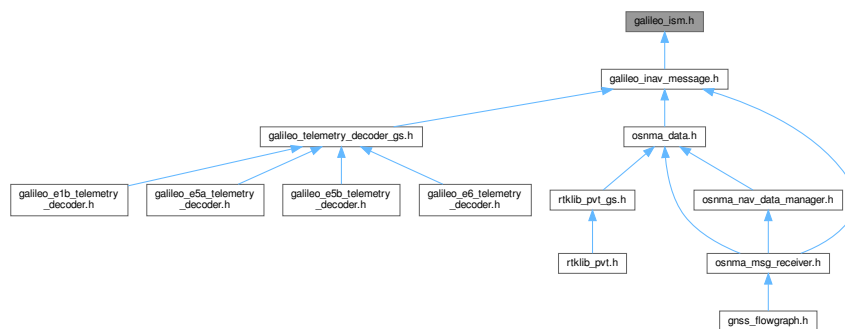
#include "Galileo_INAV.h"
#include <boost/crc.hpp>
#include <bitset>
#include <cstdint>
#include <unordered_map>
#include <vector>

```

Include dependency graph for `galileo_ism.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Galileo\\_ISM](#)

*This class is a storage for the GALILEO Integrity Support Message as described in Galileo ICD paragraph 5.2.*

## 11.723.1 Detailed Description

Interface of a Galileo Integrity Support Message.

### Author

Carles Fernandez, 2024. cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [galileo\\_ism.h](#).

---

## 11.724 galileo\_ism.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file galileo_ism.h
00003  * \brief Interface of a Galileo Integrity Support Message
00004  * \author Carles Fernandez, 2024. cfernandez(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2024 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----

```

```

00015  */
00016
00017
00018 #ifndef GNSS_SDR_GALILEO_ISM_H
00019 #define GNSS_SDR_GALILEO_ISM_H
00020
00021 #include "Galileo_INAV.h"
00022 #include <boost/crc.hpp>
00023 #include <bitset>
00024 #include <stdint>
00025 #include <unordered_map>
00026 #include <vector>
00027
00028 /** \addtogroup Core
00029  * \{ */
00030 /** \addtogroup System_Parameters
00031  * \{ */
00032
00033
00034 /*!
00035  * \brief This class is a storage for the GALILEO Integrity Support Message as described
00036  * in Galileo ICD paragraph 5.2
00037  *
00038  * See https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo\_OS\_SIS\_ICD\_v2.1.pdf
00039  */
00040 class Galileo_ISM
00041 {
00042 public:
00043     /*!
00044      * Default constructor
00045      */
00046     Galileo_ISM() = default;
00047
00048     void set_ism_constellation_id(uint8_t const_id);
00049     void set_ism_service_level_id(uint8_t sl_id);
00050     void set_ism_wn(uint16_t wn_ism);
00051     void set_ism_t0(uint16_t t0);
00052     void set_ism_mask_msb(bool mask_msb);
00053     void set_ism_mask(uint32_t mask);
00054     void set_ism_pconst(uint8_t pconst);
00055     void set_ism_psat(uint8_t psat);
00056     void set_ism_ura(uint8_t ura);
00057     void set_ism_ure(uint8_t ure);
00058     void set_ism_bnom(uint8_t bnom);
00059     void set_ism_Tvalidity(uint8_t tvalidity);
00060
00061     bool check_ism_crc(const std::bitset<GALILEO_DATA_JK_BITS>& bits);
00062
00063     double get_pconst_value() const;
00064     double get_psat_value() const;
00065     float get_ura_m() const;
00066     float get_ure_m() const;
00067     float get_bnom_m() const;
00068     uint32_t get_mask_ISM() const;
00069     uint16_t get_WN_ISM() const;
00070     uint16_t get_t0_ISM() const;
00071     uint16_t get_Tvalidity_hours() const;
00072     bool get_ism_mask_msb() const;
00073     bool ism_parameters_apply(uint32_t prn) const;
00074
00075 private:
00076     uint32_t compute_crc(const std::vector<uint8_t>& data);
00077     boost::crc_optimal<32, 0x814141AB, 0, 0, false, false> d_crc32_ism;
00078
00079     // ICD 2.1 Table 97
00080     std::unordered_map<uint8_t, double> d_ISM_PCONST_MAP = {
00081         {0, 1.0e-8},
00082         {1, 1.0e-7},
00083         {2, 1.0e-6},
00084         {3, 3.0e-6},
00085         {4, 6.0e-6},
00086         {5, 8.0e-6},
00087         {6, 1.0e-5},
00088         {7, 2.0e-5},
00089         {8, 4.0e-5},
00090         {9, 6.0e-5},
00091         {10, 8.0e-5},
00092         {11, 1.0e-4},
00093         {12, 1.25e-4},
00094         {13, 1.5e-4},
00095         {14, 1.75e-4},
00096         {15, 2.0e-4}};
00097
00098     // ICD 2.1 Table 98
00099     std::unordered_map<uint8_t, double> d_ISM_PSAT_MAP = {
00100         {0, 1.0e-7},
00101         {1, 3.0e-7},

```

```
00102         {2, 6.0e-7},
00103         {3, 1.0e-6},
00104         {4, 2.0e-6},
00105         {5, 3.0e-6},
00106         {6, 5.0e-6},
00107         {7, 7.0e-6},
00108         {8, 1.0e-5},
00109         {9, 1.2e-5},
00110         {10, 1.4e-5},
00111         {11, 1.7e-5},
00112         {12, 2.05e-5},
00113         {13, 2.4e-5},
00114         {14, 2.8e-5},
00115         {15, 3.0e-5}};
00116
00117 // ICD 2.1 Table 99
00118 std::unordered_map<uint8_t, float> d_ISM_URA_MAP = {
00119     {0, 0.75},
00120     {1, 1.0},
00121     {2, 1.5},
00122     {3, 2.0},
00123     {4, 2.25},
00124     {5, 2.50},
00125     {6, 2.75},
00126     {7, 3.0},
00127     {8, 3.25},
00128     {9, 3.50},
00129     {10, 3.75},
00130     {11, 4.0},
00131     {12, 4.50},
00132     {13, 5.0},
00133     {14, 5.50},
00134     {15, 6.0}};
00135
00136 // ICD 2.1 Table 100
00137 std::unordered_map<uint8_t, float> d_ISM_URE_MAP = {
00138     {0, 0.25},
00139     {1, 0.50},
00140     {2, 0.75},
00141     {3, 1.00},
00142     {4, 1.25},
00143     {5, 1.50},
00144     {6, 1.75},
00145     {7, 2.0},
00146     {8, 2.25},
00147     {9, 2.50},
00148     {10, 2.75},
00149     {11, 3.0},
00150     {12, 3.25},
00151     {13, 3.50},
00152     {14, 3.75},
00153     {15, 4.00}};
00154
00155 // ICD 2.1 Table 101
00156 std::unordered_map<uint8_t, float> d_ISM_BNOM_MAP = {
00157     {0, 0.0},
00158     {1, 0.10},
00159     {2, 0.20},
00160     {3, 0.30},
00161     {4, 0.40},
00162     {5, 0.50},
00163     {6, 0.60},
00164     {7, 0.75},
00165     {8, 0.85},
00166     {9, 1.0},
00167     {10, 1.20},
00168     {11, 1.40},
00169     {12, 1.60},
00170     {13, 1.80},
00171     {14, 2.0},
00172     {15, 2.4}};
00173
00174 // ICD 2.1 Table 102
00175 std::unordered_map<uint8_t, uint16_t> d_ISM_TVALIDITY_MAP = {
00176     {0, 1},
00177     {1, 2},
00178     {2, 3},
00179     {3, 4},
00180     {4, 6},
00181     {5, 8},
00182     {6, 12},
00183     {7, 18},
00184     {8, 24},
00185     {9, 36},
00186     {10, 48},
00187     {11, 72},
00188     {12, 120},
```

```

00189         {13, 168},
00190         {14, 720},
00191         {15, 1440}};
00192
00193     uint32_t d_ism_crc{};
00194     uint32_t d_ism_mask{};
00195     uint16_t d_ism_wn{};
00196     uint16_t d_ism_t0{};
00197     uint8_t d_ism_constellation_id{};
00198     uint8_t d_ism_service_level_id{};
00199     uint8_t d_ism_pconst{};
00200     uint8_t d_ism_psat{};
00201     uint8_t d_ism_ura{};
00202     uint8_t d_ism_ure{};
00203     uint8_t d_ism_bnom{};
00204     uint8_t d_ism_Tvalidity{};
00205     bool d_ism_mask_msb{};
00206 };
00207
00208 /** \} */
00209 /** \} */
00210 #endif // GNSS_SDR_GALILEO_ISM_H

```

## 11.725 Galileo\_OSNMA.h File Reference

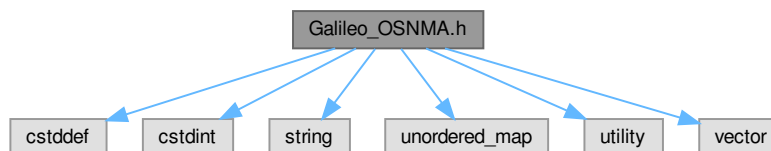
Galileo OSNMA message constants.

```

#include <cstdint>
#include <cstdint>
#include <string>
#include <unordered_map>
#include <utility>
#include <vector>

```

Include dependency graph for Galileo\_OSNMA.h:



### Classes

- class [Mack\\_lookup](#)

### Functions

- const std::string **PEMFILE\_DEFAULT** ("./OSNMA\_PublicKey.pem")
- const std::string **CRTPFILE\_DEFAULT** ("./OSNMA\_PublicKey\_20240115100000\_newPKID\_1.crt")
- const std::string **MERKLEFILE\_DEFAULT** ("./OSNMA\_MerkleTree\_20240115100000\_newPKID\_1.xml")
- const std::string **KROOTFILE\_DEFAULT** ("./OSNMA\_DSM\_KROOT\_NMAHeader.bin")

### Variables

- constexpr size\_t **SIZE\_DSM\_BLOCKS\_BYTES** = 13
- const std::unordered\_map< uint8\_t, std::string > **OSNMA\_TABLE\_1**
- const std::unordered\_map< uint8\_t, std::string > **OSNMA\_TABLE\_2**
- const std::unordered\_map< uint8\_t, std::pair< uint16\_t, uint16\_t > > **OSNMA\_TABLE\_3**
- const std::unordered\_map< uint8\_t, std::string > **OSNMA\_TABLE\_5**
- const std::unordered\_map< std::string, uint16\_t > **OSNMA\_TABLE\_6**

- `const std::unordered_map< uint8_t, std::pair< uint16_t, uint16_t > > OSNMA_TABLE_7`
- `const std::unordered_map< uint8_t, std::string > OSNMA_TABLE_8`
- `const std::unordered_map< uint8_t, uint16_t > OSNMA_TABLE_10`
- `const std::unordered_map< uint8_t, uint8_t > OSNMA_TABLE_11`
- `const std::unordered_map< std::string, uint16_t > OSNMA_TABLE_15`
- `const std::unordered_map< uint8_t, Mack_lookup > OSNMA_TABLE_16`

## 11.725.1 Detailed Description

Galileo OSNMA message constants.

Author

Carles Fernandez, 2023. cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2023 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [Galileo\\_OSNMA.h](#).

---

## 11.726 Galileo\_OSNMA.h

[Go to the documentation of this file.](#)

```
00001  /*!
00002  * \file Galileo_OSNMA.h
00003  * \brief Galileo OSNMA message constants
00004  * \author Carles Fernandez, 2023. cfernandez(at)cttc.es
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2023 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018 #ifndef GNSS_SDR_GALILEO_OSNMA_H
00019 #define GNSS_SDR_GALILEO_OSNMA_H
00020
00021 #include <cstdint>
00022 #include <string>
00023 #include <unordered_map>
00024 #include <utility>
00025 #include <vector>
00026
00027 /** \addtogroup Core
00028  * \{ */
00029 /** \addtogroup System_Parameters
00030  * \{ */
00031
00032 constexpr size_t SIZE_DSM_BLOCKS_BYTES = 13;
00033
00034 // OSNMA User ICD, Issue 1.1, Table 1
00035 const std::unordered_map<uint8_t, std::string> OSNMA_TABLE_1 = {
00036     {0, std::string("Reserved")},
00037     {1, std::string("Test")},
00038     {2, std::string("Operational")},
00039     {3, std::string("Don't use")}; // key: nmas, value: nmas status
00040
00041 // OSNMA User ICD, Issue 1.1, Table 2
00042 const std::unordered_map<uint8_t, std::string> OSNMA_TABLE_2 = {
00043     {0, std::string("Reserved")},
00044     {1, std::string("Nominal")},
00045     {2, std::string("End of Chain (EOC)")},
00046     {3, std::string("Chain Revoked (CREV)")},
00047     {4, std::string("New Public Key (NPK)")},
00048     {5, std::string("Public Key Revoked (PKREV)")},
00049     {6, std::string("New Merkle Tree (NMT)")},
00050     {7, std::string("Alert Message (AM)"); // key: cpks, value: cpks status
00051
00052 // OSNMA User ICD for the Test Phase, Issue 1.0, Table 3
00053 const std::unordered_map<uint8_t, std::pair<uint16_t, uint16_t> OSNMA_TABLE_3 = {
```

```

00056     {0, {0, 0}},
00057     {1, {0, 0}},
00058     {2, {0, 0}},
00059     {3, {0, 0}},
00060     {4, {0, 0}},
00061     {5, {0, 0}},
00062     {6, {0, 0}},
00063     {7, {13, 1352}},
00064     {8, {14, 1456}},
00065     {9, {15, 1560}},
00066     {10, {16, 1664}},
00067     {11, {0, 0}},
00068     {12, {0, 0}},
00069     {13, {0, 0}},
00070     {14, {0, 0}},
00071     {15, {0, 0}}}; // key: nb_dp, value: {num_blocks, l_dp_bits}
00072
00073 const std::unordered_map<uint8_t, std::string> OSNMA_TABLE_5 = {
00074     {0, std::string("Reserved")},
00075     {1, std::string("ECDSA P-256")},
00076     {2, std::string("Reserved")},
00077     {3, std::string("ECDSA P-521")},
00078     {4, std::string("OAM")},
00079     {5, std::string("Reserved")},
00080     {6, std::string("Reserved")},
00081     {7, std::string("Reserved")},
00082     {8, std::string("Reserved")},
00083     {9, std::string("Reserved")},
00084     {10, std::string("Reserved")},
00085     {11, std::string("Reserved")},
00086     {12, std::string("Reserved")},
00087     {13, std::string("Reserved")},
00088     {14, std::string("Reserved")},
00089     {15, std::string("Reserved")}}; // key: nptk, value: message
00090
00091 const std::unordered_map<std::string, uint16_t> OSNMA_TABLE_6 = {
00092     {std::string("ECDSA P-256"), 264},
00093     {std::string("ECDSA P-521"), 536}};
00094
00095 // OSNMA User ICD, Issue 1.1, Table 7
00096 const std::unordered_map<uint8_t, std::pair<uint16_t, uint16_t>> OSNMA_TABLE_7 = {
00097     {0, {0, 0}},
00098     {1, {7, 728}},
00099     {2, {8, 832}},
00100     {3, {9, 936}},
00101     {4, {10, 1040}},
00102     {5, {11, 1144}},
00103     {6, {12, 1248}},
00104     {7, {13, 1352}},
00105     {8, {14, 1456}},
00106     {9, {0, 0}},
00107     {10, {0, 0}},
00108     {11, {0, 0}},
00109     {12, {0, 0}},
00110     {13, {0, 0}},
00111     {14, {0, 0}},
00112     {15, {0, 0}}}; // key: nb_dk, value: {num_blocks, l_dk_bits}
00113
00114 const std::unordered_map<uint8_t, std::string> OSNMA_TABLE_8 = {
00115     {0, std::string("SHA-256")},
00116     {1, std::string("Reserved")},
00117     {2, std::string("SHA3-256")},
00118     {3, std::string("Reserved")}}; // key: hs, value: hash_function
00119
00120 const std::unordered_map<uint8_t, uint16_t> OSNMA_TABLE_10 = {
00121     {0, 96},
00122     {1, 104},
00123     {2, 112},
00124     {3, 120},
00125     {4, 128},
00126     {5, 160},
00127     {6, 192},
00128     {7, 224},
00129     {8, 256},
00130     {9, 0},
00131     {10, 0},
00132     {11, 0},
00133     {12, 0},
00134     {13, 0},
00135     {15, 0},
00136     {15, 0}}; // key: ks, value: lk_bits
00137
00138 const std::unordered_map<uint8_t, uint8_t> OSNMA_TABLE_11 = {
00139     {0, 0},
00140     {1, 0},
00141     {2, 0},
00142     {3, 0},

```

```

00143     {4, 0},
00144     {5, 20},
00145     {6, 24},
00146     {7, 28},
00147     {8, 32},
00148     {9, 40},
00149     {10, 0},
00150     {11, 0},
00151     {12, 0},
00152     {13, 0},
00153     {14, 0},
00154     {15, 0},
00155 };
00156
00157 const std::unordered_map<std::string, uint16_t> OSNMA_TABLE_15 = {
00158     {std::string("ECDSA P-256"), 512},
00159     {std::string("ECDSA P-521"), 1056}}; // key: ECDSA Curve and hash function, value: {l_ds_bits}
00160
00161 const std::string PEMFILE_DEFAULT("./OSNMA_PublicKey.pem");
00162 const std::string CRTFILE_DEFAULT("./OSNMA_PublicKey_20240115100000_newPKID_1.crt");
00163 const std::string MERKLEFILE_DEFAULT("./OSNMA_MerkleTree_20240115100000_newPKID_1.xml");
00164 const std::string KROOTFILE_DEFAULT("./OSNMA_DSM_KROOT_NMAHeader.bin");
00165
00166 class Mack_lookup
00167 {
00168 public:
00169     Mack_lookup() = default;
00170     Mack_lookup(uint8_t msg_,
00171         uint8_t nt_,
00172         const std::vector<std::string>& s1_,
00173         const std::vector<std::string>& s2_) : msg(msg_),
00174         nt(nt_),
00175         sequence1(s1_),
00176         sequence2(s2_) {};
00177     uint8_t msg{};
00178     uint8_t nt{};
00179     std::vector<std::string> sequence1;
00180     std::vector<std::string> sequence2;
00181 };
00182
00183 const std::unordered_map<uint8_t, Mack_lookup> OSNMA_TABLE_16 = {
00184     {27, {2, 6, {"00S", "00E", "00E", "12S", "00E"}, {"00S", "00E", "00E", "04S", "12S",
00185         "00E"}}},
00186     {28, {2, 10, {"00S", "00E", "00E", "00E", "00S", "00E", "00E", "12S", "00E", "00E"}, {"00S",
00187         "00E", "00E", "00S", "00E", "00E", "04S", "12S", "00E", "00E"}}},
00188     {31, {2, 5, {"00S", "00E", "00E", "12S", "00E"}, {"00S", "00E", "00E", "12S", "04S"}}},
00189     {33, {2, 6, {"00S", "00E", "04S", "00E", "12S", "00E"}, {"00S", "00E", "00E", "12S", "00E",
00190         "12E"}}},
00191     {34, {2, 6, {"00S", "FLX", "04S", "FLX", "12S", "00E"}, {"00S", "FLX", "00E", "12S", "00E",
00192         "12E"}}},
00193     {35, {2, 6, {"00S", "FLX", "04S", "FLX", "12S", "FLX"}, {"00S", "FLX", "FLX", "12S", "FLX",
00194         "FLX"}}},
00195     {36, {2, 5, {"00S", "FLX", "04S", "FLX", "12S"}, {"00S", "FLX", "00E", "12S", "12E"}}},
00196     {37, {2, 5, {"00S", "00E", "04S", "00E", "12S"}, {"00S", "00E", "00E", "12S", "12E"}}},
00197     {38, {2, 5, {"00S", "FLX", "04S", "FLX", "12S"}, {"00S", "FLX", "FLX", "12S", "FLX"}}},
00198     {39, {2, 4, {"00S", "FLX", "04S", "FLX"}, {"00S", "FLX", "00E", "12S"}}},
00199     {40, {2, 4, {"00S", "00E", "04S", "12S"}, {"00S", "00E", "00E", "12E"}}},
00200     {41, {2, 4, {"00S", "FLX", "04S", "FLX"}, {"00S", "FLX", "FLX", "12S"}}};
00201
00202 /** */
00203 /** */
00204 #endif // GNSS_SDR_GALILEO_OSNMA_H

```

## 11.727 galileo\_reduced\_ced.h File Reference

Galileo Reduced Clock and Ephemeris Data storage class.

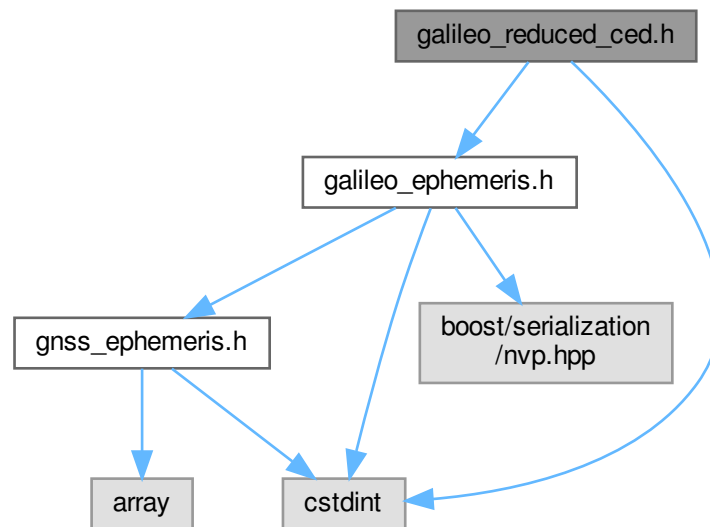
```

#include "galileo_ephemeris.h"
#include <cstdint>

```



Include dependency graph for galileo\_reduced\_ced.h:



## Classes

- class [Galileo\\_Reduced\\_CED](#)

*This class manages the Galileo Reduced Clock and Ephemeris Data.*

### 11.727.1 Detailed Description

Galileo Reduced Clock and Ephemeris Data storage class.

#### Author

Carles Fernandez, 2021. cfernandez(at)cttc.cat

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [galileo\\_reduced\\_ced.h](#).

---

## 11.728 galileo\_reduced\_ced.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002   * \file galileo_reduced_ced.h
00003   * \brief Galileo Reduced Clock and Ephemeris Data storage class
00004   * \author Carles Fernandez, 2021. cfernandez(at)cttc.cat
00005   *
00006   * -----
00007   *
00008   * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009   * This file is part of GNSS-SDR.
00010   *
00011   * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)
00012   * SPDX-License-Identifier: GPL-3.0-or-later
00013   *
00014   * -----
00015   */
00016
00017
00018  #ifndef GNSS_SDR_GALILEO_REDUCED_CED_H

```

```

00019 #define GNSS_SDR_GALILEO_REDUCED_CED_H
00020
00021 #include "galileo_ephemeris.h"
00022 #include <stdint>
00023
00024 /** \addtogroup Core
00025  * \{ */
00026 /** \addtogroup System_Parameters
00027  * \{ */
00028
00029
00030 /*!
00031  * \brief This class manages the Galileo Reduced Clock and Ephemeris Data
00032  */
00033 class Galileo_Reduced_CED
00034 {
00035 public:
00036     /*!
00037      * Default constructor
00038      */
00039     Galileo_Reduced_CED() = default;
00040
00041     /*!
00042      * Convert to Galileo_Ephemeris
00043      */
00044     Galileo_Ephemeris compute_eph() const;
00045
00046     uint32_t PRN{};          //!< Satellite ID
00047     int32_t TOTRedCED{};     //!< Start time of transmission of the Reduced CED word in GST
00048     int32_t IODnav{};        //!< Issue of Data
00049     double DeltaAred{};      //!< Difference between the Reduced CED semi-major axis and the nominal
00050                             semi-major axis [meters]
00051     double exred{};          //!< Reduced CED eccentricity vector component x
00052     double eyred{};          //!< Reduced CED eccentricity vector component y
00053     double Deltai0red{};     //!< Difference between the Reduced CED inclination angle at reference time
00054                             and the nominal inclination [rad]
00055     double Omega0red{};      //!< Reduced CED longitude of ascending node at weekly epoch [rad]
00056     double lambda0red{};     //!< Reduced CED mean argument of latitude [rad]
00057     double af0red{};         //!< Reduced CED satellite clock bias correction coefficient [seconds]
00058     double aflred{};         //!< Reduced CED satellite clock drift correction coefficient
00059                             [seconds/seconds]
00060 };
00061
00062 /** \} */
00063 /** \} */
00064 #endif // GNSS_SDR_GALILEO_REDUCED_CED_H

```

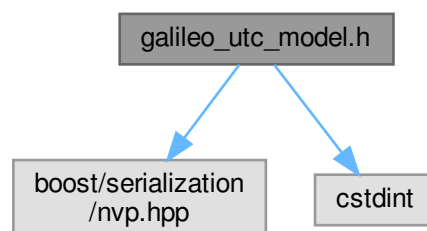
## 11.729 galileo\_utc\_model.h File Reference

Interface of a Galileo UTC MODEL storage.

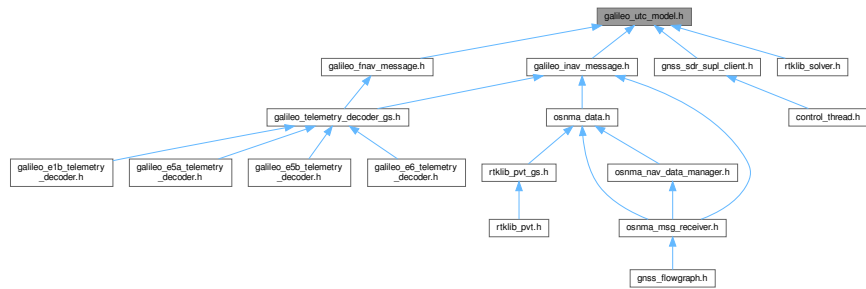
```
#include <boost/serialization/nvp.hpp>
```

```
#include <stdint>
```

Include dependency graph for galileo\_utc\_model.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Galileo\\_Utc\\_Model](#)

*This class is a storage for the GALILEO UTC MODEL data as described in Galileo ICD [https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo\\_OS\\_SIS\\_ICD\\_v2.0.pdf](https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OS_SIS_ICD_v2.0.pdf) paragraph 5.1.7.*

## 11.729.1 Detailed Description

Interface of a Galileo UTC MODEL storage.

### Author

Javier Arribas, 2013. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [galileo\\_utc\\_model.h](#).

---

## 11.730 galileo\_utc\_model.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file galileo_utc_model.h
00003  * \brief Interface of a Galileo UTC MODEL storage
00004  * \author Javier Arribas, 2013. jarribas(at)cttc.es
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018
00019 #ifndef GNSS_SDR_GALILEO_UTC_MODEL_H
00020 #define GNSS_SDR_GALILEO_UTC_MODEL_H
00021
00022 #include <boost/serialization/nvp.hpp>
00023 #include <cstdint>
00024
00025 /** \addtogroup Core
00026  * \{ */
00027 /** \addtogroup System_Parameters
00028  * \{ */
00029
00030
00031 /*!
00032  * \brief This class is a storage for the GALILEO UTC MODEL data as described in Galileo ICD
00033  * https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo\_OS\_SIS\_ICD\_v2.0.pdf
```

```

00034  * paragraph 5.1.7
00035  */
00036  class Galileo_Utc_Model
00037  {
00038  public:
00039      /*!
00040       * Default constructor
00041       */
00042      Galileo_Utc_Model() = default;
00043
00044      // double TOW;
00045      double GST_to_UTC_time(double t_e, int32_t WN) const; //!< GST-UTC Conversion Algorithm and
Parameters
00046
00047      // Word type 6: GST-UTC conversion parameters
00048      double A0{};
00049      double A1{};
00050      int32_t Delta_tLS{};
00051      int32_t tot{}; //!< UTC data reference Time of Week [s]
00052      int32_t WNot{}; //!< UTC data reference Week number [week]
00053      int32_t WN_LSF{};
00054      int32_t DN{};
00055      int32_t Delta_tLSF{};
00056
00057      // GPS to Galileo GST conversion parameters
00058      double A_0G{};
00059      double A_1G{};
00060      int32_t t_0G{};
00061      int32_t WN_0G{};
00062
00063      bool flag_utc_model{};
00064
00065      template <class Archive>
00066
00067      /*!
00068       * \brief Serialize is a boost standard method to be called by the boost XML serialization.
00069       * Here is used to save the UTC data on disk file.
00070       */
00071      inline void serialize(Archive& archive, const unsigned int version)
00072      {
00073          if (version)
00074          {
00075              };
00076          archive& BOOST_SERIALIZATION_NVP(A0);
00077          archive& BOOST_SERIALIZATION_NVP(A1);
00078          archive& BOOST_SERIALIZATION_NVP(Delta_tLS);
00079          archive& BOOST_SERIALIZATION_NVP(tot);
00080          archive& BOOST_SERIALIZATION_NVP(WNot);
00081          archive& BOOST_SERIALIZATION_NVP(WN_LSF);
00082          archive& BOOST_SERIALIZATION_NVP(DN);
00083          archive& BOOST_SERIALIZATION_NVP(Delta_tLSF);
00084          archive& BOOST_SERIALIZATION_NVP(flag_utc_model);
00085      }
00086  };
00087
00088
00089  /** \} */
00090  /** \} */
00091  #endif // GNSS_SDR_GALILEO_UTC_MODEL_H

```

## 11.731 glonass\_gnav\_almanac.h File Reference

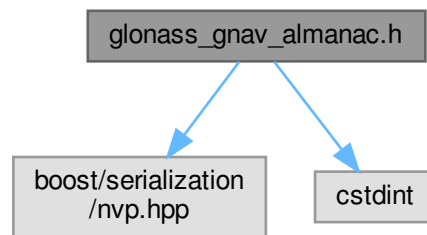
Interface of a GLONASS GNAV ALMANAC storage.

```

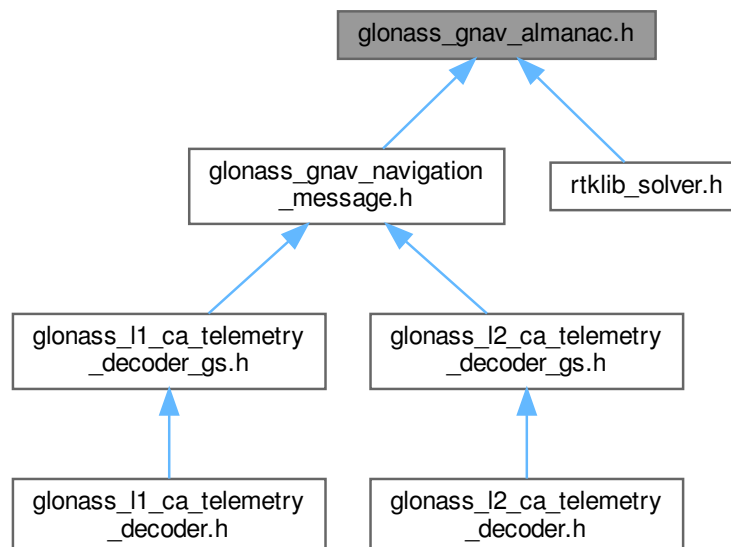
#include <boost/serialization/nvp.hpp>
#include <cstdint>

```

Include dependency graph for glonass\_gnav\_almanac.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Glonass\\_Gnav\\_Almanac](#)

*This class is a storage for the GLONASS SV ALMANAC data as described GLONASS ICD (Edition 5.1)*

### 11.731.1 Detailed Description

Interface of a GLONASS GNAV ALMANAC storage.

#### Note

Code added as part of GSoC 2017 program

## Author

Damian Miralles, 2017. dmiralles2009(at)gmail.com

## See also

[GLONASS ICD](#)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [glonass\\_gnav\\_almanac.h](#).

---

## 11.732 glonass\_gnav\_almanac.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file glonass_gnav_almanac.h
00003  * \brief Interface of a GLONASS GNAV ALMANAC storage
00004  * \note Code added as part of GSOC 2017 program
00005  * \author Damian Miralles, 2017. dmiralles2009(at)gmail.com
00006  * \see <a
00007  href="http://russianspacesystems.ru/wp-content/uploads/2016/08/ICD_GLONASS_eng_v5.1.pdf">GLONASS
00008  ICD</a>
00009  *
00010  * -----
00011  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00012  * This file is part of GNSS-SDR.
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019
00020 #ifndef GNSS_SDR_GLONASS_ALMANAC_H
00021 #define GNSS_SDR_GLONASS_ALMANAC_H
00022
00023 #include <boost/serialization/nvp.hpp>
00024 #include <cstdint>
00025
00026 /** \addtogroup Core
00027  * \{ */
00028 /** \addtogroup System_Parameters
00029  * \{ */
00030
00031
00032  /*!
00033  * \brief This class is a storage for the GLONASS SV ALMANAC data as described GLONASS ICD (Edition
00034  5.1)
00035  * \note Code added as part of GSOC 2017 program
00036  * \see <a
00037  href="http://russianspacesystems.ru/wp-content/uploads/2016/08/ICD_GLONASS_eng_v5.1.pdf">GLONASS
00038  ICD</a>
00039  */
00040  class Glonass_Gnav_Almanac
00041  {
00042  public:
00043  /*!
00044  * Default constructor
00045  */
00046  Glonass_Gnav_Almanac() = default;
00047
00048  double d_n_A{};          //!< Conventional number of satellite within GLONASS space segment
00049  [dimensionless]
00050  double d_H_n_A{};        //!< Carrier frequency number of navigation RF signal transmitted by
00051  d_n_A satellite as table 4.10 (0-31) [dimensionless]
00052  double d_lambda_n_A{};    //!< Longitude of the first (within the d_n_A day) ascending node of
00053  d_n_A [radians]
00054  double d_t_lambda_n_A{};  //!< Time of first ascending node passage [s]
00055  double d_Delta_i_n_A{};   //!< Correction of the mean value of inclination of d_n_A satellite
00056  at instant t_lambda_n_A [radians]
00057  double d_Delta_T_n_A{};   //!< Correction to the mean value of Draconian period of d_n_A
00058  satellite at instant t_lambda_n_A [s / orbital period]
00059  double d_Delta_T_n_A_dot{}; //!< Rate of change of Draconian period of d_n_A satellite at instant
00060  t_lambda_n_A [s / orbital period^2]
00061  double d_epsilon_n_A{};   //!< Eccentricity of d_n_A satellite at instant t_lambda_n_A
00062  [dimensionless]
00063  double d_omega_n_A{};     //!< Argument of perigee of d_n_A satellite at instant t_lambda_n_A
00064  [radians]

```

```

00054     double d_M_n_A{};           //!< Type of satellite n_A [dimensionless]
00055     double d_KP{};              //!< Notification on forthcoming leap second correction of UTC
                                [dimensionless]
00056     double d_tau_n_A{};         //!< Coarse value of d_n_A satellite time correction to GLONASS time
                                at instant t_lambdan_A[s]
00057     bool d_C_n{};               //!< Generalized "unhealthy flag" of n_A satellite at instant of
                                almanac upload [dimensionless]
00058     bool d_l_n{};              //!< Health flag for nth satellite; ln = 0 indicates the n-th
                                satellite is helthy, ln = 1 indicates malfunction of this nth satellite [dimensionless]
00059
00060     // Satellite Identification Information
00061     int32_t i_satellite_freq_channel{};  //!< SV Frequency Channel Number
00062     uint32_t PRN{};                //!< SV PRN Number, equivalent to slot number for
                                compatibility with GPS
00063     uint32_t i_satellite_slot_number{};  //!< SV Slot Number
00064
00065     template <class Archive>
00066     /*!
00067      * \brief Serialize is a boost standard method to be called by the boost XML
00068      * serialization. Here is used to save the almanac data on disk file.
00069      */
00070     void serialize(Archive& archive, const uint32_t version)
00071     {
00072         if (version)
00073         {
00074             };
00075
00076             archive& BOOST_SERIALIZATION_NVP(i_satellite_freq_channel);
00077             archive& BOOST_SERIALIZATION_NVP(PRN);
00078             archive& BOOST_SERIALIZATION_NVP(i_satellite_slot_number);
00079             archive& BOOST_SERIALIZATION_NVP(d_n_A);
00080             archive& BOOST_SERIALIZATION_NVP(d_H_n_A);
00081             archive& BOOST_SERIALIZATION_NVP(d_lambda_n_A);
00082             archive& BOOST_SERIALIZATION_NVP(d_t_lambda_n_A);
00083             archive& BOOST_SERIALIZATION_NVP(d_Delta_i_n_A);
00084             archive& BOOST_SERIALIZATION_NVP(d_Delta_T_n_A);
00085             archive& BOOST_SERIALIZATION_NVP(d_Delta_T_n_A_dot);
00086             archive& BOOST_SERIALIZATION_NVP(d_epsilon_n_A);
00087             archive& BOOST_SERIALIZATION_NVP(d_omega_n_A);
00088             archive& BOOST_SERIALIZATION_NVP(d_M_n_A);
00089             archive& BOOST_SERIALIZATION_NVP(d_KP);
00090             archive& BOOST_SERIALIZATION_NVP(d_tau_n_A);
00091             archive& BOOST_SERIALIZATION_NVP(d_C_n);
00092             archive& BOOST_SERIALIZATION_NVP(d_l_n);
00093         }
00094     };
00095
00096
00097     /** \} */
00098     /** \} */
00099 #endif // GNSS_SDR_GLOASS_ALMANAC_H

```

## 11.733 glonass\_gnav\_ephemeris.h File Reference

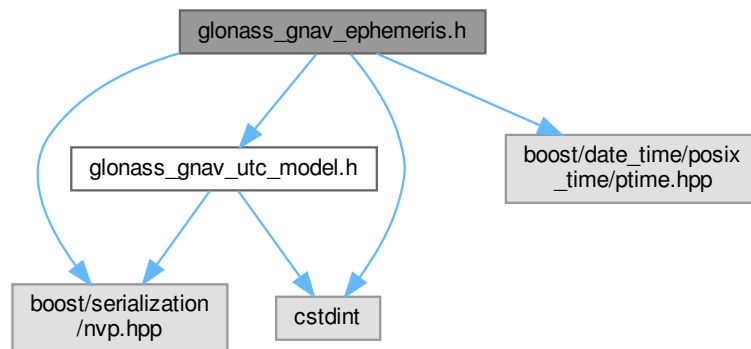
Interface of a GLONASS EPHEMERIS storage.

```

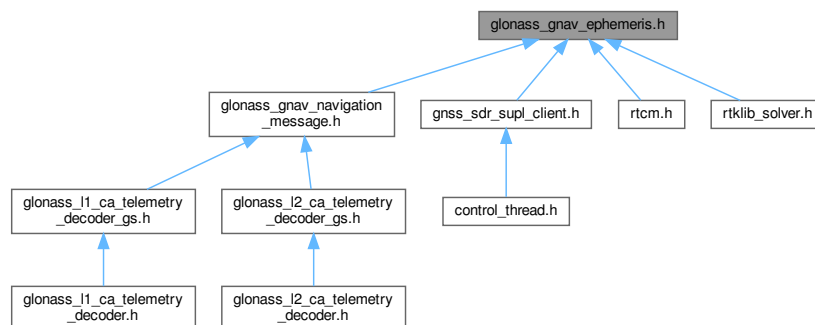
#include "glonass_gnav_utc_model.h"
#include <boost/date_time/posix_time/ptime.hpp>
#include <boost/serialization/nvp.hpp>
#include <cstdint>

```

Include dependency graph for `glonass_gnav_ephemeris.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Glonass\\_Gnav\\_Ephemeris](#)

*This class is a storage and orbital model functions for the GLONASS SV ephemeris data as described in GLONASS ICD (Edition 5.1)*

### 11.733.1 Detailed Description

Interface of a GLONASS EPHEMERIS storage.

#### Note

Code added as part of GSoC 2017 program

#### Author

Damian Miralles, 2017. [dmiralles2009\(at\)gmail.com](mailto:dmiralles2009(at)gmail.com)

#### See also

[GLONASS ICD](#)

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR. Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later Definition in file [glonass\\_gnav\\_ephemeris.h](#).



## 11.734 glonass\_gnav\_ephemeris.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file glonass_gnav_ephemeris.h
00003  * \brief Interface of a GLONASS EPHEMERIS storage
00004  * \note Code added as part of GSoC 2017 program
00005  * \author Damian Miralles, 2017. dmiralles2009(at)gmail.com
00006  * \see <a
href="http://russianspacesystems.ru/wp-content/uploads/2016/08/ICD_GLONASS_eng_v5.1.pdf">GLONASS
ICD</a>
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019
00020 #ifndef GNSS_SDR_GLONASS_GNAV_EPHEMERIS_H
00021 #define GNSS_SDR_GLONASS_GNAV_EPHEMERIS_H
00022
00023
00024 #include "glonass_gnav_utc_model.h"
00025 #include <boost/date_time/posix_time/ptime.hpp> // for ptime
00026 #include <boost/serialization/nvp.hpp>
00027 #include <cstdint>
00028
00029 /** \addtogroup Core
00030 * \{ */
00031 /** \addtogroup System_Parameters
00032 * \{ */
00033
00034
00035 /*!
00036 * \brief This class is a storage and orbital model functions for the GLONASS SV ephemeris data as
described in GLONASS ICD (Edition 5.1)
00037 * \note Code added as part of GSoC 2017 program
00038 * \see <a
href="http://russianspacesystems.ru/wp-content/uploads/2016/08/ICD_GLONASS_eng_v5.1.pdf">GLONASS
ICD</a>
00039 */
00040 class Glonass_Gnav_Ephemeris
00041 {
00042 public:
00043     /*!
00044     * Default constructor
00045     */
00046     Glonass_Gnav_Ephemeris() = default;
00047
00048     double d_m{}; //!< String number within frame [dimensionless]
00049     double d_t_k{}; //!< GLONASS Time (UTC(SU) + 3 h) referenced to the beginning of the
frame within the current day [s]
00050     double d_t_b{}; //!< Reference ephemeris relative time in GLONASS Time (UTC(SU) + 3 h).
Index of a time interval within current day according to UTC(SU) + 03 hours 00 min. [s]
00051     double d_M{}; //!< Type of satellite transmitting navigation signal [dimensionless]
00052     double d_gamma_n{}; //!< Relative deviation of predicted carrier frequency value of n-
satellite from nominal value at the instant tb [dimensionless]
00053     double d_tau_n{}; //!< Correction to the nth satellite time (tn) relative to GLONASS time
(te),
00054     double d_Xn{}; //!< Earth-fixed coordinate x of the satellite in PZ-90.02 coordinate
system [km].
00055     double d_Yn{}; //!< Earth-fixed coordinate y of the satellite in PZ-90.02 coordinate
system [km]
00056     double d_Zn{}; //!< Earth-fixed coordinate z of the satellite in PZ-90.02 coordinate
system [km]
00057     double d_VXn{}; //!< Earth-fixed velocity coordinate x of the satellite in PZ-90.02
coordinate system [km/s]
00058     double d_VYn{}; //!< Earth-fixed velocity coordinate y of the satellite in PZ-90.02
coordinate system [km/s]
00059     double d_VZn{}; //!< Earth-fixed velocity coordinate z of the satellite in PZ-90.02
coordinate system [km/s]
00060     double d_AXn{}; //!< Earth-fixed acceleration coordinate x of the satellite in PZ-90.02
coordinate system [km/s^2]
00061     double d_AYn{}; //!< Earth-fixed acceleration coordinate y of the satellite in PZ-90.02
coordinate system [km/s^2]
00062     double d_AZn{}; //!< Earth-fixed acceleration coordinate z of the satellite in PZ-90.02
coordinate system [km/s^2]
00063     double d_B_n{}; //!< Health flag [dimensionless]
00064     double d_P{}; //!< Technological parameter of control segment, indication the satellite
operation mode in respect of time parameters [dimensionless]

```

```

00065     double d_N_T{};           //!< Current date, calendar number of day within four-year interval
                                starting from the 1-st of January in a leap year [days]
00066     double d_F_T{};           //!< Parameter that provides the predicted satellite user range accuracy
                                at time tb [dimensionless]
00067     double d_n{};             //!< Index of the satellite transmitting given navigation signal. It
                                corresponds to a slot number within GLONASS constellation
00068     double d_Delta_tau_n{};    //!< Time difference between navigation RF signal transmitted in L2 sub-
                                band and aviation RF signal transmitted in L1 sub-band by nth satellite. [dimensionless]
00069     double d_E_n{};           //!< Characterises "age" of a current information [days]
00070     double d_P_1{};           //!< Flag of the immediate data updating [minutes]
00071     bool d_P_2{};             //!< Flag of oddness ("1") or evenness ("0") of the value of (tb)
                                [dimensionless]
00072     bool d_P_3{};             //!< Flag indicating a number of satellites for which almanac is
                                transmitted within given frame: "1" corresponds to 5 satellites and "0" corresponds to 4 satellites
                                [dimensionless]
00073     bool d_P_4{};             //!< Flag to show that ephemeris parameters are present. "1" indicates
                                that updated ephemeris or frequency/time parameters have been uploaded by the control segment
                                [dimensionless]
00074     bool d_l3rd_n{};          //!< Health flag for nth satellite; ln = 0 indicates the n-th satellite
                                is healthy, ln = 1 indicates malfunction of this nth satellite [dimensionless]
00075     bool d_l5th_n{};          //!< Health flag for nth satellite; ln = 0 indicates the n-th satellite
                                is healthy, ln = 1 indicates malfunction of this nth satellite [dimensionless]
00076
00077     // Immediate deliverables of ephemeris information
00078     // Satellite Identification Information
00079     int32_t i_satellite_freq_channel{}; //!< SV Frequency Channel Number
00080     uint32_t PRN{};             //!< SV PRN Number, equivalent to slot number for
                                compatibility with GPS
00081     uint32_t i_satellite_slot_number{}; //!< SV Slot Number
00082     double d_ylr = 1972.0;      //!< Current year
00083     double d_satClkDrift{};      //!< GLONASS clock error
00084     double d_dtr{};             //!< relativistic clock correction term
00085     double d_iode{};            //!< Issue of data, ephemeris (Bit 0-6 of tb)
00086     double d_tau_c{};           //!< GLONASST 2 UTC correction (todo) may be eliminated
00087     double d_TOW{};             //!< GLONASST IN GPST seconds of week
00088     int32_t d_WN{};             //!< GLONASST IN GPST week number of the start of frame
00089     double d_tod{};             //!< Time of Day since ephemeris where decoded
00090
00091     /*!
00092     * \brief Sets (\a d_satClkDrift) and returns the clock drift in seconds according to the User
                                Algorithm for SV Clock Correction
00093     */
00094     double sv_clock_drift(double transmitTime, double timeCorrUTC);
00095
00096     /*!
00097     * \brief Computes the GLONASS System Time and returns a boost::posix_time::ptime object
00098     * \param offset_time Is the start of day offset to compute the time
00099     */
00100     boost::posix_time::ptime compute_GLONASS_time(double offset_time) const;
00101
00102     /*!
00103     * \brief Converts from GLONASST to UTC
00104     * \details The function simply adjust for the 6 hrs offset between GLONASST and UTC
00105     * \param[in] offset_time Is the start of day offset
00106     * \param[in] glot2utc_corr Correction from GLONASST to UTC
00107     * \returns UTC time as a boost::posix_time::ptime object
00108     */
00109     boost::posix_time::ptime glot_to_utc(double offset_time, double glot2utc_corr) const;
00110
00111     /*!
00112     * \brief Converts from GLONASST to GPST
00113     * \details Converts from GLONASST to GPST in time of week (TOW) and week number (WN) format
00114     * \param[in] tod_offset Is the start of day offset
00115     * \param[in] glot2utc_corr Correction from GLONASST to UTC
00116     * \param[in] glot2gpst_corr Correction from GLONASST to GPST
00117     * \param[out] WN Week Number, not in mod(1024) format
00118     * \param[out] TOW Time of Week in seconds of week
00119     */
00120     void glot_to_gpst(double tod_offset, double glot2utc_corr, double glot2gpst_corr, int32_t* WN,
                                double* TOW) const;
00121
00122     template <class Archive>
00123
00124     /*!
00125     * \brief Serialize is a boost standard method to be called by the boost XML
00126     * serialization. Here is used to save the ephemeris data on disk file.
00127     */
00128     void serialize(Archive& archive, const uint32_t version)
00129     {
00130         if (version)
00131         {
00132             };
00133
00134         archive< BOOST_SERIALIZATION_NVP(i_satellite_freq_channel); //!< SV PRN frequency channel
                                number
00135         archive< BOOST_SERIALIZATION_NVP(PRN);
00136         archive< BOOST_SERIALIZATION_NVP(i_satellite_slot_number);

```

```

00137         archive& BOOST_SERIALIZATION_NVP(d_m);           //!< String number within frame
[dimensionless]
00138         archive& BOOST_SERIALIZATION_NVP(d_t_k);         //!< Time referenced to the beginning of the
frame within the current day [hours, minutes, seconds]
00139         archive& BOOST_SERIALIZATION_NVP(d_t_b);         //!< Index of a time interval within current
day according to UTC(SU) + 03 hours 00 min. [minutes]
00140         archive& BOOST_SERIALIZATION_NVP(d_M);           //!< Type of satellite transmitting
navigation signal [dimensionless]
00141         archive& BOOST_SERIALIZATION_NVP(d_gamma_n);     //!< Relative deviation of predicted carrier
frequency value of n- satellite from nominal value at the instant tb [dimensionless]
00142         archive& BOOST_SERIALIZATION_NVP(d_tau_n);       //!< Correction to the nth satellite time
(tn) relative to GLONASS time (te)
00143         archive& BOOST_SERIALIZATION_NVP(d_Xn);          //!< Earth-fixed coordinate x of the
satellite in PZ-90.02 coordinate system [km].
00144         archive& BOOST_SERIALIZATION_NVP(d_Yn);          //!< Earth-fixed coordinate y of the
satellite in PZ-90.02 coordinate system [km]
00145         archive& BOOST_SERIALIZATION_NVP(d_Zn);          //!< Earth-fixed coordinate z of the
satellite in PZ-90.02 coordinate system [km]
00146         archive& BOOST_SERIALIZATION_NVP(d_VXn);        //!< Earth-fixed velocity coordinate x of
the satellite in PZ-90.02 coordinate system [km/s]
00147         archive& BOOST_SERIALIZATION_NVP(d_VYn);        //!< Earth-fixed velocity coordinate y of
the satellite in PZ-90.02 coordinate system [km/s]
00148         archive& BOOST_SERIALIZATION_NVP(d_VZn);        //!< Earth-fixed velocity coordinate z of
the satellite in PZ-90.02 coordinate system [km/s]
00149         archive& BOOST_SERIALIZATION_NVP(d_AXn);        //!< Earth-fixed acceleration coordinate x
of the satellite in PZ-90.02 coordinate system [km/s^2]
00150         archive& BOOST_SERIALIZATION_NVP(d_AYn);        //!< Earth-fixed acceleration coordinate y
of the satellite in PZ-90.02 coordinate system [km/s^2]
00151         archive& BOOST_SERIALIZATION_NVP(d_AZn);        //!< Earth-fixed acceleration coordinate z
of the satellite in PZ-90.02 coordinate system [km/s^2]
00152         archive& BOOST_SERIALIZATION_NVP(d_B_n);        //!< Health flag [dimensionless]
00153         archive& BOOST_SERIALIZATION_NVP(d_P);          //!< Technological parameter of control
segment, indication the satellite operation mode in respect of time parameters [dimensionless]
00154         archive& BOOST_SERIALIZATION_NVP(d_N_T);        //!< Current date, calendar number of day
within four-year interval starting from the 1-st of January in a leap year [days]
00155         archive& BOOST_SERIALIZATION_NVP(d_F_T);        //!< Parameter that provides the predicted
satellite user range accuracy at time tb [dimensionless]
00156         archive& BOOST_SERIALIZATION_NVP(d_n);          //!< Index of the satellite transmitting
given navigation signal. It corresponds to a slot number within GLONASS constellation
00157         archive& BOOST_SERIALIZATION_NVP(d_Delta_tau_n); //!< Time difference between navigation RF
signal transmitted in L2 sub- band and aviation RF signal transmitted in L1 sub-band by nth satellite.
[dimensionless]
00158         archive& BOOST_SERIALIZATION_NVP(d_E_n);        //!< Characterises "age" of a current
information [days]
00159         archive& BOOST_SERIALIZATION_NVP(d_P_1);        //!< Flag of the immediate data updating.
00160         archive& BOOST_SERIALIZATION_NVP(d_P_2);        //!< Flag of oddness ("1") or evenness ("0")
of the value of (tb) [dimensionless]
00161         archive& BOOST_SERIALIZATION_NVP(d_P_3);        //!< Flag indicating a number of satellites
for which almanac is transmitted within given frame: "1" corresponds to 5 satellites and "0"
corresponds to 4 satellites [dimensionless]
00162         archive& BOOST_SERIALIZATION_NVP(d_P_4);        //!< Flag to show that ephemeris parameters
are present. "1" indicates that updated ephemeris or frequency/time parameters have been uploaded by
the control segment [dimensionless]
00163         archive& BOOST_SERIALIZATION_NVP(d_l3rd_n);     //!< Health flag for nth satellite; ln = 0
indicates the n-th satellite is healthy, ln = 1 indicates malfunction of this nth satellite
[dimensionless]
00164         archive& BOOST_SERIALIZATION_NVP(d_l5th_n);     //!< Health flag for nth satellite; ln = 0
indicates the n-th satellite is healthy, ln = 1 indicates malfunction of this nth satellite
[dimensionless]
00165     }
00166
00167 private:
00168     /*
00169     * Accounts for the beginning or end of week crossover
00170     *
00171     * \param[in] - time in seconds
00172     * \param[out] - corrected time, in seconds
00173     */
00174     double check_t(double time);
00175 };
00176
00177
00178 /** @} */
00179 /** @} */
00180 #endif // GNSS_SDR_GLO_NASS_GNAV_EPHEMERIS_H

```

## 11.735 glonass\_gnav\_navigation\_message.h File Reference

Interface of a GLONASS GNAV Data message decoder as described in GLONASS ICD (Edition 5.1)

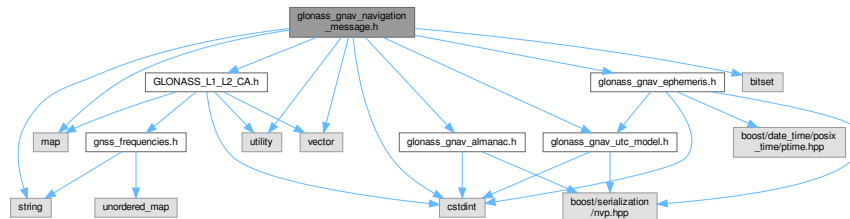
```

#include "GLONASS_L1_L2_CA.h"
#include "glonass_gnav_almanac.h"
#include "glonass_gnav_ephemeris.h"

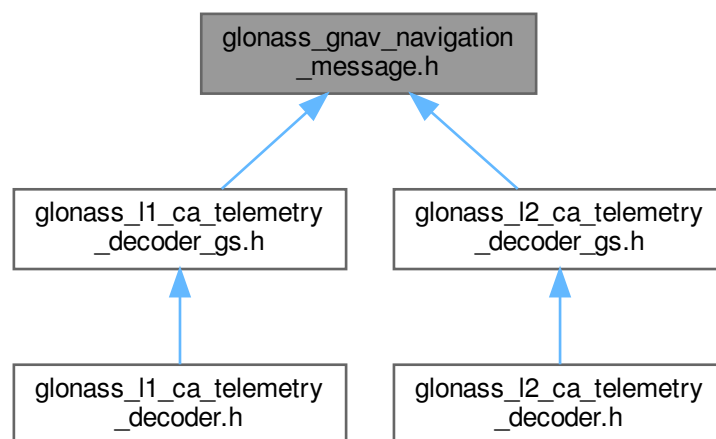
```

```
#include "glonass_gnav_utc_model.h"
#include <bitset>
#include <stdint>
#include <map>
#include <string>
#include <utility>
#include <vector>
```

Include dependency graph for glonass\_gnav\_navigation\_message.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Glonass\\_Gnav\\_Navigation\\_Message](#)

*This class decodes a GLONASS GNAV Data message as described in GLONASS ICD (Edition 5.1)*

## 11.735.1 Detailed Description

Interface of a GLONASS GNAV Data message decoder as described in GLONASS ICD (Edition 5.1)

### Note

Code added as part of GSoC 2017 program

## Author

Damian Miralles, 2017. dmiralles2009(at)gmail.com

## See also

GLONASS ICD

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [glonass\\_gnav\\_navigation\\_message.h](#).

---

## 11.736 glonass\_gnav\_navigation\_message.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file glonass_gnav_navigation_message.h
00003  * \brief Interface of a GLONASS GNAV Data message decoder as described in GLONASS ICD (Edition 5.1)
00004  * \note Code added as part of GSoC 2017 program
00005  * \author Damian Miralles, 2017. dmiralles2009(at)gmail.com
00006  * \see <a
00007  href="http://russianspacesystems.ru/wp-content/uploads/2016/08/ICD_GLONASS_eng_v5.1.pdf">GLONASS
00008  ICD</a>
00009  *
00010  * -----
00011  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00012  * This file is part of GNSS-SDR.
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019 #ifndef GNSS_SDR_GLO_NASS_GNAV_NAVIGATION_MESSAGE_H
00020 #define GNSS_SDR_GLO_NASS_GNAV_NAVIGATION_MESSAGE_H
00021
00022 #include "GLONASS_L1_L2_CA.h"
00023 #include "glonass_gnav_almanac.h"
00024 #include "glonass_gnav_ephemeris.h"
00025 #include "glonass_gnav_utc_model.h"
00026 #include <bitset>
00027 #include <cstdint>
00028 #include <map>
00029 #include <string>
00030 #include <utility> // for pair
00031 #include <vector> // for vector
00032
00033 /** \addtogroup Core
00034  * \{
00035  /** \addtogroup System_Parameters
00036  * \{
00037
00038  /*!
00039  * \brief This class decodes a GLONASS GNAV Data message as described in GLONASS ICD (Edition 5.1)
00040  * \note Code added as part of GSoC 2017 program
00041  * \see <a
00042  href="http://russianspacesystems.ru/wp-content/uploads/2016/08/ICD_GLONASS_eng_v5.1.pdf">GLONASS
00043  ICD</a>
00044  */
00045
00046 class Glonass_Gnav_Navigation_Message
00047 {
00048 public:
00049  /*!
00050  * Default constructor
00051  */
00052  Glonass_Gnav_Navigation_Message();
00053
00054  /*!
00055  * \brief Compute CRC for GLONASS GNAV strings
00056  * \param bits Bits of the string message where to compute CRC
00057  */
00058  bool CRC_test(std::bitset<GLONASS_GNAV_STRING_BITS>& bits) const;
00059
00060  /*!
00061  * \brief Computes the frame number being decoded given the satellite slot number
00062  * \param satellite_slot_number [in] Satellite slot number identifier

```

```

00063     * \returns Frame number being decoded, 0 if operation was not successful.
00064     */
00065 uint32_t get_frame_number(uint32_t satellite_slot_number);
00066
00067     /*!
00068     * \brief Obtain a GLONASS GNAV SV Ephemeris class filled with current SV data
00069     */
00070 Glonass_Gnav_Ephemeris get_ephemeris() const
00071 {
00072     return gnav_ephemeris;
00073 }
00074
00075     /*!
00076     * \brief Obtain a GLONASS GNAV UTC model parameters class filled with current SV data
00077     */
00078 inline Glonass_Gnav_Utc_Model get_utc_model() const
00079 {
00080     return gnav_utc_model;
00081 }
00082
00083     /*!
00084     * \brief Returns a Glonass_Gnav_Almanac object filled with the latest navigation data received
00085     * \param satellite_slot_number Slot number identifier for the satellite
00086     * \returns Returns the Glonass_Gnav_Almanac object for the input slot number
00087     */
00088 Glonass_Gnav_Almanac get_almanac(uint32_t satellite_slot_number) const;
00089
00090     /*!
00091     * \brief Returns true if a new Glonass_Gnav_Ephemeris object has arrived.
00092     */
00093 bool have_new_ephemeris();
00094
00095     /*!
00096     * \brief Returns true if new Glonass_Gnav_Utc_Model object has arrived
00097     */
00098 bool have_new_utc_model();
00099
00100     /*!
00101     * \brief Returns true if new Glonass_Gnav_Almanac object has arrived.
00102     */
00103 bool have_new_almanac();
00104
00105     /*!
00106     * \brief Decodes the GLONASS GNAV string
00107     * \param frame_string [in] is the string message within the parsed frame
00108     * \returns Returns the ID of the decoded string
00109     */
00110 int32_t string_decoder(const std::string& frame_string);
00111
00112 inline bool get_flag_CRC_test() const
00113 {
00114     return flag_CRC_test;
00115 }
00116
00117 inline void set_rf_link(int32_t rf_link)
00118 {
00119     gnav_ephemeris.i_satellite_freq_channel = rf_link;
00120 }
00121
00122 inline uint32_t get_alm_satellite_slot_number() const
00123 {
00124     return i_alm_satellite_slot_number;
00125 }
00126
00127 inline bool get_flag_update_slot_number() const
00128 {
00129     return flag_update_slot_number;
00130 }
00131
00132 inline void set_flag_update_slot_number(bool flag_slot)
00133 {
00134     flag_update_slot_number = flag_slot;
00135 }
00136
00137 inline bool get_flag_TOW_new() const
00138 {
00139     return flag_TOW_new;
00140 }
00141
00142 inline void set_flag_TOW_new(bool tow_new)
00143 {
00144     flag_TOW_new = tow_new;
00145 }
00146
00147 inline bool is_flag_TOW_set() const
00148 {
00149     return flag_TOW_set;

```

```

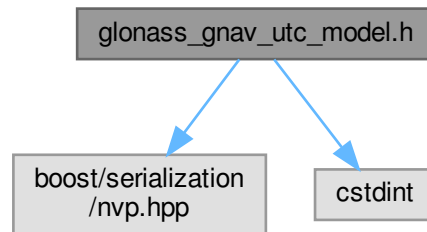
00150     }
00151
00152     inline void set_flag_ephemeris_str_1(bool ephemeris_str_1)
00153     {
00154         flag_ephemeris_str_1 = ephemeris_str_1;
00155     }
00156
00157     inline void set_flag_ephemeris_str_2(bool ephemeris_str_2)
00158     {
00159         flag_ephemeris_str_2 = ephemeris_str_2;
00160     }
00161
00162     inline void set_flag_ephemeris_str_3(bool ephemeris_str_3)
00163     {
00164         flag_ephemeris_str_3 = ephemeris_str_3;
00165     }
00166
00167     inline void set_flag_ephemeris_str_4(bool ephemeris_str_4)
00168     {
00169         flag_ephemeris_str_4 = ephemeris_str_4;
00170     }
00171
00172 private:
00173     uint64_t read_navigation_unsigned(const std::bitset<GLONASS_GNAV_STRING_BITS>& bits, const
std::vector<std::pair<int32_t, int32_t>& parameter) const;
00174     int64_t read_navigation_signed(const std::bitset<GLONASS_GNAV_STRING_BITS>& bits, const
std::vector<std::pair<int32_t, int32_t>& parameter) const;
00175     bool read_navigation_bool(const std::bitset<GLONASS_GNAV_STRING_BITS>& bits, const
std::vector<std::pair<int32_t, int32_t>& parameter) const;
00176
00177     Glonass_Gnav_Ephemeris gnav_ephemeris{};           // Ephemeris information decoded
00178     Glonass_Gnav_Utc_Model gnav_utc_model{};           // UTC model information
00179     Glonass_Gnav_Almanac gnav_almanac[GLONASS_CA_NBR_SATS]{}; // Almanac information for all 24
satellites
00180
00181     std::map<int, std::string> satelliteBlock; // Map that stores to which block the PRN belongs
00182
00183     double d_previous_tb{}; // Previous iode for the Glonass_Gnav_Ephemeris
object. Used to determine when new data arrives
00184     double d_previous_Na[GLONASS_CA_NBR_SATS]{}; // Previous time for almanac of the
Glonass_Gnav_Almanac object
00185
00186     uint32_t d_frame_ID{};
00187     uint32_t d_string_ID{};
00188     uint32_t i_alm_satellite_slot_number{}; // SV Orbit Slot Number
00189
00190     bool flag_CRC_test{};
00191     bool flag_update_slot_number{};
00192
00193     // Ephemeris Flags and control variables
00194     bool flag_all_ephemeris{}; // Flag indicating that all strings containing ephemeris have been
received
00195     bool flag_ephemeris_str_1{}; // Flag indicating that ephemeris 1/4 (string 1) have been received
00196     bool flag_ephemeris_str_2{}; // Flag indicating that ephemeris 2/4 (string 2) have been received
00197     bool flag_ephemeris_str_3{}; // Flag indicating that ephemeris 3/4 (string 3) have been received
00198     bool flag_ephemeris_str_4{}; // Flag indicating that ephemeris 4/4 (string 4) have been received
00199
00200     // Almanac Flags
00201     bool flag_almanac_str_6{}; // Flag indicating that almanac of string 6 have been received
00202     bool flag_almanac_str_7{}; // Flag indicating that almanac of string 7 have been received
00203     bool flag_almanac_str_8{}; // Flag indicating that almanac of string 8 have been received
00204     bool flag_almanac_str_9{}; // Flag indicating that almanac of string 9 have been received
00205     bool flag_almanac_str_10{}; // Flag indicating that almanac of string 10 have been received
00206     bool flag_almanac_str_11{}; // Flag indicating that almanac of string 11 have been received
00207     bool flag_almanac_str_12{}; // Flag indicating that almanac of string 12 have been received
00208     bool flag_almanac_str_13{}; // Flag indicating that almanac of string 13 have been received
00209     bool flag_almanac_str_14{}; // Flag indicating that almanac of string 14 have been received
00210     bool flag_almanac_str_15{}; // Flag indicating that almanac of string 15 have been received
00211
00212     // UTC and System Clocks Flags
00213     bool flag_utc_model_str_5{}; // Clock info send in string 5 of navigation data
00214
00215     bool flag_TOW_set{}; // Flag indicating when the TOW has been set
00216     bool flag_TOW_new{}; // Flag indicating when a new TOW has been computed
00217 };
00218
00219
00220 /** \} */
00221 /** \} */
00222 #endif // GNSS_SDR_GLOASS_GNAV_NAVIGATION_MESSAGE_H

```

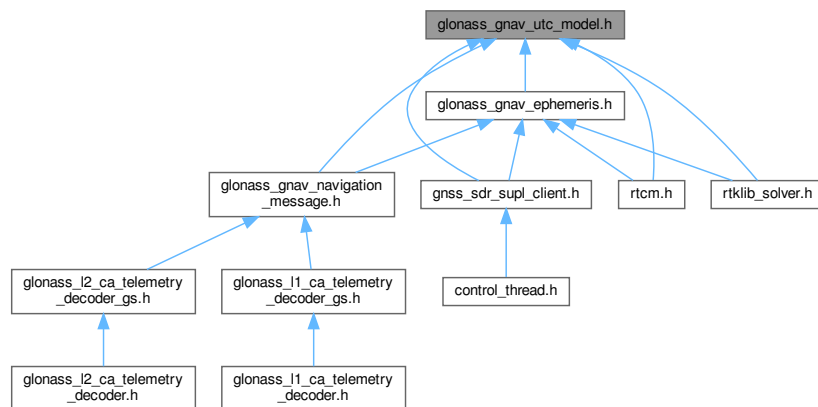
## 11.737 glonass\_gnav\_utc\_model.h File Reference

Interface of a GLONASS GNAV UTC MODEL storage.

```
#include <boost/serialization/nvp.hpp>
#include <cstdint>
Include dependency graph for glonass_gnav_utc_model.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [Glonass\\_Gnav\\_Utc\\_Model](#)

*This class is a storage for the GLONASS GNAV UTC MODEL data as described in GLONASS ICD (Edition 5.1)*

## 11.737.1 Detailed Description

Interface of a GLONASS GNAV UTC MODEL storage.

### Note

Code added as part of GSoC 2017 program

### Author

Damian Miralles, 2017. dmiralles2009(at)gmail.com



See also

GLONASS ICD

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [glonass\\_gnav\\_utc\\_model.h](#).

---

## 11.738 glonass\_gnav\_utc\_model.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file glonass_gnav_utc_model.h
00003  * \brief Interface of a GLONASS GNAV UTC MODEL storage
00004  * \note Code added as part of GSoC 2017 program
00005  * \author Damian Miralles, 2017. dmiralles2009(at)gmail.com
00006  * \see <a
00007  href="http://russianspacesystems.ru/wp-content/uploads/2016/08/ICD_GLONASS_eng_v5.1.pdf">GLONASS
00008  ICD</a>
00009  *
00010  * -----
00011  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00012  * This file is part of GNSS-SDR.
00013  *
00014  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00015  * SPDX-License-Identifier: GPL-3.0-or-later
00016  * -----
00017  */
00018
00019
00020 #ifndef GNSS_SDR_GLOMSS_GNAV_UTC_MODEL_H
00021 #define GNSS_SDR_GLOMSS_GNAV_UTC_MODEL_H
00022
00023 #include <boost/serialization/nvp.hpp>
00024 #include <cstdint>
00025
00026 /** \addtogroup Core
00027  * \{ */
00028 /** \addtogroup System_Parameters
00029  * \{ */
00030
00031
00032  /*!
00033  * \brief This class is a storage for the GLONASS GNAV UTC MODEL data as described in GLONASS ICD
00034  (Edition 5.1)
00035  * \note Code added as part of GSoC 2017 program
00036  * \see <a
00037  href="http://russianspacesystems.ru/wp-content/uploads/2016/08/ICD_GLONASS_eng_v5.1.pdf">GLONASS
00038  ICD</a>
00039  */
00040  class Glonass_Gnav_Utc_Model
00041  {
00042  public:
00043  /*!
00044  * Default constructor
00045  */
00046  Glonass_Gnav_Utc_Model() = default;
00047
00048  bool valid{};
00049  // Clock Parameters
00050  double d_tau_c{};    //!< GLONASS time scale correction to UTC(SU) time. [s]
00051  double d_tau_gps{};  //!< Correction to GPS time to GLONASS time [day]
00052  double d_N_4{};      //!< Four year interval number starting from 1996 [4 year interval]
00053  double d_N_A{};      //!< Calendar day number within the four-year period beginning since the leap
00054  year for Almanac data [days]
00055  double d_B1{};        //!< Coefficient to determine DeltaUT1 [s]
00056  double d_B2{};        //!< Coefficient to determine DeltaUT1 [s/msd]
00057
00058  /*!
00059  * \brief Computes the Coordinated Universal Time (UTC) and
00060  * returns it in [s] (GLONASS ICD (Edition 5.1) Section 3.3.3 GLONASS Time)
00061  */
00062  double utc_time(double glonass_time_corrected) const;
00063
00064  template <class Archive>
00065  /*!
00066  * \brief Serialize is a boost standard method to be called by the boost XML
00067  * serialization. Here is used to save the almanac data on disk file.
00068  */
00069  void serialize(Archive& archive, const uint32_t version)
00070  {

```

```

00067         if (version)
00068         {
00069             };
00070         archive& BOOST_SERIALIZATION_NVP (valid);
00071         archive& BOOST_SERIALIZATION_NVP (d_tau_c);
00072         archive& BOOST_SERIALIZATION_NVP (d_tau_gps);
00073         archive& BOOST_SERIALIZATION_NVP (d_N_4);
00074         archive& BOOST_SERIALIZATION_NVP (d_N_A);
00075         archive& BOOST_SERIALIZATION_NVP (d_B1);
00076         archive& BOOST_SERIALIZATION_NVP (d_B2);
00077     }
00078 };
00079
00080
00081 /** \} */
00082 /** \} */
00083 #endif // GNSS_SDR_GLO_NASS_GNAV_UTC_MODEL_H

```

## 11.739 GLONASS\_L1\_L2\_CA.h File Reference

Defines system parameters for GLONASS L1 C/A signal and NAV data.

```
#include "gnss_frequencies.h"
```

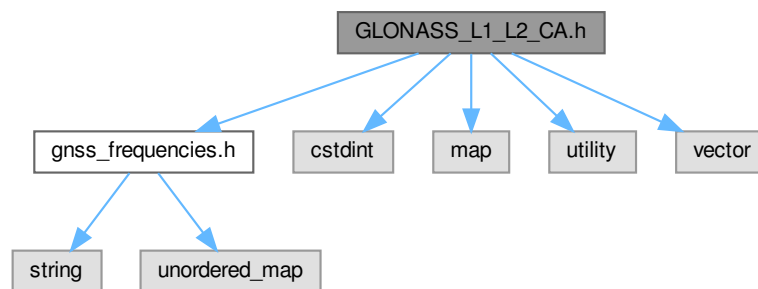
```
#include <cstdint>
```

```
#include <map>
```

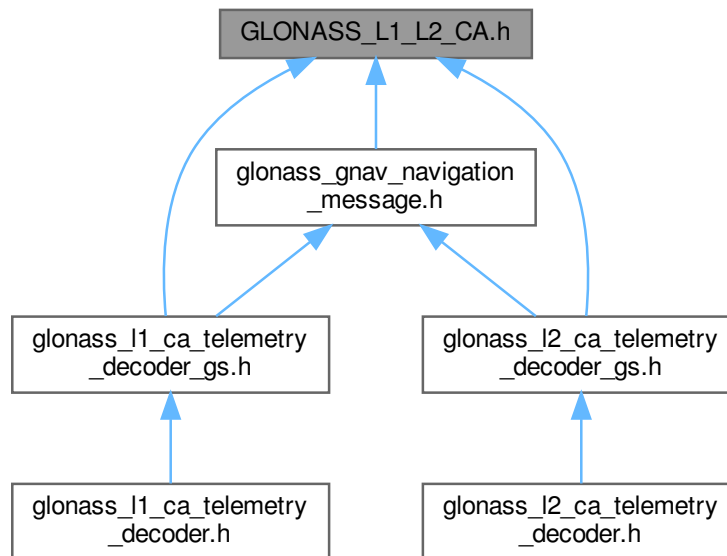
```
#include <utility>
```

```
#include <vector>
```

Include dependency graph for GLONASS\_L1\_L2\_CA.h:



This graph shows which files directly or indirectly include this file:



## Macros

- `#define` [GLONASS\\_GNAV\\_PREAMBLE](#)

## Functions

- `const std::vector< std::pair< int32_t, int32_t > >` **STRING\_ID** ({2, 4}}
- `const std::vector< std::pair< int32_t, int32_t > >` **KX** ({78, 8}}
- `const std::vector< std::pair< int32_t, int32_t > >` **P1** ({8, 2}}
- `const std::vector< std::pair< int32_t, int32_t > >` **T\_K\_HR** ({10, 5}}
- `const std::vector< std::pair< int32_t, int32_t > >` **T\_K\_MIN** ({15, 6}}
- `const std::vector< std::pair< int32_t, int32_t > >` **T\_K\_SEC** ({21, 1}}
- `const std::vector< std::pair< int32_t, int32_t > >` **X\_N\_DOT** ({22, 24}}
- `const std::vector< std::pair< int32_t, int32_t > >` **X\_N\_DOT\_DOT** ({46, 5}}
- `const std::vector< std::pair< int32_t, int32_t > >` **X\_N** ({51, 27}}
- `const std::vector< std::pair< int32_t, int32_t > >` **B\_N** ({6, 3}}
- `const std::vector< std::pair< int32_t, int32_t > >` **P2** ({9, 1}}
- `const std::vector< std::pair< int32_t, int32_t > >` **T\_B** ({10, 7}}
- `const std::vector< std::pair< int32_t, int32_t > >` **Y\_N\_DOT** ({22, 24}}
- `const std::vector< std::pair< int32_t, int32_t > >` **Y\_N\_DOT\_DOT** ({46, 5}}
- `const std::vector< std::pair< int32_t, int32_t > >` **Y\_N** ({51, 27}}
- `const std::vector< std::pair< int32_t, int32_t > >` **P3** ({6, 1}}
- `const std::vector< std::pair< int32_t, int32_t > >` **GAMMA\_N** ({7, 11}}
- `const std::vector< std::pair< int32_t, int32_t > >` **P** ({19, 2}}
- `const std::vector< std::pair< int32_t, int32_t > >` **EPH\_L\_N** ({21, 1}}
- `const std::vector< std::pair< int32_t, int32_t > >` **Z\_N\_DOT** ({22, 24}}
- `const std::vector< std::pair< int32_t, int32_t > >` **Z\_N\_DOT\_DOT** ({46, 5}}
- `const std::vector< std::pair< int32_t, int32_t > >` **Z\_N** ({51, 27}}
- `const std::vector< std::pair< int32_t, int32_t > >` **TAU\_N** ({6, 22}}

- `const std::vector< std::pair< int32_t, int32_t > > DELTA_TAU_N` ( $\{\{28, 5\}\}$ )
- `const std::vector< std::pair< int32_t, int32_t > > E_N` ( $\{\{33, 5\}\}$ )
- `const std::vector< std::pair< int32_t, int32_t > > P4` ( $\{\{52, 1\}\}$ )
- `const std::vector< std::pair< int32_t, int32_t > > F_T` ( $\{\{53, 4\}\}$ )
- `const std::vector< std::pair< int32_t, int32_t > > N_T` ( $\{\{60, 11\}\}$ )
- `const std::vector< std::pair< int32_t, int32_t > > N` ( $\{\{71, 5\}\}$ )
- `const std::vector< std::pair< int32_t, int32_t > > M` ( $\{\{76, 2\}\}$ )
- `const std::vector< std::pair< int32_t, int32_t > > DAY_NUMBER_A` ( $\{\{6, 11\}\}$ )
- `const std::vector< std::pair< int32_t, int32_t > > TAU_C` ( $\{\{17, 32\}\}$ )
- `const std::vector< std::pair< int32_t, int32_t > > N_4` ( $\{\{50, 5\}\}$ )
- `const std::vector< std::pair< int32_t, int32_t > > TAU_GPS` ( $\{\{55, 22\}\}$ )
- `const std::vector< std::pair< int32_t, int32_t > > ALM_L_N` ( $\{\{77, 1\}\}$ )
- `const std::vector< std::pair< int32_t, int32_t > > C_N` ( $\{\{6, 1\}\}$ )
- `const std::vector< std::pair< int32_t, int32_t > > M_N_A` ( $\{\{7, 2\}\}$ )
- `const std::vector< std::pair< int32_t, int32_t > > N_A` ( $\{\{9, 5\}\}$ )
- `const std::vector< std::pair< int32_t, int32_t > > TAU_N_A` ( $\{\{14, 10\}\}$ )
- `const std::vector< std::pair< int32_t, int32_t > > LAMBDA_N_A` ( $\{\{24, 21\}\}$ )
- `const std::vector< std::pair< int32_t, int32_t > > DELTA_I_N_A` ( $\{\{45, 18\}\}$ )
- `const std::vector< std::pair< int32_t, int32_t > > EPSILON_N_A` ( $\{\{63, 15\}\}$ )
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA_N_A` ( $\{\{6, 16\}\}$ )
- `const std::vector< std::pair< int32_t, int32_t > > T_LAMBDA_N_A` ( $\{\{22, 21\}\}$ )
- `const std::vector< std::pair< int32_t, int32_t > > DELTA_T_N_A` ( $\{\{43, 22\}\}$ )
- `const std::vector< std::pair< int32_t, int32_t > > DELTA_T_DOT_N_A` ( $\{\{65, 7\}\}$ )
- `const std::vector< std::pair< int32_t, int32_t > > H_N_A` ( $\{\{72, 5\}\}$ )
- `const std::vector< std::pair< int32_t, int32_t > > B1` ( $\{\{6, 11\}\}$ )
- `const std::vector< std::pair< int32_t, int32_t > > B2` ( $\{\{17, 10\}\}$ )

## Variables

- `constexpr double GLONASS_F_M_A = 0.35e9`  
*Gravitational constant of atmosphere [ $m^3/s^2$ ].*
- `constexpr double GLONASS_SEMI_MAJOR_AXIS = 6378136`  
*Semi-major axis of Earth [m].*
- `constexpr double GLONASS_FLATTENING = 1.0 / 29825784.0`  
*Flattening parameter.*
- `constexpr double GLONASS_GRAVITY = 97803284.0`  
*Equatorial acceleration of gravity [mGal].*
- `constexpr double GLONASS_GRAVITY_CORRECTION = 0.87`  
*Correction to acceleration of gravity at sea-level due to Atmosphere[uGal].*
- `constexpr double GLONASS_J2 = 1082625.75e-9`  
*Second zonal harmonic of the geopotential.*
- `constexpr double GLONASS_J4 = -2370.89e-9`  
*Fourth zonal harmonic of the geopotential.*
- `constexpr double GLONASS_J6 = 6.08e-9`  
*Sixth zonal harmonic of the geopotential.*
- `constexpr double GLONASS_J8 = 1.40e-11`  
*Eighth zonal harmonic of the geopotential.*
- `constexpr double GLONASS_U0 = 62636861.4`  
*Normal potential at surface of common terrestrial ellipsoid [ $m^2/s^2$ ].*
- `constexpr double GLONASS_C20 = -1082.63e-6`  
*Second zonal coefficient of spherical harmonic expansion.*
- `constexpr double GLONASS_EARTH_RADIUS = 6378.136`  
*Equatorial radius of Earth [km].*

- constexpr double [GLONASS\\_EARTH\\_INCLINATION](#) = 0.000409148809899e3  
*Mean inclination of ecliptic to equator (23 deg 26 min 33 sec) [rad].*
- constexpr double [GLONASS\\_TAU\\_0](#) = -0.005835151531174e3  
*(-334 deg 19 min 46.40 sec) [rad];*
- constexpr double [GLONASS\\_TAU\\_1](#) = 0.071018041257371e3  
*(4069 deg 02 min 02.52 sec) [rad];*
- constexpr double [GLONASS\\_MOON\\_Q0](#) = -0.001115184961435e3  
*(-63 deg 53 min 43.41 sec) [rad]*
- constexpr double [GLONASS\\_MOON\\_Q1](#) = 8.328691103668023e3  
*(477198 deg 50 min 56.79 sec) [rad]*
- constexpr double [GLONASS\\_MOON\\_OMEGA\\_0](#) = 0.004523601514852e3  
*(259 deg 10 min 59.79 sec) [rad]*
- constexpr double [GLONASS\\_MOON\\_OMEGA\\_1](#) = -0.033757146246552e3  
*(-1934 deg 08 min 31.23 sec) [rad]*
- constexpr double [GLONASS\\_MOON\\_GM](#) = 4902.835  
*Lunar gravitational constant [km<sup>3</sup>/s<sup>2</sup>].*
- constexpr double [GLONASS\\_MOON\\_SEMI\\_MAJOR\\_AXIS](#) = 3.84385243e5  
*Semi-major axis of lunar orbit [km].;*
- constexpr double [GLONASS\\_MOON\\_ECCENTRICITY](#) = 0.054900489  
*Eccentricity of lunar orbit.*
- constexpr double [GLONASS\\_MOON\\_INCLINATION](#) = 0.000089803977407e3  
*Inclination of lunar orbit to ecliptic plane (5 deg 08 min 43.4 sec) [rad].*
- constexpr double [GLONASS\\_SUN\\_OMEGA](#) = 0.004908229466869e3  
*TODO What is this operation in the seconds with T?(281 deg 13 min 15.0 + 6189.03 x T sec) [rad].*
- constexpr double [GLONASS\\_SUN\\_Q0](#) = 0.006256583774423e3  
*(358 deg 28 min 33.04 sec) [rad]*
- constexpr double [GLONASS\\_SUN\\_Q1](#) = 0e3  
*TODO Why is the value greater than 60?(129596579.10 sec) [rad].*
- constexpr double [GLONASS\\_SUN\\_GM](#) = 0.1325263e12  
*Solar gravitational constant [km<sup>3</sup>/s<sup>2</sup>].*
- constexpr double [GLONASS\\_SUN\\_SEMI\\_MAJOR\\_AXIS](#) = 1.49598e8  
*Semi-major axis of solar orbit [km].;*
- constexpr double [GLONASS\\_SUN\\_ECCENTRICITY](#) = 0.016719  
*Eccentricity of solar orbit.*
- constexpr double [GLONASS\\_L2\\_CA\\_FREQ\\_HZ](#) = [FREQ2\\_GLO](#)  
*L2 [Hz].*
- constexpr double [GLONASS\\_L2\\_CA\\_DFREQ\\_HZ](#) = [DFRQ2\\_GLO](#)  
*Freq Bias for GLONASS L1 [Hz].*
- constexpr double [GLONASS\\_L2\\_CA\\_CODE\\_RATE\\_CPS](#) = 0.511e6  
*GLONASS L1 C/A code rate [chips/s].*
- constexpr double [GLONASS\\_L2\\_CA\\_CODE\\_LENGTH\\_CHIPS](#) = 511.0  
*GLONASS L1 C/A code length [chips].*
- constexpr double [GLONASS\\_L2\\_CA\\_CODE\\_PERIOD\\_S](#) = 0.001  
*GLONASS L1 C/A code period [seconds].*
- constexpr double [GLONASS\\_L2\\_CA\\_CHIP\\_PERIOD\\_S](#) = 1.9569e-06  
*GLONASS L1 C/A chip period [seconds].*
- constexpr double [GLONASS\\_L2\\_CA\\_SYMBOL\\_RATE\\_BPS](#) = 1000.0
- constexpr double [GLONASS\\_L1\\_CA\\_FREQ\\_HZ](#) = [FREQ1\\_GLO](#)  
*L1 [Hz].*
- constexpr double [GLONASS\\_L1\\_CA\\_DFREQ\\_HZ](#) = [DFRQ1\\_GLO](#)  
*Freq Bias for GLONASS L1 [Hz].*

- constexpr double [GLONASS\\_L1\\_CA\\_CODE\\_RATE\\_CPS](#) = 0.511e6  
*GLONASS L1 C/A code rate [chips/s].*
- constexpr double [GLONASS\\_L1\\_CA\\_CODE\\_LENGTH\\_CHIPS](#) = 511.0  
*GLONASS L1 C/A code length [chips].*
- constexpr double [GLONASS\\_L1\\_CA\\_CODE\\_PERIOD\\_S](#) = 0.001  
*GLONASS L1 C/A code period [seconds].*
- constexpr double [GLONASS\\_L1\\_CA\\_CHIP\\_PERIOD\\_S](#) = 1.9569e-06  
*GLONASS L1 C/A chip period [seconds].*
- constexpr double [GLONASS\\_L1\\_CA\\_SYMBOL\\_RATE\\_BPS](#) = 1000.0
- constexpr int32\_t [GLONASS\\_CA\\_NBR\\_SATS](#) = 24
- constexpr int32\_t [GLONASS\\_L1\\_CA\\_HISTORY\\_DEEP](#) = 100
- constexpr double [GLONASS\\_GNAV\\_PREAMBLE\\_DURATION\\_S](#) = 0.300
- constexpr int32\_t [GLONASS\\_GNAV\\_PREAMBLE\\_LENGTH\\_BITS](#) = 30
- constexpr int32\_t [GLONASS\\_GNAV\\_PREAMBLE\\_LENGTH\\_SYMBOLS](#) = 300
- constexpr int32\_t [GLONASS\\_GNAV\\_PREAMBLE\\_PERIOD\\_SYMBOLS](#) = 2000
- constexpr int32\_t [GLONASS\\_GNAV\\_TELEMETRY\\_RATE\\_BITS\\_SECOND](#) = 50  
*NAV message bit rate [bits/s].*
- constexpr int32\_t [GLONASS\\_GNAV\\_TELEMETRY\\_SYMBOLS\\_PER\\_BIT](#) = 10
- constexpr int32\_t [GLONASS\\_GNAV\\_TELEMETRY\\_SYMBOLS\\_PER\\_PREAMBLE\\_BIT](#) = 10
- constexpr int32\_t [GLONASS\\_GNAV\\_TELEMETRY\\_RATE\\_SYMBOLS\\_SECOND](#) = [GLONASS\\_GNAV\\_TELEMETRY\\_RATE\\_BITS\\_SECOND](#) \* [GLONASS\\_GNAV\\_TELEMETRY\\_SYMBOLS\\_PER\\_BIT](#)  
*NAV message bit rate [symbols/s].*
- constexpr int32\_t [GLONASS\\_GNAV\\_STRING\\_SYMBOLS](#) = 2000  
*Number of bits per string in the GNAV message (85 data bits + 30 time mark bits) [bits].*
- constexpr int32\_t [GLONASS\\_GNAV\\_STRING\\_BITS](#) = 85  
*Number of bits per string in the GNAV message (85 data bits + 30 time mark bits) [bits].*
- constexpr int32\_t [GLONASS\\_GNAV\\_HAMMING\\_CODE\\_BITS](#) = 8  
*Number of bits in hamming code sequence of GNAV message.*
- constexpr int32\_t [GLONASS\\_GNAV\\_DATA\\_SYMBOLS](#) = 1700
- constexpr double [GLONASS\\_LEAP\\_SECONDS](#) [19][7]  
*Record of leap seconds definition for GLOT to GPST conversion and vice versa.*
- const std::map< uint32\_t, int32\_t > [GLONASS\\_PRN](#)
- const std::vector< int32\_t > [GLONASS\\_GNAV\\_CRC\\_I\\_INDEX](#) {9, 10, 12, 13, 15, 17, 19, 20, 22, 24, 26, 28, 30, 32, 34, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 66, 68, 70, 72, 74, 76, 78, 80, 82, 84}
- const std::vector< int32\_t > [GLONASS\\_GNAV\\_CRC\\_J\\_INDEX](#) {9, 11, 12, 14, 15, 18, 19, 21, 22, 25, 26, 29, 30, 33, 34, 36, 37, 40, 41, 44, 45, 48, 49, 52, 53, 56, 57, 60, 61, 64, 65, 67, 68, 71, 72, 75, 76, 79, 80, 83, 84}
- const std::vector< int32\_t > [GLONASS\\_GNAV\\_CRC\\_K\\_INDEX](#) {10, 11, 12, 16, 17, 18, 19, 23, 24, 25, 26, 31, 32, 33, 34, 38, 39, 40, 41, 46, 47, 48, 49, 54, 55, 56, 57, 62, 63, 64, 65, 69, 70, 71, 72, 77, 78, 79, 80, 85}
- const std::vector< int32\_t > [GLONASS\\_GNAV\\_CRC\\_L\\_INDEX](#) {13, 14, 15, 16, 17, 18, 19, 27, 28, 29, 30, 31, 32, 33, 34, 42, 43, 44, 45, 46, 47, 48, 49, 58, 59, 60, 61, 62, 63, 64, 65, 73, 74, 75, 76, 77, 78, 79, 80}
- const std::vector< int32\_t > [GLONASS\\_GNAV\\_CRC\\_M\\_INDEX](#) {20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 81, 82, 83, 84, 85}
- const std::vector< int32\_t > [GLONASS\\_GNAV\\_CRC\\_N\\_INDEX](#) {35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65}
- const std::vector< int32\_t > [GLONASS\\_GNAV\\_CRC\\_P\\_INDEX](#) {66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85}
- const std::vector< int32\_t > [GLONASS\\_GNAV\\_CRC\\_Q\\_INDEX](#) {9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85}
- const std::vector< int32\_t > [GLONASS\\_GNAV\\_ECC\\_LOCATOR](#) {0, 0, 1, 8, 2, 9, 10, 11, 3, 12, 13, 14, 15, 16, 17, 18, 4, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 5, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 6, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84}

## 11.739.1 Detailed Description

Defines system parameters for GLONASS L1 C/A signal and NAV data.

### Note

File renamed from GLONASS\_L1\_CA.h to [GLONASS\\_L1\\_L2\\_CA.h](#) to accommodate GLO L2 addition

### Author

Damian Miralles, 2017. dmiralles2009(at)gmail.com

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [GLONASS\\_L1\\_L2\\_CA.h](#).

---

## 11.740 GLONASS\_L1\_L2\_CA.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file GLONASS_L1_L2_CA.h
00003  * \brief Defines system parameters for GLONASS L1 C/A signal and NAV data
00004  * \note File renamed from GLONASS_L1_CA.h to GLONASS_L1_L2_CA.h to accommodate GLO L2 addition
00005  * \author Damian Miralles, 2017. dmiralles2009(at)gmail.com
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018
00019 #ifndef GNSS_SDR_GLONASS_L1_L2_CA_H
00020 #define GNSS_SDR_GLONASS_L1_L2_CA_H
00021
00022 #include "gnss_frequencies.h"
00023 #include <stdint>
00024 #include <map>
00025 #include <utility> // std::pair
00026 #include <vector>
00027
00028 /** \addtogroup Core
00029  * \{ */
00030 /** \addtogroup System_Parameters
00031  * \{ */
00032
00033
00034 // Physical constants
00035 constexpr double GLONASS_F_M_A = 0.35e9; //!< Gravitational constant of
atmosphere [m^3/s^2]
00036 constexpr double GLONASS_SEMI_MAJOR_AXIS = 6378136; //!< Semi-major axis of Earth [m]
00037 constexpr double GLONASS_FLATTENING = 1.0 / 29825784.0; //!< Flattening parameter
00038 constexpr double GLONASS_GRAVITY = 97803284.0; //!< Equatorial acceleration of
gravity [mGal]
00039 constexpr double GLONASS_GRAVITY_CORRECTION = 0.87; //!< Correction to acceleration of
gravity at sea-level due to Atmosphere[uGal]
00040 constexpr double GLONASS_J2 = 1082625.75e-9; //!< Second zonal harmonic of the
geopotential
00041 constexpr double GLONASS_J4 = -2370.89e-9; //!< Fourth zonal harmonic of the
geopotential
00042 constexpr double GLONASS_J6 = 6.08e-9; //!< Sixth zonal harmonic of the
geopotential
00043 constexpr double GLONASS_J8 = 1.40e-11; //!< Eighth zonal harmonic of the
geopotential
00044 constexpr double GLONASS_U0 = 62636861.4; //!< Normal potential at surface of
common terrestrial ellipsoid [m^2/s^2]
00045 constexpr double GLONASS_C20 = -1082.63e-6; //!< Second zonal coefficient of
spherical harmonic expansion
00046 constexpr double GLONASS_EARTH_RADIUS = 6378.136; //!< Equatorial radius of Earth
[km]
00047 constexpr double GLONASS_EARTH_INCLINATION = 0.000409148809899e3; //!< Mean inclination of ecliptic
to equator (23 deg 26 min 33 sec) [rad]
00048
00049 constexpr double GLONASS_TAU_0 = -0.005835151531174e3; //!< (-334 deg 19 min 46.40 sec) [rad];
00050 constexpr double GLONASS_TAU_1 = 0.071018041257371e3; //!< (4069 deg 02 min 02.52 sec) [rad];
```



```

00051
00052 constexpr double GLONASS_MOON_Q0 = -0.001115184961435e3;          //!< (-63 deg 53 min 43.41 sec)
[rad]
00053 constexpr double GLONASS_MOON_Q1 = 8.328691103668023e3;          //!< (477198 deg 50 min 56.79 sec)
[rad]
00054 constexpr double GLONASS_MOON_OMEGA_0 = 0.004523601514852e3;      //!< (259 deg 10 min 59.79 sec)
[rad]
00055 constexpr double GLONASS_MOON_OMEGA_1 = -0.033757146246552e3;    //!< (-1934 deg 08 min 31.23 sec)
[rad]
00056 constexpr double GLONASS_MOON_GM = 4902.835;                    //!< Lunar gravitational constant
[km^3/s^2]
00057 constexpr double GLONASS_MOON_SEMI_MAJOR_AXIS = 3.84385243e5;   //!< Semi-major axis of lunar orbit
[km];
00058 constexpr double GLONASS_MOON_ECCENTRICITY = 0.054900489;        //!< Eccentricity of lunar orbit
00059 constexpr double GLONASS_MOON_INCLINATION = 0.000089803977407e3; //!< Inclination of lunar orbit to
ecliptic plane (5 deg 08 min 43.4 sec) [rad]
00060
00061 constexpr double GLONASS_SUN_OMEGA = 0.004908229466869e3;        //!< TODO What is this operation in the
seconds with T?(281 deg 13 min 15.0 + 6189.03 x T sec) [rad]
00062 constexpr double GLONASS_SUN_Q0 = 0.006256583774423e3;          //!< (358 deg 28 min 33.04 sec) [rad]
00063 constexpr double GLONASS_SUN_Q1 = 0e3;                            //!< TODO Why is the value greater than
60?(129596579.10 sec) [rad]
00064 constexpr double GLONASS_SUN_GM = 0.1325263e12;                //!< Solar gravitational constant
[km^3/s^2]
00065 constexpr double GLONASS_SUN_SEMI_MAJOR_AXIS = 1.49598e8;      //!< Semi-major axis of solar orbit [km];
00066 constexpr double GLONASS_SUN_ECCENTRICITY = 0.016719;            //!< Eccentricity of solar orbit
00067
00068 constexpr double GLONASS_L2_CA_FREQ_HZ = FREQ2_GLO;              //!< L2 [Hz]
00069 constexpr double GLONASS_L2_CA_DFREQ_HZ = DFRQ2_GLO;             //!< Freq Bias for GLONASS L1 [Hz]
00070 constexpr double GLONASS_L2_CA_CODE_RATE_CPS = 0.511e6;         //!< GLONASS L1 C/A code rate [chips/s]
00071 constexpr double GLONASS_L2_CA_CODE_LENGTH_CHIPS = 511.0;       //!< GLONASS L1 C/A code length [chips]
00072 constexpr double GLONASS_L2_CA_CODE_PERIOD_S = 0.001;           //!< GLONASS L1 C/A code period [seconds]
00073 constexpr double GLONASS_L2_CA_CHIP_PERIOD_S = 1.9569e-06;      //!< GLONASS L1 C/A chip period [seconds]
00074 constexpr double GLONASS_L2_CA_SYMBOL_RATE_BPS = 1000.0;
00075
00076 constexpr double GLONASS_L1_CA_FREQ_HZ = FREQ1_GLO;              //!< L1 [Hz]
00077 constexpr double GLONASS_L1_CA_DFREQ_HZ = DFRQ1_GLO;             //!< Freq Bias for GLONASS L1 [Hz]
00078 constexpr double GLONASS_L1_CA_CODE_RATE_CPS = 0.511e6;         //!< GLONASS L1 C/A code rate [chips/s]
00079 constexpr double GLONASS_L1_CA_CODE_LENGTH_CHIPS = 511.0;       //!< GLONASS L1 C/A code length [chips]
00080 constexpr double GLONASS_L1_CA_CODE_PERIOD_S = 0.001;           //!< GLONASS L1 C/A code period [seconds]
00081 constexpr double GLONASS_L1_CA_CHIP_PERIOD_S = 1.9569e-06;      //!< GLONASS L1 C/A chip period [seconds]
00082 constexpr double GLONASS_L1_CA_SYMBOL_RATE_BPS = 1000.0;
00083
00084 constexpr int32_t GLONASS_CA_NBR_SATS = 24; // STRING DATA WITHOUT PREAMBLE
00085
00086 // OBSERVABLE HISTORY DEEP FOR INTERPOLATION
00087 constexpr int32_t GLONASS_L1_CA_HISTORY_DEEP = 100;
00088
00089 // NAVIGATION MESSAGE DEMODULATION AND DECODING
00090 #define GLONASS_GNAV_PREAMBLE \
00091 { \
00092     1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0 \
00093 }
00093 constexpr double GLONASS_GNAV_PREAMBLE_DURATION_S = 0.300;
00094 constexpr int32_t GLONASS_GNAV_PREAMBLE_LENGTH_BITS = 30;
00095 constexpr int32_t GLONASS_GNAV_PREAMBLE_LENGTH_SYMBOLS = 300;
00096 constexpr int32_t GLONASS_GNAV_PREAMBLE_PERIOD_SYMBOLS = 2000;
00097 constexpr int32_t GLONASS_GNAV_TELEMETRY_RATE_BITS_SECOND = 50;   //!< NAV message bit rate [bits/s]
00098 constexpr int32_t GLONASS_GNAV_TELEMETRY_SYMBOLS_PER_BIT = 10;
00099 constexpr int32_t GLONASS_GNAV_TELEMETRY_SYMBOLS_PER_PREAMBLE_BIT = 10;
00100 constexpr int32_t GLONASS_GNAV_TELEMETRY_RATE_SYMBOLS_SECOND = GLONASS_GNAV_TELEMETRY_RATE_BITS_SECOND
* GLONASS_GNAV_TELEMETRY_SYMBOLS_PER_BIT; //!< NAV message bit rate [symbols/s]
00101 constexpr int32_t GLONASS_GNAV_STRING_SYMBOLS = 2000;
00102 //!< Number of bits per string in the GNAV message (85 data bits + 30 time mark bits) [bits]
00102 constexpr int32_t GLONASS_GNAV_STRING_BITS = 85;
00103 //!< Number of bits per string in the GNAV message (85 data bits + 30 time mark bits) [bits]
00103 constexpr int32_t GLONASS_GNAV_HAMMING_CODE_BITS = 8;
00104 //!< Number of bits in hamming code sequence of GNAV message
00104 constexpr int32_t GLONASS_GNAV_DATA_SYMBOLS = 1700;
00105 // STRING DATA WITHOUT PREAMBLE
00105
00106 /*!
00107 * \brief Record of leap seconds definition for GLOT to GPST conversion and vice versa
00108 * \details Each entry is defined by an array of 7 elements consisting of
00109 yr,month,day,hr,min,sec,utc-gpst
00109 * \note Ideally should use leap seconds definitions of rtklib
00110 */
00111 constexpr double GLONASS_LEAP_SECONDS[19][7] = {
00112     {2017, 1, 1, 0, 0, 0, -18},
00113     {2015, 7, 1, 0, 0, 0, -17},
00114     {2012, 7, 1, 0, 0, 0, -16},
00115     {2009, 1, 1, 0, 0, 0, -15},
00116     {2006, 1, 1, 0, 0, 0, -14},
00117     {1999, 1, 1, 0, 0, 0, -13},
00118     {1997, 7, 1, 0, 0, 0, -12},
00119     {1996, 1, 1, 0, 0, 0, -11},
00120     {1994, 7, 1, 0, 0, 0, -10},
00121     {1993, 7, 1, 0, 0, 0, -9},

```



```
00122     {1992, 7, 1, 0, 0, 0, -8},
00123     {1991, 1, 1, 0, 0, 0, -7},
00124     {1990, 1, 1, 0, 0, 0, -6},
00125     {1988, 1, 1, 0, 0, 0, -5},
00126     {1985, 7, 1, 0, 0, 0, -4},
00127     {1983, 7, 1, 0, 0, 0, -3},
00128     {1982, 7, 1, 0, 0, 0, -2},
00129     {1981, 7, 1, 0, 0, 0, -1},
00130     {}};
00131
00132     //!< GLONASS SV's orbital slots PRN = (orbital_slot - 1)
00133     const std::map<uint32_t, int32_t> GLONASS_PRN = {
00134     {
00135         0,
00136         8,
00137     }, // For test
00138     {
00139         1,
00140         1,
00141     }, // Plane 1
00142     {
00143         2,
00144         -4,
00145     }, // Plane 1
00146     {
00147         3,
00148         5,
00149     }, // Plane 1
00150     {
00151         4,
00152         6,
00153     }, // Plane 1
00154     {
00155         5,
00156         1,
00157     }, // Plane 1
00158     {
00159         6,
00160         -4,
00161     }, // Plane 1
00162     {
00163         7,
00164         5,
00165     }, // Plane 1
00166     {
00167         8,
00168         6,
00169     }, // Plane 1
00170     {
00171         9,
00172         -2,
00173     }, // Plane 2
00174     {
00175         10,
00176         -7,
00177     }, // Plane 2
00178     {
00179         11,
00180         0,
00181     }, // Plane 2
00182     {
00183         12,
00184         -1,
00185     }, // Plane 2
00186     {
00187         13,
00188         -2,
00189     }, // Plane 2
00190     {
00191         14,
00192         -7,
00193     }, // Plane 2
00194     {
00195         15,
00196         0,
00197     }, // Plane 2
00198     {
00199         16,
00200         -1,
00201     }, // Plane 2
00202     {
00203         17,
00204         4,
00205     }, // Plane 3
00206     {
00207         18,
00208         -3,
```

```

00209     }, // Plane 3
00210     {
00211         19,
00212         3,
00213     }, // Plane 3
00214     {
00215         20,
00216         -5,
00217     }, // Plane 3
00218     {
00219         21,
00220         4,
00221     }, // Plane 3
00222     {
00223         22,
00224         -3,
00225     }, // Plane 3
00226     {
00227         23,
00228         3,
00229     }, // Plane 3
00230     {24, 2}}; // Plane 3
00231
00232 const std::vector<int32_t> GLONASS_GNAV_CRC_I_INDEX{9, 10, 12, 13, 15, 17, 19, 20, 22, 24, 26, 28, 30,
32, 34, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 66, 68, 70, 72, 74, 76, 78,
80, 82, 84};
00233 const std::vector<int32_t> GLONASS_GNAV_CRC_J_INDEX{9, 11, 12, 14, 15, 18, 19, 21, 22, 25, 26, 29, 30,
33, 34, 36, 37, 40, 41, 44, 45, 48, 49, 52, 53, 56, 57, 60, 61, 64, 65, 67, 68, 71, 72, 75, 76, 79,
80, 83, 84};
00234 const std::vector<int32_t> GLONASS_GNAV_CRC_K_INDEX{10, 11, 12, 16, 17, 18, 19, 23, 24, 25, 26, 31,
32, 33, 34, 38, 39, 40, 41, 46, 47, 48, 49, 54, 55, 56, 57, 62, 63, 64, 65, 69, 70, 71, 72, 77, 78,
79, 80, 85};
00235 const std::vector<int32_t> GLONASS_GNAV_CRC_L_INDEX{13, 14, 15, 16, 17, 18, 19, 27, 28, 29, 30, 31,
32, 33, 34, 42, 43, 44, 45, 46, 47, 48, 49, 58, 59, 60, 61, 62, 63, 64, 65, 73, 74, 75, 76, 77, 78,
79, 80};
00236 const std::vector<int32_t> GLONASS_GNAV_CRC_M_INDEX{20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31,
32, 33, 34, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 81, 82, 83, 84, 85};
00237 const std::vector<int32_t> GLONASS_GNAV_CRC_N_INDEX{35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46,
47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65};
00238 const std::vector<int32_t> GLONASS_GNAV_CRC_P_INDEX{66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77,
78, 79, 80, 81, 82, 83, 84, 85};
00239 const std::vector<int32_t> GLONASS_GNAV_CRC_Q_INDEX{9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46,
47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71,
72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85};
00240 const std::vector<int32_t> GLONASS_GNAV_ECC_LOCATOR{0, 0, 1, 8, 2, 9, 10, 11, 3, 12, 13, 14, 15, 16,
17, 18, 4, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 5, 34, 35, 36, 37, 38, 39, 40,
41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 6, 65,
66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84};
00241
00242 // GLONASS GNAV NAVIGATION MESSAGE STRUCTURE
00243 // NAVIGATION MESSAGE FIELDS POSITIONS
00244
00245 // FRAME 1-4
00246 // COMMON FIELDS
00247 const std::vector<std::pair<int32_t, int32_t>> STRING_ID({{2, 4}});
00248 const std::vector<std::pair<int32_t, int32_t>> KX({{78, 8}});
00249 // STRING 1
00250 const std::vector<std::pair<int32_t, int32_t>> P1({{8, 2}});
00251 const std::vector<std::pair<int32_t, int32_t>> T_K_HR({{10, 5}});
00252 const std::vector<std::pair<int32_t, int32_t>> T_K_MIN({{15, 6}});
00253 const std::vector<std::pair<int32_t, int32_t>> T_K_SEC({{21, 1}});
00254 const std::vector<std::pair<int32_t, int32_t>> X_N_DOT({{22, 24}});
00255 const std::vector<std::pair<int32_t, int32_t>> X_N_DOT_DOT({{46, 5}});
00256 const std::vector<std::pair<int32_t, int32_t>> X_N({{51, 27}});
00257
00258 // STRING 2
00259 const std::vector<std::pair<int32_t, int32_t>> B_N({{6, 3}});
00260 const std::vector<std::pair<int32_t, int32_t>> P2({{9, 1}});
00261 const std::vector<std::pair<int32_t, int32_t>> T_B({{10, 7}});
00262 const std::vector<std::pair<int32_t, int32_t>> Y_N_DOT({{22, 24}});
00263 const std::vector<std::pair<int32_t, int32_t>> Y_N_DOT_DOT({{46, 5}});
00264 const std::vector<std::pair<int32_t, int32_t>> Y_N({{51, 27}});
00265
00266 // STRING 3
00267 const std::vector<std::pair<int32_t, int32_t>> P3({{6, 1}});
00268 const std::vector<std::pair<int32_t, int32_t>> GAMMA_N({{7, 11}});
00269 const std::vector<std::pair<int32_t, int32_t>> P({{19, 2}});
00270 const std::vector<std::pair<int32_t, int32_t>> EPH_L_N({{21, 1}});
00271 const std::vector<std::pair<int32_t, int32_t>> Z_N_DOT({{22, 24}});
00272 const std::vector<std::pair<int32_t, int32_t>> Z_N_DOT_DOT({{46, 5}});
00273 const std::vector<std::pair<int32_t, int32_t>> Z_N({{51, 27}});
00274
00275 // STRING 4
00276 const std::vector<std::pair<int32_t, int32_t>> TAU_N({{6, 22}});
00277 const std::vector<std::pair<int32_t, int32_t>> DELTA_TAU_N({{28, 5}});
00278 const std::vector<std::pair<int32_t, int32_t>> E_N({{33, 5}});

```

```

00279 const std::vector<std::pair<int32_t, int32_t> P4({{52, 1}});
00280 const std::vector<std::pair<int32_t, int32_t> F_T({{53, 4}});
00281 const std::vector<std::pair<int32_t, int32_t> N_T({{60, 11}});
00282 const std::vector<std::pair<int32_t, int32_t> N({{71, 5}});
00283 const std::vector<std::pair<int32_t, int32_t> M({{76, 2}});
00284
00285 // STRING 5
00286 const std::vector<std::pair<int32_t, int32_t> DAY_NUMBER_A({{6, 11}});
00287 const std::vector<std::pair<int32_t, int32_t> TAU_C({{17, 32}});
00288 const std::vector<std::pair<int32_t, int32_t> N_4({{50, 5}});
00289 const std::vector<std::pair<int32_t, int32_t> TAU_GPS({{55, 22}});
00290 const std::vector<std::pair<int32_t, int32_t> ALM_L_N({{77, 1}});
00291
00292 // STRING 6, 8, 10, 12, 14
00293 const std::vector<std::pair<int32_t, int32_t> C_N({{6, 1}});
00294 const std::vector<std::pair<int32_t, int32_t> M_N_A({{7, 2}});
00295 const std::vector<std::pair<int32_t, int32_t> N_A({{9, 5}});
00296 const std::vector<std::pair<int32_t, int32_t> TAU_N_A({{14, 10}});
00297 const std::vector<std::pair<int32_t, int32_t> LAMBDA_N_A({{24, 21}});
00298 const std::vector<std::pair<int32_t, int32_t> DELTA_T_N_A({{45, 18}});
00299 const std::vector<std::pair<int32_t, int32_t> EPSILON_N_A({{63, 15}});
00300
00301 // STRING 7, 9, 11, 13, 15
00302 const std::vector<std::pair<int32_t, int32_t> OMEGA_N_A({{6, 16}});
00303 const std::vector<std::pair<int32_t, int32_t> T_LAMBDA_N_A({{22, 21}});
00304 const std::vector<std::pair<int32_t, int32_t> DELTA_T_N_A({{43, 22}});
00305 const std::vector<std::pair<int32_t, int32_t> DELTA_T_DOT_N_A({{65, 7}});
00306 const std::vector<std::pair<int32_t, int32_t> H_N_A({{72, 5}});
00307
00308 // STRING 14 FRAME 5
00309 const std::vector<std::pair<int32_t, int32_t> B1({{6, 11}});
00310 const std::vector<std::pair<int32_t, int32_t> B2({{17, 10}});
00311
00312
00313 /** \} */
00314 /** \} */
00315 #endif // GNSS_SDR_GLO_NASS_L1_L2_CA_H

```

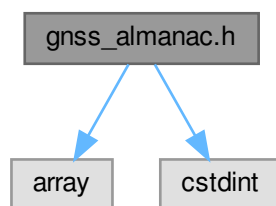
## 11.741 gnss\_almanac.h File Reference

Base class for GNSS almanac storage.

```
#include <array>
```

```
#include <cstdint>
```

Include dependency graph for gnss\_almanac.h:





```

00033 class Gnss_Almanac
00034 {
00035 public:
00036     /*!
00037      * Default constructor
00038      */
00039     Gnss_Almanac() = default;
00040
00041     /*!
00042      * \brief Computes prediction of the Doppler shift for a given time and receiver's position and
00043      * velocity.
00044      * 
$$f_d = - \mathbf{v} \cdot \frac{\mathbf{x}^T}{|\mathbf{x}|} \frac{f_L}{c}$$

00045      * \f]
00046      * where:
00047      * 
$$\mathbf{v} = \mathbf{v}_{sat} - \mathbf{v}_{rx}$$

00048      * 
$$\mathbf{x} = \mathbf{x}_{sat} - \mathbf{x}_{rx}$$

00049      * \f]
00050      * 
$$|\mathbf{x}| = \sqrt{\mathbf{x} \cdot \mathbf{x}}$$

00051      * 
$$|\mathbf{x}| = \sqrt{\mathbf{x} \cdot \mathbf{x}}$$

00052      * \f]
00053      * \left| \mathbf{x} \right| = \sqrt{\mathbf{x} \cdot \mathbf{x}}
00054      * \f]
00055      *
00056      * @param[in] rx_time_s Time of Week in seconds
00057      * @param[in] lat Receiver's latitude in degrees
00058      * @param[in] lon Receiver's longitude in degrees
00059      * @param[in] h Receiver's height in meters
00060      * @param[in] ve Receiver's velocity in the East direction [m/s]
00061      * @param[in] vn Receiver's velocity in the North direction [m/s]
00062      * @param[in] vu Receiver's velocity in the Up direction [m/s]
00063      * @param[in] band Signal band for which the Doppler will be computed
00064      * (1: L1 C/A, E1B, B1I; 2: L2C, B1I2; 3: B1I3; 5: L5/E5a; 6: E6B; 7: E5b; 8:
00065      * E5a+E5b)
00066      */
00067     double predicted_doppler(double rx_time_s,
00068         double lat,
00069         double lon,
00070         double h,
00071         double ve,
00072         double vn,
00073         double vu,
00074         int band) const;
00075
00076     /*!
00077      * \brief Computes satellite Position and Velocity, in ECEF, for a given time (expressed in
00078      * seconds of week)
00079      */
00079     void satellitePosVelComputation(double transmitTime, std::array<double, 7>& pos_vel_dtr) const;
00080
00081     uint32_t PRN{}; //!< SV PRN NUMBER
00082     double delta_i{}; //!< Inclination Angle at Reference Time (relative to i_0 = 0.30 semi-circles)
00083     int32_t toa{}; //!< Almanac data reference time of week [s]
00084     int32_t WNa{}; //!< Almanac week number
00085     double M_0{}; //!< Mean Anomaly at Reference Time [semi-circles]
00086     double ecc{}; //!< Eccentricity [dimensionless]
00087     double sqrtA{}; //!< Square Root of the Semi-Major Axis [sqrt(m)]
00088     double OMEGA_0{}; //!< Longitude of Ascending Node of Orbit Plane at Weekly Epoch [semi-circles]
00089     double omega{}; //!< Argument of Perigee [semi-circles]
00090     double OMEGAdot{}; //!< Rate of Right Ascension [semi-circles/s]
00091     double af0{}; //!< Coefficient 0 of code phase offset model [s]
00092     double af1{}; //!< Coefficient 1 of code phase offset model [s/s]
00093
00094 protected:
00095     char System{}; //!< Character ID of the GNSS system. 'G': GPS. 'E': Galileo. 'B': BeiDou
00096 private:
00097     double check_t(double time) const;
00098 };
00099
00100
00101 /** \} */
00102 /** \} */
00103 #endif // GNSS_SDR_GNSS_ALMANAC_H

```

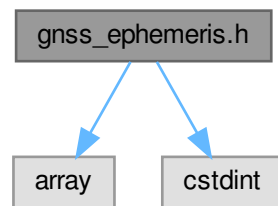
## 11.743 gnss\_ephemeris.h File Reference

Base class for GNSS Ephemeris.

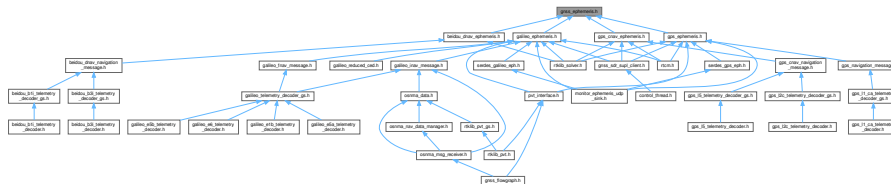
```
#include <array>
```

```
#include <cstdint>
```

Include dependency graph for `gnss_ephemeris.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Gnss\\_Ephemeris](#)  
Base class for GNSS ephemeris storage.

## 11.743.1 Detailed Description

Base class for GNSS Ephemeris.

### Author

Carles Fernandez, 2021. [cfernandez\(at\)cttc.es](mailto:cfernandez@cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gnss\\_ephemeris.h](#).

---

## 11.744 gnss\_ephemeris.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file gnss_ephemeris.h
00003  * \brief Base class for GNSS Ephemeris
00004  * \author Carles Fernandez, 2021. cfernandez(at)cttc.es
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
  
```

```

00018
00019 #ifndef GNSS_SDR_GNSS_EPHEMERIS_H
00020 #define GNSS_SDR_GNSS_EPHEMERIS_H
00021
00022 #include <array>
00023 #include <stdint>
00024
00025 /*!
00026 * \brief Base class for GNSS ephemeris storage
00027 */
00028 class Gnss_Ephemeris
00029 {
00030 public:
00031     Gnss_Ephemeris() = default;
00032
00033     /*!
00034     * \brief Sets (\a satClkDrift) and (\a dtr), and returns the clock drift in
00035     * seconds according to the User Algorithm for SV Clock Correction
00036     * (IS-GPS-200M, 20.3.3.3.1, and Galileo OS SIS ICD, 5.1.4).
00037     */
00038     double sv_clock_drift(double transmitTime);
00039
00040     /*!
00041     * \brief Computes prediction of the Doppler shift for a given time and receiver's position and
00042     * velocity.
00043     * \f[
00044     * f_{d} = - \mathbf{v} \cdot \frac{\mathbf{x}^T}{\|\mathbf{x}\|} \frac{f_L}{c}
00045     * \text{where:}
00046     * \f[
00047     * \mathbf{v} = \mathbf{v}_{\text{sat}} - \mathbf{v}_{\text{rx}}
00048     * \f[
00049     * \mathbf{x} = \mathbf{x}_{\text{sat}} - \mathbf{x}_{\text{rx}}
00050     * \f[
00051     * \mathbf{x} = \mathbf{x}_{\text{sat}} - \mathbf{x}_{\text{rx}}
00052     * \f[
00053     * \|\mathbf{x}\| = \sqrt{\mathbf{x} \cdot \mathbf{x}}
00054     * \f[
00055     *
00056     * @param[in] rx_time_s Time of Week in seconds
00057     * @param[in] lat Receiver's latitude in degrees
00058     * @param[in] lon Receiver's longitude in degrees
00059     * @param[in] h Receiver's height in meters
00060     * @param[in] ve Receiver's velocity in the East direction [m/s]
00061     * @param[in] vn Receiver's velocity in the North direction [m/s]
00062     * @param[in] vu Receiver's velocity in the Up direction [m/s]
00063     * @param[in] band Signal band for which the Doppler will be computed
00064     * (1: L1 C/A, E1B, B1i; 2: L2C, B1i; 3: B1i; 5: L5/E5a; 6: E6B; 7: E5b; 8:
00065     * E5a+E5b)
00066     */
00066     double predicted_doppler(double rx_time_s, double lat, double lon, double h, double ve, double vn,
00067     double vu, int band) const;
00068
00068     void satellitePosition(double transmitTime); //!< Computes the ECEF SV coordinates and ECEF
00069     velocity
00069
00070     uint32_t PRN{}; //!< SV ID
00071     double M_0{}; //!< Mean anomaly at reference time [rad]
00072     double delta_n{}; //!< Mean motion difference from computed value [rad/sec]
00073     double ecc{}; //!< Eccentricity
00074     double sqrtA{}; //!< Square root of the semi-major axis [meters^1/2]
00075     double OMEGA_0{}; //!< Longitude of ascending node of orbital plane at weekly epoch [rad]
00076     double i_0{}; //!< Inclination angle at reference time [rad]
00077     double omega{}; //!< Argument of perigee [rad]
00078     double OMEGAAdot{}; //!< Rate of right ascension [rad/sec]
00079     double idot{}; //!< Rate of inclination angle [rad/sec]
00080     double Cuc{}; //!< Amplitude of the cosine harmonic correction term to the argument of
00081     latitude [rad]
00081     double Cus{}; //!< Amplitude of the sine harmonic correction term to the argument of
00082     latitude [rad]
00082     double Crc{}; //!< Amplitude of the cosine harmonic correction term to the orbit radius
00083     [meters]
00083     double Crs{}; //!< Amplitude of the sine harmonic correction term to the orbit radius
00084     [meters]
00084     double Cic{}; //!< Amplitude of the cosine harmonic correction term to the angle of
00085     inclination [rad]
00085     double Cis{}; //!< Amplitude of the sine harmonic correction term to the angle of
00086     inclination [rad]
00086     int32_t toe{}; //!< Ephemeris reference time [s]
00087
00088     // Clock correction parameters
00089     int32_t toc{}; //!< Clock correction data reference Time of Week [sec]
00090     double af0{}; //!< SV clock bias correction coefficient [s]
00091     double af1{}; //!< SV clock drift correction coefficient [s/s]
00092     double af2{}; //!< SV clock drift rate correction coefficient [s/s^2]
00093
00094     double satClkDrift{}; //!< SV clock drift

```





- L5/E5a frequency (Hz)*
- constexpr double [FREQ6](#) = 1.27875e9
- E6/LEX frequency (Hz)*
- constexpr double [FREQ7](#) = 1.20714e9
- E5b frequency (Hz)*
- constexpr double [FREQ8](#) = 1.191795e9
- E5a+b frequency (Hz)*
- constexpr double [FREQ9](#) = 2.492028e9
- S frequency (Hz)*
- constexpr double [FREQ1\\_GLO](#) = 1.60200e9
- GLONASS G1 base frequency (Hz)*
- constexpr double [DFRQ1\\_GLO](#) = 0.56250e6
- GLONASS G1 bias frequency (Hz/n)*
- constexpr double [FREQ2\\_GLO](#) = 1.24600e9
- GLONASS G2 base frequency (Hz)*
- constexpr double [DFRQ2\\_GLO](#) = 0.43750e6
- GLONASS G2 bias frequency (Hz/n)*
- constexpr double [FREQ3\\_GLO](#) = 1.202025e9
- GLONASS G3 frequency (Hz)*
- constexpr double [FREQ1\\_BDS](#) = 1.561098e9
- BeiDou B1 frequency (Hz)*
- constexpr double [FREQ2\\_BDS](#) = 1.20714e9
- BeiDou B2 frequency (Hz)*
- constexpr double [FREQ3\\_BDS](#) = 1.26852e9
- BeiDou B3 frequency (Hz)*
- const std::unordered\_map< std::string, double > [SIGNAL\\_FREQ\\_MAP](#)

### 11.745.1 Detailed Description

GNSS Frequencies.

Author

Carles Fernandez, 2017. [cfernandez\(at\)cttc.es](mailto:cfernandez(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gnss\\_frequencies.h](#).

---

## 11.746 gnss\_frequencies.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file gnss_frequencies.h
00003  * \brief GNSS Frequencies
00004  * \author Carles Fernandez, 2017. cfernandez(at)cttc.es
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018
00019 #ifndef GNSS_SDR_GNSS_FREQUENCIES_H
00020 #define GNSS_SDR_GNSS_FREQUENCIES_H

```

```

00021
00022 #include <string>
00023 #include <unordered_map>
00024
00025 /** \addtogroup Core
00026  * \{ */
00027 /** \addtogroup System_Parameters
00028  * \{ */
00029
00030
00031 constexpr double FREQ1 = 1.57542e9;      //!< L1/E1 frequency (Hz)
00032 constexpr double FREQ2 = 1.22760e9;      //!< L2 frequency (Hz)
00033 constexpr double FREQ5 = 1.17645e9;      //!< L5/E5a frequency (Hz)
00034 constexpr double FREQ6 = 1.27875e9;      //!< E6/LEX frequency (Hz)
00035 constexpr double FREQ7 = 1.20714e9;      //!< E5b frequency (Hz)
00036 constexpr double FREQ8 = 1.191795e9;     //!< E5a+b frequency (Hz)
00037 constexpr double FREQ9 = 2.492028e9;     //!< S frequency (Hz)
00038 constexpr double FREQ1_GLO = 1.60200e9;  //!< GLONASS G1 base frequency (Hz)
00039 constexpr double DFRQ1_GLO = 0.56250e6;  //!< GLONASS G1 bias frequency (Hz/n)
00040 constexpr double FREQ2_GLO = 1.24600e9;  //!< GLONASS G2 base frequency (Hz)
00041 constexpr double DFRQ2_GLO = 0.43750e6;  //!< GLONASS G2 bias frequency (Hz/n)
00042 constexpr double FREQ3_GLO = 1.202025e9; //!< GLONASS G3 frequency (Hz)
00043 constexpr double FREQ1_BDS = 1.561098e9; //!< BeiDou B1 frequency (Hz)
00044 constexpr double FREQ2_BDS = 1.20714e9;  //!< BeiDou B2 frequency (Hz)
00045 constexpr double FREQ3_BDS = 1.26852e9;  //!< BeiDou B3 frequency (Hz)
00046
00047 const std::unordered_map<std::string, double> SIGNAL_FREQ_MAP = {
00048     {"1C", FREQ1},
00049     {"2S", FREQ2},
00050     {"L5", FREQ5},
00051     {"1B", FREQ1},
00052     {"5X", FREQ5},
00053     {"E6", FREQ6},
00054     {"7X", FREQ7},
00055     {"1G", FREQ1_GLO},
00056     {"2G", FREQ2_GLO},
00057     {"B1", FREQ1_BDS},
00058     {"B2", FREQ2_BDS},
00059     {"B3", FREQ3_BDS},
00060 };
00061
00062
00063 /** \} */
00064 /** \} */
00065 #endif // GNSS_SDR_GNSS_FREQUENCIES_H

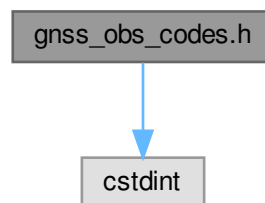
```

## 11.747 gnss\_obs\_codes.h File Reference

GNSS Observable codes.

```
#include <cstdint>
```

Include dependency graph for gnss\_obs\_codes.h:



[illegible]

- `constexpr uint32_t CODE_NONE = 0`  
*obs code: none or unknown*
- `constexpr uint32_t CODE_L1C = 1`  
*obs code: L1C/A, G1C/A, E1C (GPS, GLO, GAL, QZS, SBS)*
- `constexpr uint32_t CODE_L1P = 2`  
*obs code: L1P, G1P (GPS, GLO)*
- `constexpr uint32_t CODE_L1W = 3`  
*obs code: L1 Z-track (GPS)*
- `constexpr uint32_t CODE_L1Y = 4`  
*obs code: L1Y (GPS)*
- `constexpr uint32_t CODE_L1M = 5`  
*obs code: L1M (GPS)*
- `constexpr uint32_t CODE_L1N = 6`  
*obs code: L1codeless (GPS)*
- `constexpr uint32_t CODE_L1S = 7`  
*obs code: L1C(D) (GPS, QZS)*
- `constexpr uint32_t CODE_L1L = 8`  
*obs code: L1C(P) (GPS, QZS)*
- `constexpr uint32_t CODE_L1E = 9`  
*(not used)*
- `constexpr uint32_t CODE_L1A = 10`  
*obs code: E1A (GAL)*
- `constexpr uint32_t CODE_L1B = 11`  
*obs code: E1B (GAL)*
- `constexpr uint32_t CODE_L1X = 12`  
*obs code: E1B+C, L1C(D+P) (GAL, QZS)*
- `constexpr uint32_t CODE_L1Z = 13`  
*obs code: E1A+B+C, L1SAIF (GAL, QZS)*
- `constexpr uint32_t CODE_L2C = 14`  
*obs code: L2C/A, G1C/A (GPS, GLO)*
- `constexpr uint32_t CODE_L2D = 15`  
*obs code: L2 L1C/A-(P2-P1) (GPS)*
- `constexpr uint32_t CODE_L2S = 16`  
*obs code: L2C(M) (GPS, QZS)*
- `constexpr uint32_t CODE_L2L = 17`  
*obs code: L2C(L) (GPS, QZS)*
- `constexpr uint32_t CODE_L2X = 18`  
*obs code: L2C(M+L), B1I+Q (GPS, QZS, BDS)*
- `constexpr uint32_t CODE_L2P = 19`  
*obs code: L2P, G2P (GPS, GLO)*
- `constexpr uint32_t CODE_L2W = 20`  
*obs code: L2 Z-track (GPS)*

- constexpr uint32\_t [CODE\\_L2Y](#) = 21  
*obs code: L2Y (GPS)*
- constexpr uint32\_t [CODE\\_L2M](#) = 22  
*obs code: L2M (GPS)*
- constexpr uint32\_t [CODE\\_L2N](#) = 23  
*obs code: L2codeless (GPS)*
- constexpr uint32\_t [CODE\\_L5I](#) = 24  
*obs code: L5/E5aI (GPS,GAL,QZS,SBS)*
- constexpr uint32\_t [CODE\\_L5Q](#) = 25  
*obs code: L5/E5aQ (GPS,GAL,QZS,SBS)*
- constexpr uint32\_t [CODE\\_L5X](#) = 26  
*obs code: L5/E5aI+Q/L5B+C (GPS,GAL,QZS,IRN,SBS)*
- constexpr uint32\_t [CODE\\_L7I](#) = 27  
*obs code: E5bI,B2I (GAL,BDS)*
- constexpr uint32\_t [CODE\\_L7Q](#) = 28  
*obs code: E5bQ,B2Q (GAL,BDS)*
- constexpr uint32\_t [CODE\\_L7X](#) = 29  
*obs code: E5bI+Q,B2I+Q (GAL,BDS)*
- constexpr uint32\_t [CODE\\_L6A](#) = 30  
*obs code: E6A (GAL)*
- constexpr uint32\_t [CODE\\_L6B](#) = 31  
*obs code: E6B (GAL)*
- constexpr uint32\_t [CODE\\_L6C](#) = 32  
*obs code: E6C (GAL)*
- constexpr uint32\_t [CODE\\_L6X](#) = 33  
*obs code: E6B+C,LEXS+L,B3I+Q (GAL,QZS,BDS)*
- constexpr uint32\_t [CODE\\_L6Z](#) = 34  
*obs code: E6A+B+C (GAL)*
- constexpr uint32\_t [CODE\\_L6S](#) = 35  
*obs code: LEXS (QZS)*
- constexpr uint32\_t [CODE\\_L6L](#) = 36  
*obs code: LEXL (QZS)*
- constexpr uint32\_t [CODE\\_L8I](#) = 37  
*obs code: E5(a+b)I (GAL)*
- constexpr uint32\_t [CODE\\_L8Q](#) = 38  
*obs code: E5(a+b)Q (GAL)*
- constexpr uint32\_t [CODE\\_L8X](#) = 39  
*obs code: E5(a+b)I+Q (GAL)*
- constexpr uint32\_t [CODE\\_L2I](#) = 40  
*obs code: B1I (BDS)*
- constexpr uint32\_t [CODE\\_L2Q](#) = 41  
*obs code: B1Q (BDS)*
- constexpr uint32\_t [CODE\\_L6I](#) = 42  
*obs code: B3I (BDS)*
- constexpr uint32\_t [CODE\\_L6Q](#) = 43  
*obs code: B3Q (BDS)*
- constexpr uint32\_t [CODE\\_L3I](#) = 44  
*obs code: G3I (GLO)*
- constexpr uint32\_t [CODE\\_L3Q](#) = 45  
*obs code: G3Q (GLO)*
- constexpr uint32\_t [CODE\\_L3X](#) = 46

- obs code: G3I+Q (GLO)*
- constexpr uint32\_t `CODE_L1I` = 47
- obs code: B1I (BDS)*
- constexpr uint32\_t `CODE_L1Q` = 48
- obs code: B1Q (BDS)*
- constexpr uint32\_t `CODE_L5A` = 49
- obs code: L5A SPS (IRN)*
- constexpr uint32\_t `CODE_L5B` = 50
- obs code: L5B RS(D) (IRN)*
- constexpr uint32\_t `CODE_L5C` = 51
- obs code: L5C RS(P) (IRN)*
- constexpr uint32\_t `CODE_L9A` = 52
- obs code: SA SPS (IRN)*
- constexpr uint32\_t `CODE_L9B` = 53
- obs code: SB RS(D) (IRN)*
- constexpr uint32\_t `CODE_L9C` = 54
- obs code: SC RS(P) (IRN)*
- constexpr uint32\_t `CODE_L9X` = 55
- obs code: SB+C (IRN)*
- constexpr int32\_t `MAXCODE` = 55
- max number of obs code*

### 11.747.1 Detailed Description

GNSS Observable codes.

Author

Carles Fernandez, 2017. cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gnss\\_obs\\_codes.h](#).

---

## 11.748 gnss\_obs\_codes.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file gnss_obs_codes.h
00003  * \brief GNSS Observable codes
00004  * \author Carles Fernandez, 2017. cfernandez(at)cttc.es
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018
00019 #ifndef GNSS_SDR_GNSS_OBS_CODES_H
00020 #define GNSS_SDR_GNSS_OBS_CODES_H
00021
00022 #include <stdint>
00023
00024 /** \addtogroup Core
00025  * \{ */
00026 /** \addtogroup System_Parameters
00027  * \{ */
00028
00029
```

```

00030 constexpr uint32_t CODE_NONE = 0; //!< obs code: none or unknown
00031 constexpr uint32_t CODE_L1C = 1; //!< obs code: L1C/A, G1C/A, E1C (GPS, GLO, GAL, QZS, SBS)
00032 constexpr uint32_t CODE_L1P = 2; //!< obs code: L1P, G1P (GPS, GLO)
00033 constexpr uint32_t CODE_L1W = 3; //!< obs code: L1 Z-track (GPS)
00034 constexpr uint32_t CODE_L1Y = 4; //!< obs code: L1Y (GPS)
00035 constexpr uint32_t CODE_L1M = 5; //!< obs code: L1M (GPS)
00036 constexpr uint32_t CODE_L1N = 6; //!< obs code: L1codeless (GPS)
00037 constexpr uint32_t CODE_L1S = 7; //!< obs code: L1C(D) (GPS, QZS)
00038 constexpr uint32_t CODE_L1L = 8; //!< obs code: L1C(P) (GPS, QZS)
00039 constexpr uint32_t CODE_L1E = 9; //!< (not used)
00040 constexpr uint32_t CODE_L1A = 10; //!< obs code: E1A (GAL)
00041 constexpr uint32_t CODE_L1B = 11; //!< obs code: E1B (GAL)
00042 constexpr uint32_t CODE_L1X = 12; //!< obs code: E1B+C, L1C(D+P) (GAL, QZS)
00043 constexpr uint32_t CODE_L1Z = 13; //!< obs code: E1A+B+C, L1SAIF (GAL, QZS)
00044 constexpr uint32_t CODE_L2C = 14; //!< obs code: L2C/A, G1C/A (GPS, GLO)
00045 constexpr uint32_t CODE_L2D = 15; //!< obs code: L2 L1C/A-(P2-P1) (GPS)
00046 constexpr uint32_t CODE_L2S = 16; //!< obs code: L2C(M) (GPS, QZS)
00047 constexpr uint32_t CODE_L2L = 17; //!< obs code: L2C(L) (GPS, QZS)
00048 constexpr uint32_t CODE_L2X = 18; //!< obs code: L2C(M+L), B1I+Q (GPS, QZS, BDS)
00049 constexpr uint32_t CODE_L2P = 19; //!< obs code: L2P, G2P (GPS, GLO)
00050 constexpr uint32_t CODE_L2W = 20; //!< obs code: L2 Z-track (GPS)
00051 constexpr uint32_t CODE_L2Y = 21; //!< obs code: L2Y (GPS)
00052 constexpr uint32_t CODE_L2M = 22; //!< obs code: L2M (GPS)
00053 constexpr uint32_t CODE_L2N = 23; //!< obs code: L2codeless (GPS)
00054 constexpr uint32_t CODE_L5I = 24; //!< obs code: L5/E5aI (GPS, GAL, QZS, SBS)
00055 constexpr uint32_t CODE_L5Q = 25; //!< obs code: L5/E5aQ (GPS, GAL, QZS, SBS)
00056 constexpr uint32_t CODE_L5X = 26; //!< obs code: L5/E5aI+Q/L5B+C (GPS, GAL, QZS, IRN, SBS)
00057 constexpr uint32_t CODE_L7I = 27; //!< obs code: E5bI, B2I (GAL, BDS)
00058 constexpr uint32_t CODE_L7Q = 28; //!< obs code: E5bQ, B2Q (GAL, BDS)
00059 constexpr uint32_t CODE_L7X = 29; //!< obs code: E5bI+Q, B2I+Q (GAL, BDS)
00060 constexpr uint32_t CODE_L6A = 30; //!< obs code: E6A (GAL)
00061 constexpr uint32_t CODE_L6B = 31; //!< obs code: E6B (GAL)
00062 constexpr uint32_t CODE_L6C = 32; //!< obs code: E6C (GAL)
00063 constexpr uint32_t CODE_L6X = 33; //!< obs code: E6B+C, LEXS+L, B3I+Q (GAL, QZS, BDS)
00064 constexpr uint32_t CODE_L6Z = 34; //!< obs code: E6A+B+C (GAL)
00065 constexpr uint32_t CODE_L6S = 35; //!< obs code: LEXS (QZS)
00066 constexpr uint32_t CODE_L6L = 36; //!< obs code: LEXL (QZS)
00067 constexpr uint32_t CODE_L8I = 37; //!< obs code: E5(a+b)I (GAL)
00068 constexpr uint32_t CODE_L8Q = 38; //!< obs code: E5(a+b)Q (GAL)
00069 constexpr uint32_t CODE_L8X = 39; //!< obs code: E5(a+b)I+Q (GAL)
00070 constexpr uint32_t CODE_L2I = 40; //!< obs code: B1I (BDS)
00071 constexpr uint32_t CODE_L2Q = 41; //!< obs code: B1Q (BDS)
00072 constexpr uint32_t CODE_L6I = 42; //!< obs code: B3I (BDS)
00073 constexpr uint32_t CODE_L6Q = 43; //!< obs code: B3Q (BDS)
00074 constexpr uint32_t CODE_L3I = 44; //!< obs code: G3I (GLO)
00075 constexpr uint32_t CODE_L3Q = 45; //!< obs code: G3Q (GLO)
00076 constexpr uint32_t CODE_L3X = 46; //!< obs code: G3I+Q (GLO)
00077 constexpr uint32_t CODE_L1I = 47; //!< obs code: B1I (BDS)
00078 constexpr uint32_t CODE_L1Q = 48; //!< obs code: B1Q (BDS)
00079 constexpr uint32_t CODE_L5A = 49; //!< obs code: L5A SPS (IRN)
00080 constexpr uint32_t CODE_L5B = 50; //!< obs code: L5B RS(D) (IRN)
00081 constexpr uint32_t CODE_L5C = 51; //!< obs code: L5C RS(P) (IRN)
00082 constexpr uint32_t CODE_L9A = 52; //!< obs code: SA SPS (IRN)
00083 constexpr uint32_t CODE_L9B = 53; //!< obs code: SB RS(D) (IRN)
00084 constexpr uint32_t CODE_L9C = 54; //!< obs code: SC RS(P) (IRN)
00085 constexpr uint32_t CODE_L9X = 55; //!< obs code: SB+C (IRN)
00086 constexpr int32_t MAXCODE = 55; //!< max number of obs code
00087
00088
00089 /** \} */
00090 /** \} */
00091 #endif // GNSS_SDR_GNSS_OBS_CODES_H

```

## 11.749 gnss\_satellite.h File Reference

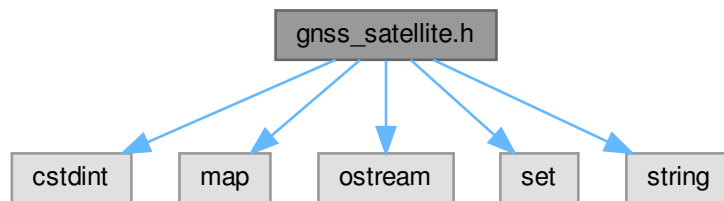
Interface of the [Gnss\\_Satellite](#) class.

```

#include <cstdint>
#include <map>
#include <ostream>
#include <set>
#include <string>

```

Include dependency graph for gnss\_satellite.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Gnss\\_Satellite](#)

*This class represents a GNSS satellite.*

## 11.749.1 Detailed Description

Interface of the [Gnss\\_Satellite](#) class.

### Author

Carles Fernandez-Prades, 2012. cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gnss\\_satellite.h](#).

---

## 11.750 gnss\_satellite.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file gnss_satellite.h
00003  * \brief Interface of the Gnss_Satellite class
00004  * \author Carles Fernandez-Prades, 2012. cfernandez(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_GNSS_SATELLITE_H
00019 #define GNSS_SDR_GNSS_SATELLITE_H
00020
00021 #include <cstdint>
00022 #include <map>
00023 #include <ostream>
  
```

```

00024 #include <set>
00025 #include <string>
00026
00027 /** \addtogroup Core
00028 * \{ */
00029 /** \addtogroup System_Parameters
00030 * \{ */
00031
00032
00033 /*!
00034 * \brief This class represents a GNSS satellite.
00035 *
00036 * It contains information about the space vehicles currently operational
00037 * of GPS, Glonass, SBAS and Galileo constellations.
00038 */
00039 class Gnss_Satellite
00040 {
00041 public:
00042     Gnss_Satellite() = default; //!< Default Constructor.
00043     Gnss_Satellite(const std::string& system_, uint32_t PRN_); //!< Concrete GNSS satellite
        Constructor.
00044     ~Gnss_Satellite() = default; //!< Default Destructor.
00045
00046     Gnss_Satellite(const Gnss_Satellite& other) noexcept; //!< Copy constructor
00047     Gnss_Satellite& operator=(const Gnss_Satellite&) noexcept; //!< Copy assignment operator
00048     Gnss_Satellite(Gnss_Satellite&& other) noexcept; //!< Move constructor
00049     Gnss_Satellite& operator=(Gnss_Satellite&& other) noexcept; //!< Move assignment operator
00050
00051     friend bool operator==(const Gnss_Satellite& /*sat1*/, const Gnss_Satellite& /*sat2*/); //!<
operator== for comparison
00052     friend std::ostream& operator<<(std::ostream& /*out*/, const Gnss_Satellite& /*sat*/); //!<
operator<< for pretty printing
00053
00054     void update_PRN(uint32_t PRN); //!< Updates the PRN Number
when information is decoded, only applies to GLONASS GNAV messages
00055     uint32_t get_PRN() const; //!< Gets satellite's PRN
00056     int32_t get_rf_link() const; //!< Gets the satellite's rf
link
00057     std::string get_system() const; //!< Gets the satellite system
{"GPS", "GLONASS", "SBAS", "Galileo", "Beidou"}
00058     std::string get_system_short() const; //!< Gets the satellite system
{"G", "R", "SBAS", "E", "C"}
00059     std::string get_block() const; //!< Gets the satellite block.
If GPS, returns {"IIA", "IIR", "IIR-M", "IIF"}
00060     std::string what_block(const std::string& system_, uint32_t PRN_); //!< Gets the block of a given
satellite
00061
00062 private:
00063     const std::set<std::string> system_set = {"GPS", "Glonass", "SBAS", "Galileo", "Beidou"};
00064     const std::map<std::string, std::string> satelliteSystem = {"GPS", "G"}, {"Glonass", "R"},
{"SBAS", "S"}, {"Galileo", "E"}, {"Beidou", "C"};
00065     void set_system(const std::string& system); // Sets the satellite system {"GPS", "GLONASS",
"SBAS", "Galileo", "Beidou"}.
00066     void set_PRN(uint32_t PRN); // Sets satellite's PRN
00067     void set_block(const std::string& system_, uint32_t PRN_);
00068     void reset();
00069     void set_rf_link(int32_t rf_link_);
00070     std::string system{};
00071     std::string block{};
00072     uint32_t PRN{};
00073     int32_t rf_link{};
00074 };
00075
00076
00077 /** \} */
00078 /** \} */
00079 #endif // GNSS_SDR_GNSS_SATELLITE_H

```

## 11.751 gnss\_signal.h File Reference

Implementation of the [Gnss\\_Signal](#) class.

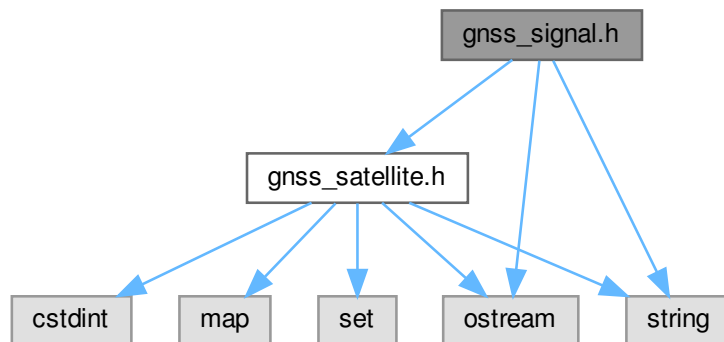
```

#include "gnss_satellite.h"
#include <ostream>
#include <string>

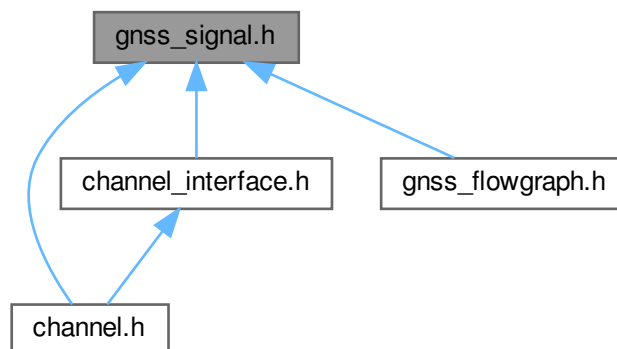
```



Include dependency graph for gnss\_signal.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Gnss\\_Signal](#)

*This class represents a GNSS signal.*

### 11.751.1 Detailed Description

Implementation of the [Gnss\\_Signal](#) class.

#### Author

Luis Esteve, 2012. [luis\(at\)epsilon-formacion.com](mailto:luis(at)epsilon-formacion.com) Javier Arribas, 2012. [jarribas\(at\)cttc.es](mailto:jarribas(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gnss\\_signal.h](#).

---

## 11.752 gnss\_signal.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file gnss_signal.h
00003  * \brief Implementation of the Gnss_Signal class
00004  * \author
00005  * Luis Esteve, 2012. luis(at)epsilon-formacion.com
00006  * Javier Arribas, 2012. jarribas(at)cttc.es
00007  *
00008  * -----
00009  *
00010  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00011  * This file is part of GNSS-SDR.
00012  *
00013  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00014  * SPDX-License-Identifier: GPL-3.0-or-later
00015  *
00016  * -----
00017  */
00018
00019 #ifndef GNSS_SDR_GNSS_SIGNAL_H
00020 #define GNSS_SDR_GNSS_SIGNAL_H
00021
00022 #include "gnss_satellite.h"
00023 #include <ostream>
00024 #include <string>
00025
00026 /** \addtogroup Core
00027  * \{ */
00028 /** \addtogroup System_Parameters
00029  * \{ */
00030
00031
00032  /*!
00033  * \brief This class represents a GNSS signal.
00034  *
00035  * It contains information about the space vehicle and the specific signal.
00036  */
00037  class Gnss_Signal
00038  {
00039  public:
00040      Gnss_Signal() = default;
00041      explicit Gnss_Signal(const std::string& signal_);
00042      Gnss_Signal(const Gnss_Satellite& satellite_, const std::string& signal_);
00043      ~Gnss_Signal() = default;
00044      std::string get_signal_str() const;    //!< Get the satellite signal {"1C" for GPS L1 C/A, "2S"
for GPS L2C (M), "L5" for GPS L5, "1G" for GLONASS L1 C/A, "1B" for Galileo E1B, "5X" for Galileo E5a.
00045      Gnss_Satellite get_satellite() const;    //!< Get the Gnss_Satellite associated to the signal
00046
00047      friend bool operator==(const Gnss_Signal& /*sig1*/, const Gnss_Signal& /*sig2*/);    //!<
operator== for comparison
00048      friend std::ostream& operator<<(std::ostream& /*out*/, const Gnss_Signal& /*sig*/);    //!< operator<<
for pretty printing
00049
00050  private:
00051      Gnss_Satellite satellite{};
00052      std::string signal{};
00053  };
00054
00055
00056  /** \} */
00057  /** \} */
00058 #endif // GNSS_SDR_GNSS_SIGNAL_H

```

## 11.753 gnss\_synchro.h File Reference

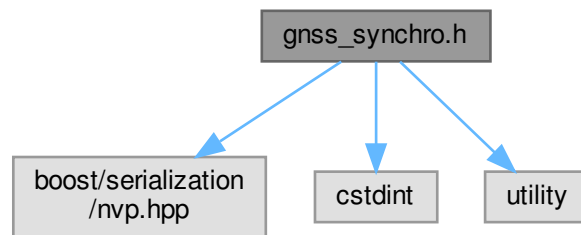
Interface of the [Gnss\\_Synchro](#) class.

```

#include <boost/serialization/nvp.hpp>
#include <cstdint>
#include <utility>

```

Include dependency graph for gnss\_synchro.h:



## Classes

- class [Gnss\\_Synchro](#)

*This is the class that contains the information that is shared by the processing blocks.*

### 11.753.1 Detailed Description

Interface of the [Gnss\\_Synchro](#) class.

#### Author

Luis Esteve, 2012. [luis\(at\)epsilon-formacion.com](mailto:luis(at)epsilon-formacion.com) Javier Arribas, 2012. [jarribas\(at\)cttc.es](mailto:jarribas(at)cttc.es) Álvaro Cebrián Juan, 2018. [acebrianjuan\(at\)gmail.com](mailto:acebrianjuan(at)gmail.com)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [gnss\\_synchro.h](#).

---

## 11.754 gnss\_synchro.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file gnss_synchro.h
00003  * \brief Interface of the Gnss_Synchro class
00004  * \author
00005  * Luis Esteve, 2012. luis\(at\)epsilon-formacion.com
00006  * Javier Arribas, 2012. jarribas\(at\)cttc.es
00007  * Álvaro Cebrián Juan, 2018. acebrianjuan\(at\)gmail.com
00008  *
00009  * -----
00010  *
00011  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00012  * This file is part of GNSS-SDR.
00013  *
00014  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00015  * SPDX-License-Identifier: GPL-3.0-or-later
00016  *
00017  * -----
00018  */
00019
00020 #ifndef GNSS_SDR_GNSS_SYNCHRO_H
00021 #define GNSS_SDR_GNSS_SYNCHRO_H
00022
00023 #include <boost/serialization/nvp.hpp>
00024 #include <cstdint>
00025 #include <utility>
00026
00027 /** \addtogroup Core
00028  * \{ */
00029 /** \addtogroup System_Parameters core_system_parameters
00030  * GNSS parameters
  
```

```

00031  * \{ */
00032
00033
00034  /*!
00035  * \brief This is the class that contains the information that is shared
00036  * by the processing blocks.
00037  */
00038  class Gnss_Synchro
00039  {
00040  public:
00041      Gnss_Synchro() = default;    //!< Default constructor
00042
00043      ~Gnss_Synchro() = default;  //!< Default destructor
00044
00045      // Satellite and signal info
00046      char System{};                //!< Set by Channel::set_signal(Gnss_Signal gnss_signal)
00047      char Signal[3]{};            //!< Set by Channel::set_signal(Gnss_Signal gnss_signal)
00048      uint32_t PRN{};              //!< Set by Channel::set_signal(Gnss_Signal gnss_signal)
00049      int32_t Channel_ID{};        //!< Set by Channel constructor
00050
00051      // Acquisition
00052      double Acq_delay_samples{};  //!< Set by Acquisition processing block
00053      double Acq_doppler_hz{};    //!< Set by Acquisition processing block
00054      uint64_t Acq_samplestamp_samples{}; //!< Set by Acquisition processing block
00055      uint32_t Acq_doppler_step{}; //!< Set by Acquisition processing block
00056
00057      // Tracking
00058      int64_t fs{};                //!< Set by Tracking processing block
00059      double Prompt_I{};           //!< Set by Tracking processing block
00060      double Prompt_Q{};           //!< Set by Tracking processing block
00061      double CN0_dB_hz{};          //!< Set by Tracking processing block
00062      double Carrier_Doppler_hz{}; //!< Set by Tracking processing block
00063      double Carrier_phase_rads{}; //!< Set by Tracking processing block
00064      double Code_phase_samples{}; //!< Set by Tracking processing block
00065      uint64_t Tracking_sample_counter{}; //!< Set by Tracking processing block
00066      int32_t correlation_length_ms{}; //!< Set by Tracking processing block
00067
00068      // Telemetry Decoder
00069      uint32_t TOW_at_current_symbol_ms{}; //!< Set by Telemetry Decoder processing block
00070
00071      // Observables
00072      double Pseudorange_m{};      //!< Set by Observables processing block
00073      double RX_time{};            //!< Set by Observables processing block
00074      double interp_TOW_ms{};      //!< Set by Observables processing block
00075
00076      // Flags
00077      bool Flag_valid_acquisition{}; //!< Set by Acquisition processing block
00078      bool Flag_valid_symbol_output{}; //!< Set by Tracking processing block
00079      bool Flag_valid_word{};        //!< Set by Telemetry Decoder processing block
00080      bool Flag_valid_pseudorange{}; //!< Set by Observables processing block
00081      bool Flag_PLL_180_deg_phase_locked{}; //!< Set by Telemetry Decoder processing block
00082
00083      /// Copy constructor
00084      Gnss_Synchro(const Gnss_Synchro& other) noexcept = default;
00085
00086      /// Copy assignment operator
00087      Gnss_Synchro& operator=(const Gnss_Synchro& rhs) noexcept
00088      {
00089          // Only do assignment if RHS is a different object from this.
00090          if (this != &rhs)
00091          {
00092              this->System = rhs.System;
00093              this->Signal[0] = rhs.Signal[0];
00094              this->Signal[1] = rhs.Signal[1];
00095              this->Signal[2] = rhs.Signal[2];
00096              this->PRN = rhs.PRN;
00097              this->Channel_ID = rhs.Channel_ID;
00098              this->Acq_delay_samples = rhs.Acq_delay_samples;
00099              this->Acq_doppler_hz = rhs.Acq_doppler_hz;
00100              this->Acq_samplestamp_samples = rhs.Acq_samplestamp_samples;
00101              this->Acq_doppler_step = rhs.Acq_doppler_step;
00102              this->fs = rhs.fs;
00103              this->Prompt_I = rhs.Prompt_I;
00104              this->Prompt_Q = rhs.Prompt_Q;
00105              this->CN0_dB_hz = rhs.CN0_dB_hz;
00106              this->Carrier_Doppler_hz = rhs.Carrier_Doppler_hz;
00107              this->Carrier_phase_rads = rhs.Carrier_phase_rads;
00108              this->Code_phase_samples = rhs.Code_phase_samples;
00109              this->Tracking_sample_counter = rhs.Tracking_sample_counter;
00110              this->correlation_length_ms = rhs.correlation_length_ms;
00111              this->TOW_at_current_symbol_ms = rhs.TOW_at_current_symbol_ms;
00112              this->Pseudorange_m = rhs.Pseudorange_m;
00113              this->RX_time = rhs.RX_time;
00114              this->interp_TOW_ms = rhs.interp_TOW_ms;
00115              this->Flag_valid_acquisition = rhs.Flag_valid_acquisition;
00116              this->Flag_valid_symbol_output = rhs.Flag_valid_symbol_output;
00117              this->Flag_valid_word = rhs.Flag_valid_word;

```

```

00118         this->Flag_valid_pseudorange = rhs.Flag_valid_pseudorange;
00119         this->Flag_PLL180_deg_phase_locked = rhs.Flag_PLL180_deg_phase_locked;
00120     }
00121     return *this;
00122 };
00123
00124 /// Move constructor
00125 Gnss_Synchro(Gnss_Synchro&& other) noexcept = default;
00126
00127 /// Move assignment operator
00128 Gnss_Synchro& operator=(Gnss_Synchro&& other) noexcept
00129 {
00130     if (this != &other)
00131     {
00132         this->System = other.System;
00133         this->Signal[0] = other.Signal[0];
00134         this->Signal[1] = other.Signal[1];
00135         this->Signal[2] = other.Signal[2];
00136         this->PRN = other.PRN;
00137         this->Channel_ID = other.Channel_ID;
00138         this->Acq_delay_samples = other.Acq_delay_samples;
00139         this->Acq_doppler_hz = other.Acq_doppler_hz;
00140         this->Acq_samplestamp_samples = other.Acq_samplestamp_samples;
00141         this->Acq_doppler_step = other.Acq_doppler_step;
00142         this->fs = other.fs;
00143         this->Prompt_I = other.Prompt_I;
00144         this->Prompt_Q = other.Prompt_Q;
00145         this->CNO_dB_hz = other.CNO_dB_hz;
00146         this->Carrier_Doppler_hz = other.Carrier_Doppler_hz;
00147         this->Carrier_phase_rads = other.Carrier_phase_rads;
00148         this->Code_phase_samples = other.Code_phase_samples;
00149         this->Tracking_sample_counter = other.Tracking_sample_counter;
00150         this->correlation_length_ms = other.correlation_length_ms;
00151         this->TOW_at_current_symbol_ms = other.TOW_at_current_symbol_ms;
00152         this->Pseudorange_m = other.Pseudorange_m;
00153         this->RX_time = other.RX_time;
00154         this->interp_TOW_ms = other.interp_TOW_ms;
00155         this->Flag_valid_acquisition = other.Flag_valid_acquisition;
00156         this->Flag_valid_symbol_output = other.Flag_valid_symbol_output;
00157         this->Flag_valid_word = other.Flag_valid_word;
00158         this->Flag_valid_pseudorange = other.Flag_valid_pseudorange;
00159         this->Flag_PLL180_deg_phase_locked = other.Flag_PLL180_deg_phase_locked;
00160
00161         // Leave the source object in a valid but unspecified state
00162         other.Signal[0] = '\0';
00163         other.Signal[1] = '\0';
00164         other.Signal[2] = '\0';
00165         other.System = 0;
00166         other.PRN = 0;
00167         other.Channel_ID = 0;
00168         other.Acq_delay_samples = 0.0;
00169         other.Acq_doppler_hz = 0.0;
00170         other.Acq_samplestamp_samples = 0;
00171         other.Acq_doppler_step = 0;
00172         other.fs = 0;
00173         other.Prompt_I = 0.0;
00174         other.Prompt_Q = 0.0;
00175         other.CNO_dB_hz = 0.0;
00176         other.Carrier_Doppler_hz = 0.0;
00177         other.Carrier_phase_rads = 0.0;
00178         other.Code_phase_samples = 0.0;
00179         other.Tracking_sample_counter = 0;
00180         other.correlation_length_ms = 0;
00181         other.TOW_at_current_symbol_ms = 0;
00182         other.Pseudorange_m = 0.0;
00183         other.RX_time = 0.0;
00184         other.interp_TOW_ms = 0.0;
00185         other.Flag_valid_acquisition = false;
00186         other.Flag_valid_symbol_output = false;
00187         other.Flag_valid_word = false;
00188         other.Flag_valid_pseudorange = false;
00189         other.Flag_PLL180_deg_phase_locked = false;
00190     }
00191     return *this;
00192 };
00193
00194 /*!
00195 * \brief This member function serializes and restores
00196 * Gnss_Synchro objects from a byte stream.
00197 */
00198 template <class Archive>
00199 void serialize(Archive& ar, const unsigned int version)
00200 {
00201     if (version)
00202     {
00203     };
00204 }

```

```

00205         // Satellite and signal info
00206         ar& BOOST_SERIALIZATION_NVP(System);
00207         ar& BOOST_SERIALIZATION_NVP(Signal);
00208         ar& BOOST_SERIALIZATION_NVP(PRN);
00209         ar& BOOST_SERIALIZATION_NVP(Channel_ID);
00210         // Acquisition
00211         ar& BOOST_SERIALIZATION_NVP(Acq_delay_samples);
00212         ar& BOOST_SERIALIZATION_NVP(Acq_doppler_hz);
00213         ar& BOOST_SERIALIZATION_NVP(Acq_samplestamp_samples);
00214         ar& BOOST_SERIALIZATION_NVP(Acq_doppler_step);
00215         // Tracking
00216         ar& BOOST_SERIALIZATION_NVP(fs);
00217         ar& BOOST_SERIALIZATION_NVP(Prompt_I);
00218         ar& BOOST_SERIALIZATION_NVP(Prompt_Q);
00219         ar& BOOST_SERIALIZATION_NVP(CN0_dB_hz);
00220         ar& BOOST_SERIALIZATION_NVP(Carrier_Doppler_hz);
00221         ar& BOOST_SERIALIZATION_NVP(Carrier_phase_rads);
00222         ar& BOOST_SERIALIZATION_NVP(Code_phase_samples);
00223         ar& BOOST_SERIALIZATION_NVP(Tracking_sample_counter);
00224         ar& BOOST_SERIALIZATION_NVP(correlation_length_ms);
00225         // Telemetry Decoder
00226         ar& BOOST_SERIALIZATION_NVP(TOW_at_current_symbol_ms);
00227         // Observables
00228         ar& BOOST_SERIALIZATION_NVP(Pseudorange_m);
00229         ar& BOOST_SERIALIZATION_NVP(RX_time);
00230         ar& BOOST_SERIALIZATION_NVP(interp_TOW_ms);
00231         // Flags
00232         ar& BOOST_SERIALIZATION_NVP(Flag_valid_acquisition);
00233         ar& BOOST_SERIALIZATION_NVP(Flag_valid_symbol_output);
00234         ar& BOOST_SERIALIZATION_NVP(Flag_valid_word);
00235         ar& BOOST_SERIALIZATION_NVP(Flag_valid_pseudorange);
00236         ar& BOOST_SERIALIZATION_NVP(Flag_PLL_180_deg_phase_locked);
00237     }
00238 };
00239
00240
00241 /** \} */
00242 /** \} */
00243 #endif // GNSS_SDR_GNSS_SYNCRO_H

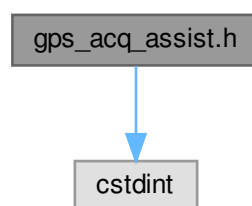
```

## 11.755 gps\_acq\_assist.h File Reference

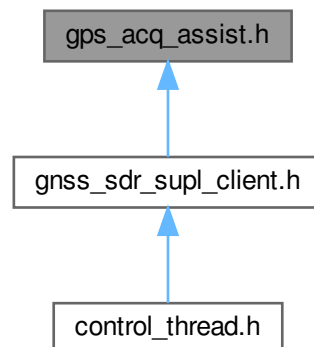
Interface of a GPS RRLL ACQUISITION ASSISTACE storage.

#include <cstdint>

Include dependency graph for gps\_acq\_assist.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Gps\\_Acq\\_Assist](#)

*This class is a storage for the GPS GSM RRLC acquisition assistance data as described in Digital cellular telecommunications system (Phase 2+); Location Services (LCS); Mobile Station (MS) - Serving Mobile Location Centre (SMLC) Radio Resource LCS Protocol (RRLP) (3GPP TS 44.031 version 5.12.0 Release 5)*

## 11.755.1 Detailed Description

Interface of a GPS RRLC ACQUISITION ASSISTANCE storage.

### Author

Javier Arribas, 2013. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gps\\_acq\\_assist.h](#).

---

## 11.756 gps\_acq\_assist.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file gps_acq_assist.h
00003  * \brief Interface of a GPS RRLC ACQUISITION ASSISTANCE storage
00004  * \author Javier Arribas, 2013. jarribas(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_GPS_ACQ_ASSIST_H
00019 #define GNSS_SDR_GPS_ACQ_ASSIST_H
00020
00021 #include <cstdint>
00022
00023 /** \addtogroup Core
  
```

```

00024  * \{ */
00025  /** \addtogroup System_Parameters
00026  * \{ */
00027
00028
00029  /*!
00030  * \brief This class is a storage for the GPS GSM RRLP acquisition assistance data as described in
00031  * Digital cellular telecommunications system (Phase 2+);
00032  * Location Services (LCS);
00033  * Mobile Station (MS) - Serving Mobile Location Centre (SMLC)
00034  * Radio Resource LCS Protocol (RRLP)
00035  * (3GPP TS 44.031 version 5.12.0 Release 5)
00036  */
00037  class Gps_Acq_Assist
00038  {
00039  public:
00040      /*!
00041       * Default constructor
00042       */
00043      Gps_Acq_Assist() = default;
00044
00045      uint32_t PRN{};          //!< SV PRN NUMBER
00046      double tow{};           //!< Time Of Week assigned to the acquisition data
00047      double Doppler0{};       //!< Doppler (0 order term) [Hz]
00048      double Doppler1{};       //!< Doppler (1 order term) [Hz]
00049      double dopplerUncertainty{}; //!< Doppler Uncertainty [Hz]
00050      double Code_Phase{};      //!< Code phase [chips]
00051      double Code_Phase_int{};  //!< Integer Code Phase [1 C/A code period]
00052      double GPS_Bit_Number{};  //!< GPS Bit Number
00053      double Code_Phase_window{}; //!< Code Phase search window [chips]
00054      double Azimuth{};         //!< Satellite Azimuth [deg]
00055      double Elevation{};       //!< Satellite Elevation [deg]
00056  };
00057
00058
00059  /** \} */
00060  /** \} */
00061  #endif // GNSS_SDR_GPS_ACQ_ASSIST_H

```

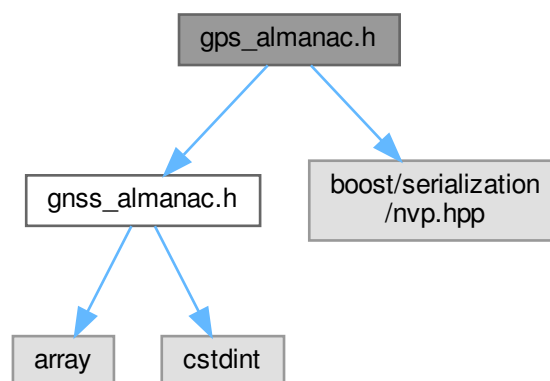
## 11.757 gps\_almanac.h File Reference

Interface of a GPS ALMANAC storage.

```
#include "gnss_almanac.h"
```

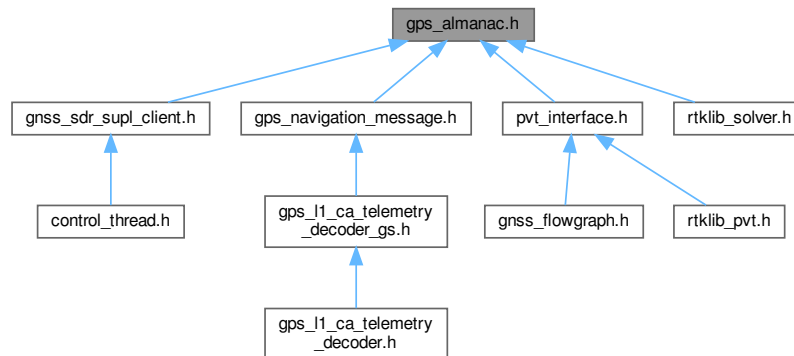
```
#include <boost/serialization/nvp.hpp>
```

Include dependency graph for gps\_almanac.h:





This graph shows which files directly or indirectly include this file:



## Classes

- class [Gps\\_Almanac](#)

*This class is a storage for the GPS SV ALMANAC data as described in IS-GPS-200M.*

## 11.757.1 Detailed Description

Interface of a GPS ALMANAC storage.

### Author

Javier Arribas, 2013. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gps\\_almanac.h](#).

---

## 11.758 gps\_almanac.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file gps_almanac.h
00003  * \brief Interface of a GPS ALMANAC storage
00004  * \author Javier Arribas, 2013. jarribas(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_GPS_ALMANAC_H
00019 #define GNSS_SDR_GPS_ALMANAC_H
00020
00021 #include "gnss_almanac.h"
00022 #include <boost/serialization/nvp.hpp>
00023
00024 /** \addtogroup Core
00025  * \{ */
00026 /** \addtogroup System_Parameters
00027  * \{ */
00028
00029

```

```

00030  /*!
00031  * \brief This class is a storage for the GPS SV ALMANAC data as described in IS-GPS-200M
00032  *
00033  * See https://www.gps.gov/technical/icwg/IS-GPS-200M.pdf Appendix II
00034  */
00035  class Gps_Almanac : public Gnss_Almanac
00036  {
00037  public:
00038      /*!
00039       * Default constructor
00040       */
00041      Gps_Almanac()
00042      {
00043          this->System = 'G';
00044      };
00045
00046      int32_t SV_health{}; //!< SV Health
00047      int32_t AS_status{}; //!< Anti-Spoofing Flags and SV Configuration
00048
00049      template <class Archive>
00050
00051      void serialize(Archive& ar, const unsigned int version)
00052      {
00053          if (version)
00054          {
00055              };
00056          ar& BOOST_SERIALIZATION_NVP (PRN);
00057          ar& BOOST_SERIALIZATION_NVP (delta_i);
00058          ar& BOOST_SERIALIZATION_NVP (toa);
00059          ar& BOOST_SERIALIZATION_NVP (WNa);
00060          ar& BOOST_SERIALIZATION_NVP (M_0);
00061          ar& BOOST_SERIALIZATION_NVP (ecc);
00062          ar& BOOST_SERIALIZATION_NVP (sqrtA);
00063          ar& BOOST_SERIALIZATION_NVP (OMEGA_0);
00064          ar& BOOST_SERIALIZATION_NVP (omega);
00065          ar& BOOST_SERIALIZATION_NVP (OMEGAdot);
00066          ar& BOOST_SERIALIZATION_NVP (af0);
00067          ar& BOOST_SERIALIZATION_NVP (af1);
00068          ar& BOOST_SERIALIZATION_NVP (SV_health);
00069          ar& BOOST_SERIALIZATION_NVP (AS_status);
00070      }
00071  };
00072
00073
00074  /** @} */
00075  /** @} */
00076  #endif // GNSS_SDR_GPS_ALMANAC_H

```

## 11.759 GPS\_CNAV.h File Reference

Defines parameters for GPS CNAV.

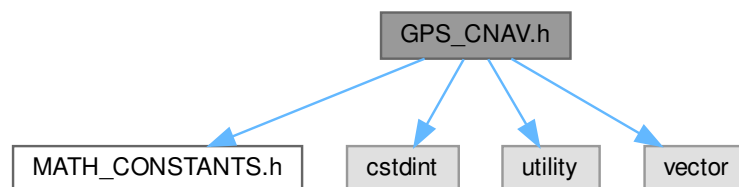
```
#include "MATH_CONSTANTS.h"
```

```
#include <cstdint>
```

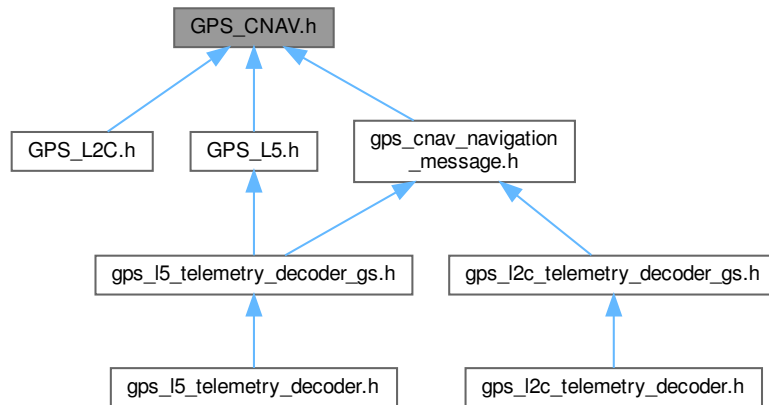
```
#include <utility>
```

```
#include <vector>
```

Include dependency graph for GPS\_CNAV.h:



This graph shows which files directly or indirectly include this file:



## Functions

- `const std::vector< std::pair< int32_t, int32_t > > CNAV_PRN` ({9, 6}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_MSG_TYPE` ({15, 6}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_TOW` ({21, 17}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_ALERT_FLAG` ({38, 1}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_WN` ({39, 13}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_HEALTH` ({52, 3}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_TOP1` ({55, 11}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_URA` ({66, 5}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_TOE1` ({71, 11}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_DELTA_A` ({82, 26}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_A_DOT` ({108, 25}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_DELTA_N0` ({133, 17}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_DELTA_N0_DOT` ({150, 23}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_M0` ({173, 33}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_E_ECCENTRICITY` ({206, 33}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_OMEGA` ({239, 33}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_INTEGRITY_FLAG` ({272, 1}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_L2_PHASING_FLAG` ({273, 1}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_TOE2` ({39, 11}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_OMEGA0` ({50, 33}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_I0` ({83, 33}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_DELTA_OMEGA_DOT` ({116, 17}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_I0_DOT` ({133, 15}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_CIS` ({148, 16}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_CIC` ({164, 16}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_CRS` ({180, 24}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_CRC` ({204, 24}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_CUS` ({228, 21}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_CUC` ({249, 21}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_TOP2` ({39, 11}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_URA_NED0` ({50, 5}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_URA_NED1` ({55, 3}}

- `const std::vector< std::pair< int32_t, int32_t > > CNAV_URA_NED2` {{{58, 3}}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_TOC` {{{61, 11}}}
- `const std::vector< std::pair< int, int > > CNAV_AF0` {{{72, 26}}}
- `const std::vector< std::pair< int, int > > CNAV_AF1` {{{98, 20}}}
- `const std::vector< std::pair< int, int > > CNAV_AF2` {{{118, 10}}}
- `const std::vector< std::pair< int, int > > CNAV_TGD` {{{128, 13}}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_ISCL1` {{{141, 13}}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_ISCL2` {{{154, 13}}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_ISCL5I` {{{167, 13}}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_ISCL5Q` {{{180, 13}}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_ALPHA0` {{{193, 8}}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_ALPHA1` {{{201, 8}}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_ALPHA2` {{{209, 8}}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_ALPHA3` {{{217, 8}}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_BETA0` {{{225, 8}}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_BETA1` {{{233, 8}}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_BETA2` {{{241, 8}}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_BETA3` {{{249, 8}}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_WNOP` {{{257, 8}}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_A0` {{{128, 16}}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_A1` {{{144, 13}}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_A2` {{{157, 7}}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_DELTA_TLS` {{{164, 8}}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_TOT` {{{172, 16}}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_WN_OT` {{{188, 13}}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_WN_LSF` {{{201, 13}}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_DN` {{{214, 4}}}
- `const std::vector< std::pair< int32_t, int32_t > > CNAV_DELTA_TLSF` {{{218, 8}}}

## Variables

- `constexpr int32_t GPS_CNAV_DATA_PAGE_BITS` = 300
- `constexpr int32_t CNAV_TOW_LSB` = 6
- `constexpr int32_t CNAV_TOP1_LSB` = 300
- `constexpr int32_t CNAV_TOE1_LSB` = 300
- `constexpr double CNAV_DELTA_A_LSB` = TWO\_N9
- `constexpr double CNAV_A_DOT_LSB` = TWO\_N21
- `constexpr double CNAV_DELTA_N0_LSB` = TWO\_N44 \* GNSS\_PI
- `constexpr double CNAV_DELTA_N0_DOT_LSB` = TWO\_N57 \* GNSS\_PI
- `constexpr double CNAV_M0_LSB` = TWO\_N32 \* GNSS\_PI
- `constexpr double CNAV_E_ECCENTRICITY_LSB` = TWO\_N34
- `constexpr double CNAV_OMEGA_LSB` = TWO\_N32 \* GNSS\_PI
- `constexpr int32_t CNAV_TOE2_LSB` = 300
- `constexpr double CNAV_OMEGA0_LSB` = TWO\_N32 \* GNSS\_PI
- `constexpr double CNAV_I0_LSB` = TWO\_N32 \* GNSS\_PI
- `constexpr double CNAV_DELTA_OMEGA_DOT_LSB` = TWO\_N44 \* GNSS\_PI
- `constexpr double CNAV_I0_DOT_LSB` = TWO\_N44 \* GNSS\_PI
- `constexpr double CNAV_CIS_LSB` = TWO\_N30
- `constexpr double CNAV_CIC_LSB` = TWO\_N30
- `constexpr double CNAV_CRS_LSB` = TWO\_N8
- `constexpr double CNAV_CRC_LSB` = TWO\_N8
- `constexpr double CNAV_CUS_LSB` = TWO\_N30
- `constexpr double CNAV_CUC_LSB` = TWO\_N30
- `constexpr int32_t CNAV_TOP2_LSB` = 300
- `constexpr int32_t CNAV_TOC_LSB` = 300

- constexpr double CNAV\_AF0\_LSB = TWO\_N35
- constexpr double CNAV\_AF1\_LSB = TWO\_N48
- constexpr double CNAV\_AF2\_LSB = TWO\_N60
- constexpr double CNAV\_TGD\_LSB = TWO\_N35
- constexpr double CNAV\_ISCL1\_LSB = TWO\_N35
- constexpr double CNAV\_ISCL2\_LSB = TWO\_N35
- constexpr double CNAV\_ISCL5I\_LSB = TWO\_N35
- constexpr double CNAV\_ISCL5Q\_LSB = TWO\_N35
- constexpr double CNAV\_ALPHA0\_LSB = TWO\_N30
- constexpr double CNAV\_ALPHA1\_LSB = TWO\_N27
- constexpr double CNAV\_ALPHA2\_LSB = TWO\_N24
- constexpr double CNAV\_ALPHA3\_LSB = TWO\_N24
- constexpr double CNAV\_BETA0\_LSB = TWO\_P11
- constexpr double CNAV\_BETA1\_LSB = TWO\_P14
- constexpr double CNAV\_BETA2\_LSB = TWO\_P16
- constexpr double CNAV\_BETA3\_LSB = TWO\_P16
- constexpr double CNAV\_A0\_LSB = TWO\_N35
- constexpr double CNAV\_A1\_LSB = TWO\_N51
- constexpr double CNAV\_A2\_LSB = TWO\_N68
- constexpr int32\_t CNAV\_DELTA\_TLS\_LSB = 1
- constexpr int32\_t CNAV\_TOT\_LSB = TWO\_P4
- constexpr int32\_t CNAV\_WN\_OT\_LSB = 1
- constexpr int32\_t CNAV\_WN\_LSF\_LSB = 1
- constexpr int32\_t CNAV\_DN\_LSB = 1
- constexpr int32\_t CNAV\_DELTA\_TLSF\_LSB = 1
- constexpr double CNAV\_A\_REF = 26559710.0
- constexpr double CNAV\_OMEGA\_DOT\_REF = -2.6e-9

### 11.759.1 Detailed Description

Defines parameters for GPS CNAV.

#### Author

Antonio Ramos, 2017. antonio.ramos(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [GPS\\_CNAV.h](#).

---

## 11.760 GPS\_CNAV.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file GPS_CNAV.h
00003  * \brief Defines parameters for GPS CNAV
00004  * \author Antonio Ramos, 2017. antonio.ramos(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_GPS_CNAV_H
00019 #define GNSS_SDR_GPS_CNAV_H
00020
00021 #include "MATH_CONSTANTS.h"

```

```

00022 #include <cstdint>
00023 #include <utility> // std::pair
00024 #include <vector>
00025
00026 /** \addtogroup Core
00027 * \{ */
00028 /** \addtogroup System_Parameters
00029 * \{ */
00030
00031
00032 // CNAV GPS NAVIGATION MESSAGE STRUCTURE
00033 // NAVIGATION MESSAGE FIELDS POSITIONS (from IS-GPS-200M Appendix III)
00034
00035 constexpr int32_t GPS_CNAV_DATA_PAGE_BITS = 300;
00036
00037 // common to all messages
00038 const std::vector<std::pair<int32_t, int32_t> > CNAV_PRN({{9, 6}});
00039 const std::vector<std::pair<int32_t, int32_t> > CNAV_MSG_TYPE({{15, 6}});
00040 const std::vector<std::pair<int32_t, int32_t> > CNAV_TOW({{21, 17}}); // GPS Time Of Week in seconds
00041 constexpr int32_t CNAV_TOW_LSB = 6;
00042 const std::vector<std::pair<int32_t, int32_t> > CNAV_ALERT_FLAG({{38, 1}});
00043
00044 // MESSAGE TYPE 10 (Ephemeris 1)
00045 const std::vector<std::pair<int32_t, int32_t> > CNAV_WN({{39, 13}});
00046 const std::vector<std::pair<int32_t, int32_t> > CNAV_HEALTH({{52, 3}});
00047 const std::vector<std::pair<int32_t, int32_t> > CNAV_TOP1({{55, 11}});
00048 constexpr int32_t CNAV_TOP1_LSB = 300;
00049 const std::vector<std::pair<int32_t, int32_t> > CNAV_URA({{66, 5}});
00050
00051 const std::vector<std::pair<int32_t, int32_t> > CNAV_TOE1({{71, 11}});
00052 constexpr int32_t CNAV_TOE1_LSB = 300;
00053
00054 const std::vector<std::pair<int32_t, int32_t> > CNAV_DELTA_A({{82, 26}}); // Relative to AREF =
26,559,710 meters
00055 constexpr double CNAV_DELTA_A_LSB = TWO_N9;
00056
00057 const std::vector<std::pair<int32_t, int32_t> > CNAV_A_DOT({{108, 25}});
00058 constexpr double CNAV_A_DOT_LSB = TWO_N21;
00059
00060 const std::vector<std::pair<int32_t, int32_t> > CNAV_DELTA_N0({{133, 17}});
00061 constexpr double CNAV_DELTA_N0_LSB = TWO_N44 * GNSS_PI; // semi-circles to radians
00062 const std::vector<std::pair<int32_t, int32_t> > CNAV_DELTA_N0_DOT({{150, 23}});
00063 constexpr double CNAV_DELTA_N0_DOT_LSB = TWO_N57 * GNSS_PI; // semi-circles to radians
00064 const std::vector<std::pair<int32_t, int32_t> > CNAV_M0({{173, 33}});
00065 constexpr double CNAV_M0_LSB = TWO_N32 * GNSS_PI; // semi-circles to radians
00066 const std::vector<std::pair<int32_t, int32_t> > CNAV_E_ECCENTRICITY({{206, 33}});
00067 constexpr double CNAV_E_ECCENTRICITY_LSB = TWO_N34;
00068 const std::vector<std::pair<int32_t, int32_t> > CNAV_OMEGA({{239, 33}});
00069 constexpr double CNAV_OMEGA_LSB = TWO_N32 * GNSS_PI; // semi-circles to radians
00070 const std::vector<std::pair<int32_t, int32_t> > CNAV_INTEGRITY_FLAG({{272, 1}});
00071 const std::vector<std::pair<int32_t, int32_t> > CNAV_L2_PHASING_FLAG({{273, 1}});
00072
00073 // MESSAGE TYPE 11 (Ephemeris 2)
00074 const std::vector<std::pair<int32_t, int32_t> > CNAV_TOE2({{39, 11}});
00075 constexpr int32_t CNAV_TOE2_LSB = 300;
00076 const std::vector<std::pair<int32_t, int32_t> > CNAV_OMEGA0({{50, 33}});
00077 constexpr double CNAV_OMEGA0_LSB = TWO_N32 * GNSS_PI; // semi-circles to radians
00078 const std::vector<std::pair<int32_t, int32_t> > CNAV_I0({{83, 33}});
00079 constexpr double CNAV_I0_LSB = TWO_N32 * GNSS_PI; // semi-circles to
radians
00080 const std::vector<std::pair<int32_t, int32_t> > CNAV_DELTA_OMEGA_DOT({{116, 17}}); // Relative to REF
= -2.6 x 10^-9 semi-circles/second.
00081 constexpr double CNAV_DELTA_OMEGA_DOT_LSB = TWO_N44 * GNSS_PI; // semi-circles to
radians
00082 const std::vector<std::pair<int32_t, int32_t> > CNAV_I0_DOT({{133, 15}});
00083 constexpr double CNAV_I0_DOT_LSB = TWO_N44 * GNSS_PI; // semi-circles to radians
00084 const std::vector<std::pair<int32_t, int32_t> > CNAV_CIS({{148, 16}});
00085 constexpr double CNAV_CIS_LSB = TWO_N30;
00086 const std::vector<std::pair<int32_t, int32_t> > CNAV_CIC({{164, 16}});
00087 constexpr double CNAV_CIC_LSB = TWO_N30;
00088 const std::vector<std::pair<int32_t, int32_t> > CNAV_CRS({{180, 24}});
00089 constexpr double CNAV_CRS_LSB = TWO_N8;
00090 const std::vector<std::pair<int32_t, int32_t> > CNAV_CRC({{204, 24}});
00091 constexpr double CNAV_CRC_LSB = TWO_N8;
00092 const std::vector<std::pair<int32_t, int32_t> > CNAV_CUS({{228, 21}});
00093 constexpr double CNAV_CUS_LSB = TWO_N30;
00094 const std::vector<std::pair<int32_t, int32_t> > CNAV_CUC({{249, 21}});
00095 constexpr double CNAV_CUC_LSB = TWO_N30;
00096
00097
00098 // MESSAGE TYPE 30 (CLOCK, IONO, GROUP DELAY)
00099 const std::vector<std::pair<int32_t, int32_t> > CNAV_TOP2({{39, 11}});
00100 constexpr int32_t CNAV_TOP2_LSB = 300;
00101 const std::vector<std::pair<int32_t, int32_t> > CNAV_URA_NED0({{50, 5}});
00102 const std::vector<std::pair<int32_t, int32_t> > CNAV_URA_NED1({{55, 3}});
00103 const std::vector<std::pair<int32_t, int32_t> > CNAV_URA_NED2({{58, 3}});
00104 const std::vector<std::pair<int32_t, int32_t> > CNAV_TOC({{61, 11}});

```

```

00105 constexpr int32_t CNAV_TOC_LSB = 300;
00106 const std::vector<std::pair<int, int> > CNAV_AF0({{72, 26}});
00107 constexpr double CNAV_AF0_LSB = TWO_N35;
00108 const std::vector<std::pair<int, int> > CNAV_AF1({{98, 20}});
00109 constexpr double CNAV_AF1_LSB = TWO_N48;
00110 const std::vector<std::pair<int, int> > CNAV_AF2({{118, 10}});
00111 constexpr double CNAV_AF2_LSB = TWO_N60;
00112 const std::vector<std::pair<int, int> > CNAV_TGD({{128, 13}});
00113 constexpr double CNAV_TGD_LSB = TWO_N35;
00114 const std::vector<std::pair<int32_t, int32_t> > CNAV_ISCL1({{141, 13}});
00115 constexpr double CNAV_ISCL1_LSB = TWO_N35;
00116 const std::vector<std::pair<int32_t, int32_t> > CNAV_ISCL2({{154, 13}});
00117 constexpr double CNAV_ISCL2_LSB = TWO_N35;
00118 const std::vector<std::pair<int32_t, int32_t> > CNAV_ISCL5I({{167, 13}});
00119 constexpr double CNAV_ISCL5I_LSB = TWO_N35;
00120 const std::vector<std::pair<int32_t, int32_t> > CNAV_ISCL5Q({{180, 13}});
00121 constexpr double CNAV_ISCL5Q_LSB = TWO_N35;
00122 // Ionospheric parameters
00123 const std::vector<std::pair<int32_t, int32_t> > CNAV_ALPHA0({{193, 8}});
00124 constexpr double CNAV_ALPHA0_LSB = TWO_N30;
00125 const std::vector<std::pair<int32_t, int32_t> > CNAV_ALPHA1({{201, 8}});
00126 constexpr double CNAV_ALPHA1_LSB = TWO_N27;
00127 const std::vector<std::pair<int32_t, int32_t> > CNAV_ALPHA2({{209, 8}});
00128 constexpr double CNAV_ALPHA2_LSB = TWO_N24;
00129 const std::vector<std::pair<int32_t, int32_t> > CNAV_ALPHA3({{217, 8}});
00130 constexpr double CNAV_ALPHA3_LSB = TWO_N24;
00131 const std::vector<std::pair<int32_t, int32_t> > CNAV_BETA0({{225, 8}});
00132 constexpr double CNAV_BETA0_LSB = TWO_P11;
00133 const std::vector<std::pair<int32_t, int32_t> > CNAV_BETA1({{233, 8}});
00134 constexpr double CNAV_BETA1_LSB = TWO_P14;
00135 const std::vector<std::pair<int32_t, int32_t> > CNAV_BETA2({{241, 8}});
00136 constexpr double CNAV_BETA2_LSB = TWO_P16;
00137 const std::vector<std::pair<int32_t, int32_t> > CNAV_BETA3({{249, 8}});
00138 constexpr double CNAV_BETA3_LSB = TWO_P16;
00139 const std::vector<std::pair<int32_t, int32_t> > CNAV_WNOP({{257, 8}});
00140
00141
00142 // MESSAGE TYPE 33 (CLOCK and UTC)
00143 const std::vector<std::pair<int32_t, int32_t> > CNAV_A0({{128, 16}});
00144 constexpr double CNAV_A0_LSB = TWO_N35;
00145 const std::vector<std::pair<int32_t, int32_t> > CNAV_A1({{144, 13}});
00146 constexpr double CNAV_A1_LSB = TWO_N51;
00147 const std::vector<std::pair<int32_t, int32_t> > CNAV_A2({{157, 7}});
00148 constexpr double CNAV_A2_LSB = TWO_N68;
00149 const std::vector<std::pair<int32_t, int32_t> > CNAV_DELTA_TLS({{164, 8}});
00150 constexpr int32_t CNAV_DELTA_TLS_LSB = 1;
00151 const std::vector<std::pair<int32_t, int32_t> > CNAV_TOT({{172, 16}});
00152 constexpr int32_t CNAV_TOT_LSB = TWO_P4;
00153 const std::vector<std::pair<int32_t, int32_t> > CNAV_WN_OT({{188, 13}});
00154 constexpr int32_t CNAV_WN_OT_LSB = 1;
00155 const std::vector<std::pair<int32_t, int32_t> > CNAV_WN_LSF({{201, 13}});
00156 constexpr int32_t CNAV_WN_LSF_LSB = 1;
00157 const std::vector<std::pair<int32_t, int32_t> > CNAV_DN({{214, 4}});
00158 constexpr int32_t CNAV_DN_LSB = 1;
00159 const std::vector<std::pair<int32_t, int32_t> > CNAV_DELTA_TLSF({{218, 8}});
00160 constexpr int32_t CNAV_DELTA_TLSF_LSB = 1;
00161
00162 constexpr double CNAV_A_REF = 26559710.0; // [m] See IS-GPS-200M, Table 30-I.
00163 constexpr double CNAV_OMEGA_DOT_REF = -2.6e-9; // [semicircles / s], see IS-GPS-200M, Table 30-I.
00164
00165 // TODO: Add more frames (Almanac, etc...)
00166
00167
00168 /** \} */
00169 /** \} */
00170 #endif // GNSS_SDR_GPS_CNAV_H

```

## 11.761 gps\_cnav\_ephemeris.h File Reference

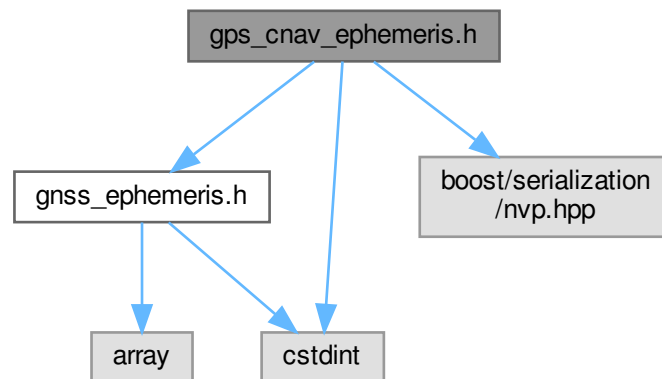
Interface of a GPS CNAV EPHEMERIS storage.

```

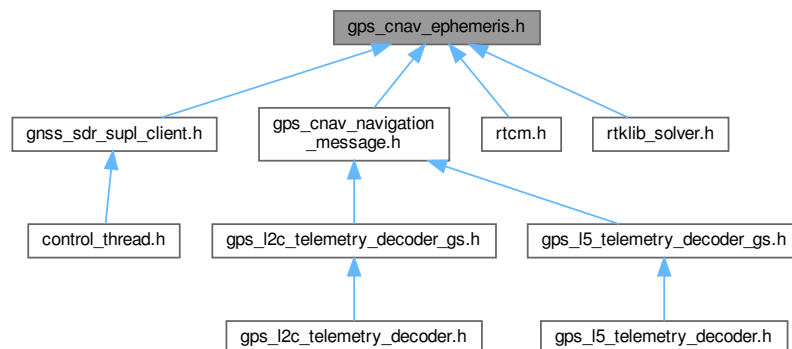
#include "gnss_ephemeris.h"
#include <boost/serialization/nvp.hpp>
#include <cstdint>

```

Include dependency graph for `gps_cnav_ephemeris.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Gps\\_CNAV\\_Ephemeris](#)

*This is a storage class for the GPS CNAV ephemeris data as described in IS-GPS-200M.*

### 11.761.1 Detailed Description

Interface of a GPS CNAV EPHEMERIS storage.

#### Author

Javier Arribas, 2015. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gps\\_cnav\\_ephemeris.h](#).

---



## 11.762 gps\_cnav\_ephemeris.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file gps_cnav_ephemeris.h
00003  * \brief Interface of a GPS CNAV EPHEMERIS storage
00004  * \author Javier Arribas, 2015. jarribas(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_GPS_CNAV_EPHEMERIS_H
00019 #define GNSS_SDR_GPS_CNAV_EPHEMERIS_H
00020
00021 #include "gnss_ephemeris.h"
00022 #include <boost/serialization/nvp.hpp>
00023 #include <cstdint>
00024
00025 /** \addtogroup Core
00026  * \{ */
00027 /** \addtogroup System_Parameters
00028  * \{ */
00029
00030
00031  /*!
00032  * \brief This is a storage class for the GPS CNAV ephemeris data as described
00033  * in IS-GPS-200M
00034  *
00035  * See https://www.gps.gov/technical/icwg/IS-GPS-200M.pdf Appendix III
00036  */
00037  class Gps_CNAV_Ephemeris : public Gnss_Ephemeris
00038  {
00039  public:
00040      /*!
00041       * Constructor
00042       */
00043      Gps_CNAV_Ephemeris()
00044      {
00045          this->System = 'G';
00046      }
00047
00048      double delta_A{};          //!< Semi-major axis difference at reference time
00049      double Adot{};             //!< Change rate in semi-major axis
00050      double delta_ndot{};       //!< Rate of mean motion difference from computed value
00051      double delta_OMEGAAdot{};  //!< Rate of Right Ascension difference [semi-circles/s]
00052      int32_t toe1{};            //!< Ephemeris data reference time of week (Ref. 20.3.3.4.3 IS-GPS-200M)
00053      [s] int32_t toe2{};        //!< Ephemeris data reference time of week (Ref. 20.3.3.4.3 IS-GPS-200M)
00054      [s] int32_t signal_health{}; //!< Signal health (L1/L2/L5)
00055      int32_t top{};             //!< Data predict time of week
00056      int32_t URA{};             //!< ED Accuracy Index
00057
00058      double URA0{};             //!< NED Accuracy Index
00059      double URA1{};             //!< NED Accuracy Change Index
00060      double URA2{};             //!< NED Accuracy Change Rate Index
00061
00062      // Group Delay Differential Parameters
00063      double TGD{};             //!< Estimated Group Delay Differential: L1-L2 correction term only for the benefit
of "L1 P(Y)" or "L2 P(Y)" s users [s]
00064      double ISCL1{};
00065      double ISCL2{};
00066      double ISCL5I{};
00067      double ISCL5Q{};
00068
00069      /*! \brief If true, enhanced level of integrity assurance.
00070       *
00071       * If false, indicates that the conveying signal is provided with the
00072       * legacy level of integrity assurance. That is, the probability that the
00073       * instantaneous URE of the conveying signal exceeds 4.42 times the upper
00074       * bound value of the current broadcast URA index, for more than 5.2
00075       * seconds, without an accompanying alert, is less than 1E-5 per hour. If
00076       * true, indicates that the conveying signal is provided with an enhanced
00077       * level of integrity assurance. That is, the probability that the
00078       * instantaneous URE of the conveying signal exceeds 5.73 times the upper
00079       * bound value of the current broadcast URA index, for more than 5.2
00080       * seconds, without an accompanying alert, is less than 1E-8 per hour.

```

```

00081     */
00082     bool integrity_status_flag{};
00083     bool l2c_phasing_flag{};
00084     bool alert_flag{};          //!< If true, indicates that the SV URA may be worse than indicated in
                                d_SV_accuracy, use that SV at our own risk.
00085     bool antispoofing_flag{};  //!< If true, the AntiSpoofing mode is ON in that SV
00086
00087     template <class Archive>
00088
00089     /*!
00090     * \brief Serialize is a boost standard method to be called by the boost XML
00091     * serialization. Here is used to save the ephemeris data on disk file.
00092     */
00093     inline void serialize(Archive& archive, const uint32_t version)
00094     {
00095         using boost::serialization::make_nvp;
00096         if (version)
00097         {
00098             };
00099
00100             archive& BOOST_SERIALIZATION_NVP(PRN);
00101             archive& BOOST_SERIALIZATION_NVP(M_0);
00102             archive& BOOST_SERIALIZATION_NVP(delta_n);
00103             archive& BOOST_SERIALIZATION_NVP(ecc);
00104             archive& BOOST_SERIALIZATION_NVP(sqrtA);
00105             archive& BOOST_SERIALIZATION_NVP(OMEGA_0);
00106             archive& BOOST_SERIALIZATION_NVP(i_0);
00107             archive& BOOST_SERIALIZATION_NVP(omega);
00108             archive& BOOST_SERIALIZATION_NVP(OMEGAdot);
00109             archive& BOOST_SERIALIZATION_NVP(idot);
00110             archive& BOOST_SERIALIZATION_NVP(Cuc);
00111             archive& BOOST_SERIALIZATION_NVP(Cus);
00112             archive& BOOST_SERIALIZATION_NVP(Crc);
00113             archive& BOOST_SERIALIZATION_NVP(Crs);
00114             archive& BOOST_SERIALIZATION_NVP(Cic);
00115             archive& BOOST_SERIALIZATION_NVP(Cis);
00116             archive& BOOST_SERIALIZATION_NVP(toe);
00117             archive& BOOST_SERIALIZATION_NVP(toc);
00118             archive& BOOST_SERIALIZATION_NVP(af0);
00119             archive& BOOST_SERIALIZATION_NVP(af1);
00120             archive& BOOST_SERIALIZATION_NVP(af2);
00121             archive& BOOST_SERIALIZATION_NVP(WN);
00122             archive& BOOST_SERIALIZATION_NVP(tow);
00123             archive& BOOST_SERIALIZATION_NVP(satClkDrift);
00124             archive& BOOST_SERIALIZATION_NVP(dtr);
00125
00126             archive& BOOST_SERIALIZATION_NVP(toe1);
00127             archive& BOOST_SERIALIZATION_NVP(toe2);
00128             archive& BOOST_SERIALIZATION_NVP(TGD);
00129             archive& BOOST_SERIALIZATION_NVP(ISCL1);
00130             archive& BOOST_SERIALIZATION_NVP(ISCL2);
00131             archive& BOOST_SERIALIZATION_NVP(ISCL5I);
00132             archive& BOOST_SERIALIZATION_NVP(ISCL5Q);
00133             archive& BOOST_SERIALIZATION_NVP(delta_A);
00134             archive& BOOST_SERIALIZATION_NVP(Adot);
00135             archive& BOOST_SERIALIZATION_NVP(delta_OMEGAdot);
00136             archive& BOOST_SERIALIZATION_NVP(integrity_status_flag);
00137             archive& BOOST_SERIALIZATION_NVP(l2c_phasing_flag);
00138             archive& BOOST_SERIALIZATION_NVP(alert_flag);
00139             archive& BOOST_SERIALIZATION_NVP(antispoofing_flag);
00140         }
00141     };
00142
00143     /** \} */
00144     /** \} */
00145     /** \} */
00146 #endif // GNSS_SDR_GPS_CNAV_EPHEMERIS_H

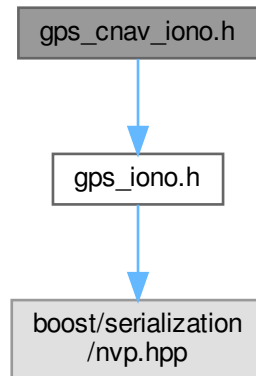
```

## 11.763 gps\_cnav\_iono.h File Reference

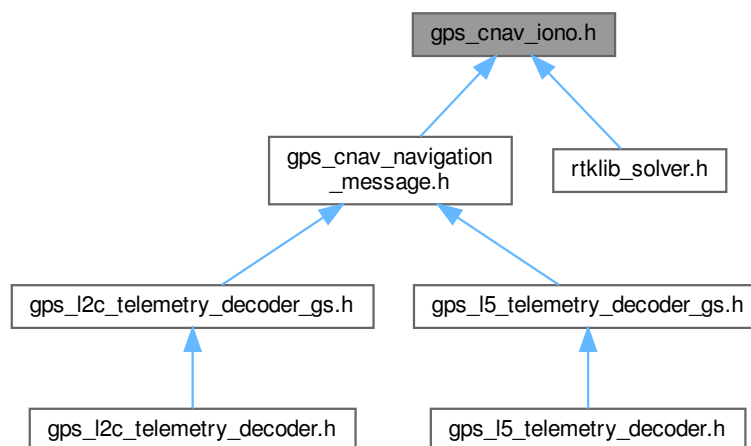
Interface of a GPS CNAV IONOSPHERIC MODEL storage.

```
#include "gps_iono.h"
```

Include dependency graph for gps\_cnav\_iono.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Gps\\_CNAV\\_Iono](#)

*This class is a storage for the GPS IONOSPHERIC data as described in IS-GPS-200M.*

## 11.763.1 Detailed Description

Interface of a GPS CNAV IONOSPHERIC MODEL storage.

**Author**

Javier Arribas, 2015. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gps\\_cnav\\_iono.h](#).

---

**11.764 gps\_cnav\_iono.h**

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file gps_cnav_iono.h
00003  * \brief Interface of a GPS CNAV IONOSPHERIC MODEL storage
00004  * \author Javier Arribas, 2015. jarribas(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_GPS_CNAV_IONO_H
00019 #define GNSS_SDR_GPS_CNAV_IONO_H
00020
00021 #include "gps_iono.h"
00022
00023 /** \addtogroup Core
00024  * \{ */
00025 /** \addtogroup System_Parameters
00026  * \{ */
00027
00028
00029 /*!
00030  * \brief This class is a storage for the GPS IONOSPHERIC data as described in IS-GPS-200M
00031  *
00032  * See https://www.gps.gov/technical/icwg/IS-GPS-200M.pdf Appendix III
00033  */
00034 class Gps_CNAV_Iono : public Gps_Iono
00035 {
00036 public:
00037     Gps_CNAV_Iono() = default; //!< Default constructor
00038 };
00039
00040
00041 /** \} */
00042 /** \} */
00043 #endif // GNSS_SDR_GPS_CNAV_IONO_H

```

**11.765 gps\_cnav\_navigation\_message.h File Reference**

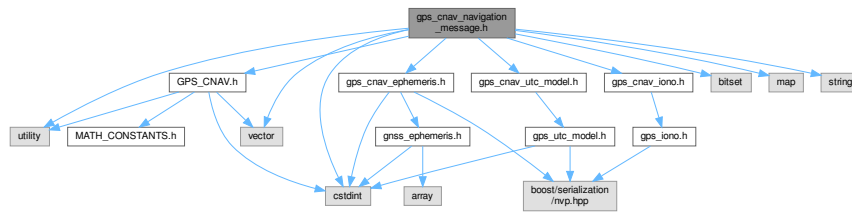
Interface of a GPS CNAV Data message decoder.

```

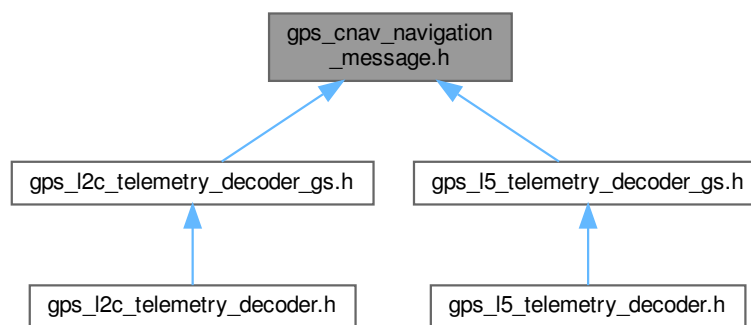
#include "GPS_CNAV.h"
#include "gps_cnav_ephemeris.h"
#include "gps_cnav_iono.h"
#include "gps_cnav_utc_model.h"
#include <bitset>
#include <cstdint>
#include <map>
#include <string>
#include <utility>
#include <vector>

```

Include dependency graph for gps\_cnav\_navigation\_message.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Gps\\_CNAV\\_Navigation\\_Message](#)

*This class decodes a GPS CNAV Data message as described in IS-GPS-200M.*

## 11.765.1 Detailed Description

Interface of a GPS CNAV Data message decoder.

### Author

Javier Arribas, 2015. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gps\\_cnav\\_navigation\\_message.h](#).

---

## 11.766 gps\_cnav\_navigation\_message.h

[Go to the documentation of this file.](#)

```
00001  /*!
00002  * \file gps_cnav_navigation_message.h
00003  * \brief Interface of a GPS CNAV Data message decoder
00004  * \author Javier Arribas, 2015. jarribas(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
```

```

00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_GPS_CNAV_NAVIGATION_MESSAGE_H
00019 #define GNSS_SDR_GPS_CNAV_NAVIGATION_MESSAGE_H
00020
00021
00022 #include "GPS_CNAV.h"
00023 #include "gps_cnav_ephemeris.h"
00024 #include "gps_cnav_iono.h"
00025 #include "gps_cnav_utc_model.h"
00026 #include <bitset>
00027 #include <cstdint>
00028 #include <map>
00029 #include <string>
00030 #include <utility>
00031 #include <vector>
00032
00033 /** \addtogroup Core
00034  * \{ */
00035 /** \addtogroup System_Parameters
00036  * \{ */
00037
00038
00039 /*!
00040  * \brief This class decodes a GPS CNAV Data message as described in IS-GPS-200M
00041  *
00042  * See https://www.gps.gov/technical/icwg/IS-GPS-200M.pdf Appendix III
00043  */
00044 class Gps_CNAV_Navigation_Message
00045 {
00046 public:
00047     /*!
00048      * Default constructor
00049      */
00050     Gps_CNAV_Navigation_Message();
00051
00052     void decode_page(const std::bitset<GPS_CNAV_DATA_PAGE_BITS>& data_bits);
00053
00054     /*!
00055      * \brief Obtain a GPS SV Ephemeris class filled with current SV data
00056      */
00057     Gps_CNAV_Ephemeris get_ephemeris() const;
00058
00059     /*!
00060      * \brief Check if we have a new iono record stored in the GPS ephemeris class
00061      */
00062     bool have_new_iono();
00063
00064     /*!
00065      * \brief Obtain a GPS ionospheric correction parameters class filled with current SV data
00066      */
00067     Gps_CNAV_Iono get_iono() const;
00068
00069     /*!
00070      * \brief Obtain a GPS UTC model parameters class filled with current SV data
00071      */
00072     Gps_CNAV_Utc_Model get_utc_model();
00073
00074     /*!
00075      * \brief Check if we have a new GPS UTC model record stored in the GPS ephemeris class
00076      */
00077     bool have_new_utc_model();
00078
00079     /*!
00080      * \brief Check if we have a new ephemeris stored in the GPS ephemeris class
00081      */
00082     bool have_new_ephemeris();
00083
00084 private:
00085     uint64_t read_navigation_unsigned(const std::bitset<GPS_CNAV_DATA_PAGE_BITS>& bits, const
std::vector<std::pair<int32_t, int32_t>& parameter) const;
00086     int64_t read_navigation_signed(const std::bitset<GPS_CNAV_DATA_PAGE_BITS>& bits, const
std::vector<std::pair<int32_t, int32_t>& parameter) const;
00087     bool read_navigation_bool(const std::bitset<GPS_CNAV_DATA_PAGE_BITS>& bits, const
std::vector<std::pair<int32_t, int32_t>& parameter) const;
00088
00089     Gps_CNAV_Ephemeris ephemeris_record{};
00090     Gps_CNAV_Iono iono_record{};
00091     Gps_CNAV_Utc_Model utc_model_record{};
00092
00093     std::map<int32_t, std::string> satelliteBlock; ///< Map that stores to which block the PRN
belongs https://www.navcen.uscg.gov/?Do=constellationStatus

```

```

00094
00095     int32_t d_TOW{};
00096
00097     bool b_flag_ephemeris_1{};
00098     bool b_flag_ephemeris_2{};
00099     bool b_flag_iono_valid{}; //!< If set, it indicates that the ionospheric parameters are filled
    and are not yet read by the get_iono
00100     bool b_flag_utc_valid{}; //!< If set, it indicates that the utc parameters are filled and are
    not yet read by the get_utc_model
00101 };
00102
00103
00104 /** @} */
00105 /** @} */
00106 #endif // GNSS_SDR_GPS_CNAV_NAVIGATION_MESSAGE_H

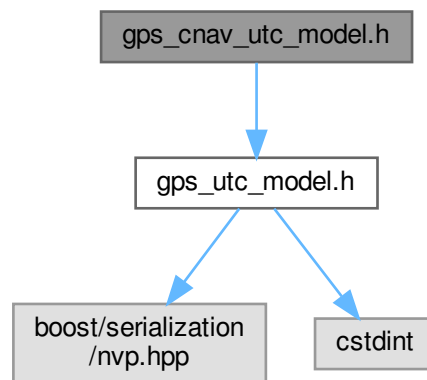
```

## 11.767 gps\_cnav\_utc\_model.h File Reference

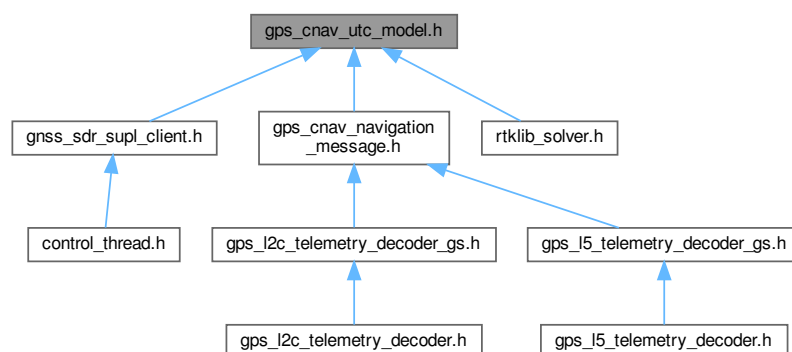
Interface of a GPS CNAV UTC MODEL storage.

```
#include "gps_utc_model.h"
```

Include dependency graph for gps\_cnav\_utc\_model.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Gps\\_CNAV\\_Utc\\_Model](#)

*This class is a storage for the GPS UTC MODEL data as described in in IS-GPS-200M.*

### 11.767.1 Detailed Description

Interface of a GPS CNAV UTC MODEL storage.

#### Author

Javier Arribas, 2013. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later

Definition in file [gps\\_cnav\\_utc\\_model.h](#).

---

## 11.768 gps\_cnav\_utc\_model.h

[Go to the documentation of this file.](#)

```
00001  /*!
00002  * \file gps_cnav_utc_model.h
00003  * \brief Interface of a GPS CNAV UTC MODEL storage
00004  * \author Javier Arribas, 2013. jarribas(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_GPS_CNAV_UTC_MODEL_H
00019 #define GNSS_SDR_GPS_CNAV_UTC_MODEL_H
00020
00021 #include "gps_utc_model.h"
00022
00023 /** \addtogroup Core
00024  * \{ */
00025 /** \addtogroup System_Parameters
00026  * \{ */
00027
00028
00029 /*!
00030 * \brief This class is a storage for the GPS UTC MODEL data as described in in IS-GPS-200M
00031 *
00032 * See https://www.gps.gov/technical/icwg/IS-GPS-200M.pdf Appendix III
00033 */
00034 class Gps_CNAV_Utc_Model : public Gps_Utc_Model
00035 {
00036 public:
00037     Gps_CNAV_Utc_Model() = default; //!< Default constructor
00038 };
00039
00040
00041 /** \} */
00042 /** \} */
00043 #endif // GNSS_SDR_GPS_CNAV_UTC_MODEL_H
```

## 11.769 gps\_ephemeris.h File Reference

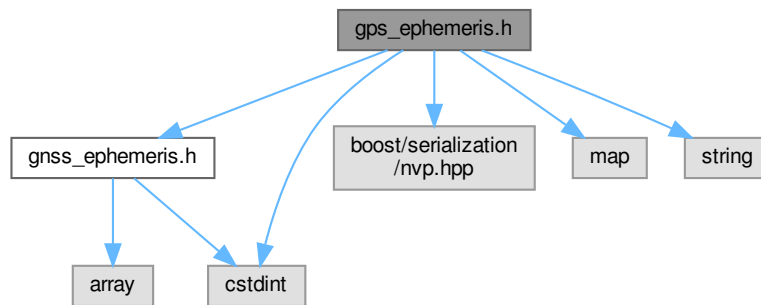
Interface of a GPS EPHEMERIS storage.

```
#include "gnss_ephemeris.h"
#include <boost/serialization/nvp.hpp>
#include <cstdint>
#include <map>
```

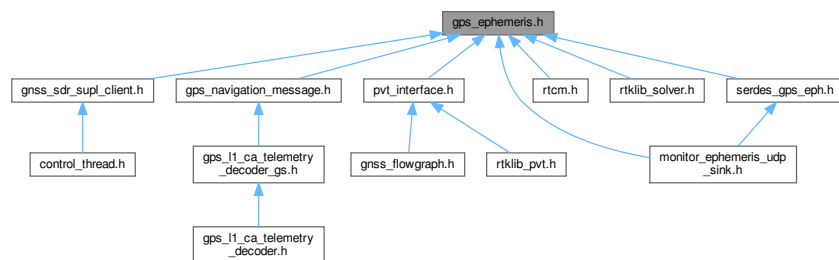


```
#include <string>
```

Include dependency graph for gps\_ephemeris.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Gps\\_Ephemeris](#)

*This class is a storage and orbital model functions for the GPS SV ephemeris data as described in IS-GPS-200M.*

## 11.769.1 Detailed Description

Interface of a GPS EPHEMERIS storage.

### Author

Javier Arribas, 2013. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gps\\_ephemeris.h](#).

---

## 11.770 gps\_ephemeris.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file gps_ephemeris.h
00003  * \brief Interface of a GPS EPHEMERIS storage
00004  * \author Javier Arribas, 2013. jarribas(at)cttc.es
00005  *
00006  * -----

```

```

00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_GPS_EPHEMERIS_H
00019 #define GNSS_SDR_GPS_EPHEMERIS_H
00020
00021
00022 #include "gnss_ephemeris.h"
00023 #include <boost/serialization/nvp.hpp>
00024 #include <stdint>
00025 #include <map>
00026 #include <string>
00027
00028 /** \addtogroup Core
00029  * \{ */
00030 /** \addtogroup System_Parameters
00031  * \{ */
00032
00033
00034 /*!
00035  * \brief This class is a storage and orbital model functions for the GPS SV
00036  * ephemeris data as described in IS-GPS-200M
00037  *
00038  * See https://www.gps.gov/technical/icwg/IS-GPS-200M.pdf Appendix II
00039  */
00040 class Gps_Ephemeris : public Gnss_Ephemeris
00041 {
00042 public:
00043     /*!
00044      * Default constructor
00045      */
00046     Gps_Ephemeris();
00047
00048     int32_t code_on_L2{};    //!< If 1, P code ON in L2; if 2, C/A code ON in L2;
00049     bool L2_P_data_flag{};  //!< When true, indicates that the NAV data stream was commanded OFF on
the P-code of the L2 channel
00050     int32_t SV_accuracy{};  //!< User Range Accuracy (URA) index of the SV (reference paragraph 6.2.1)
for the standard positioning service user (Ref 20.3.3.3.1.3 IS-GPS-200M)
00051     int32_t SV_health{};    //!< Satellite health status
00052     double TGD{};          //!< Estimated Group Delay Differential: L1-L2 correction term only for
the benefit of "L1 P(Y)" or "L2 P(Y)" s users [s]
00053     int32_t IODC{};        //!< Issue of Data, Clock
00054     int32_t IODE_SF2{};    //!< Issue of Data, Ephemeris (IODE), subframe 2
00055     int32_t IODE_SF3{};    //!< Issue of Data, Ephemeris (IODE), subframe 3
00056     int32_t AODO{};        //!< Age of Data Offset (AODO) term for the navigation message correction
table (NMCT) contained in subframe 4 (reference paragraph 20.3.3.5.1.9) [s]
00057
00058     bool fit_interval_flag{}; //!< indicates the curve-fit interval used by the CS (Block
II/IIA/IIR/IIR-M/IIF) and SS (Block IIIA) in determining the ephemeris parameters, as follows: 0 = 4
hours, 1 = greater than 4 hours.
00059     double spare1{};
00060     double spare2{};
00061
00062     // Flags
00063
00064     /*! \brief If true, enhanced level of integrity assurance.
00065      *
00066      * If false, indicates that the conveying signal is provided with the legacy level of integrity
assurance.
00067      * That is, the probability that the instantaneous URE of the conveying signal exceeds 4.42 times
the upper bound
00068      * value of the current broadcast URA index, for more than 5.2 seconds, without an accompanying
alert, is less
00069      * than 1E-5 per hour. If true, indicates that the conveying signal is provided with an enhanced
level of
00070      * integrity assurance. That is, the probability that the instantaneous URE of the conveying
signal exceeds 5.73
00071      * times the upper bound value of the current broadcast URA index, for more than 5.2 seconds,
without an
00072      * accompanying alert, is less than 1E-8 per hour.
00073      */
00074     bool integrity_status_flag{};
00075     bool alert_flag{};    //!< If true, indicates that the SV URA may be worse than indicated in
d_SV_accuracy, use that SV at our own risk.
00076     bool antispoofing_flag{}; //!< If true, the AntiSporfing mode is ON in that SV
00077
00078     std::map<int, std::string> satelliteBlock; //!< Map that stores to which block the PRN belongs
https://www.navcen.uscg.gov/?Do=constellationStatus
00079

```

```

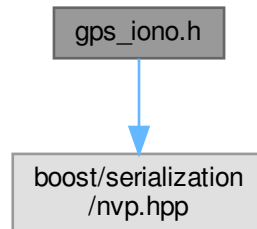
00080     template <class Archive>
00081
00082     /*!
00083      * \brief Serialize is a boost standard method to be called by the boost XML
00084      * serialization. Here is used to save the ephemeris data on disk file.
00085      */
00086     inline void serialize(Archive& archive, const uint32_t version)
00087     {
00088         using boost::serialization::make_nvp;
00089         if (version)
00090         {
00091             };
00092
00093         archive& BOOST_SERIALIZATION_NVP(PRN);
00094         archive& BOOST_SERIALIZATION_NVP(M_0);
00095         archive& BOOST_SERIALIZATION_NVP(delta_n);
00096         archive& BOOST_SERIALIZATION_NVP(ecc);
00097         archive& BOOST_SERIALIZATION_NVP(sqrtA);
00098         archive& BOOST_SERIALIZATION_NVP(OMEGA_0);
00099         archive& BOOST_SERIALIZATION_NVP(i_0);
00100         archive& BOOST_SERIALIZATION_NVP(omega);
00101         archive& BOOST_SERIALIZATION_NVP(OMEGAdot);
00102         archive& BOOST_SERIALIZATION_NVP(idot);
00103         archive& BOOST_SERIALIZATION_NVP(Cuc);
00104         archive& BOOST_SERIALIZATION_NVP(Cus);
00105         archive& BOOST_SERIALIZATION_NVP(Crc);
00106         archive& BOOST_SERIALIZATION_NVP(Crs);
00107         archive& BOOST_SERIALIZATION_NVP(Cic);
00108         archive& BOOST_SERIALIZATION_NVP(Cis);
00109         archive& BOOST_SERIALIZATION_NVP(toe);
00110         archive& BOOST_SERIALIZATION_NVP(toc);
00111         archive& BOOST_SERIALIZATION_NVP(af0);
00112         archive& BOOST_SERIALIZATION_NVP(af1);
00113         archive& BOOST_SERIALIZATION_NVP(af2);
00114         archive& BOOST_SERIALIZATION_NVP(WN);
00115         archive& BOOST_SERIALIZATION_NVP(tow);
00116         archive& BOOST_SERIALIZATION_NVP(satClkDrift);
00117         archive& BOOST_SERIALIZATION_NVP(dtr);
00118
00119         archive& BOOST_SERIALIZATION_NVP(ODE_SF2);
00120         archive& BOOST_SERIALIZATION_NVP(ODE_SF3);
00121         archive& BOOST_SERIALIZATION_NVP(code_on_L2);
00122         archive& BOOST_SERIALIZATION_NVP(L2_P_data_flag);
00123         archive& BOOST_SERIALIZATION_NVP(SV_accuracy);
00124         archive& BOOST_SERIALIZATION_NVP(SV_health);
00125         archive& BOOST_SERIALIZATION_NVP(TGD);
00126         archive& BOOST_SERIALIZATION_NVP(ODC);
00127         archive& BOOST_SERIALIZATION_NVP(AODO);
00128         archive& BOOST_SERIALIZATION_NVP(fit_interval_flag);
00129         archive& BOOST_SERIALIZATION_NVP(spare1);
00130         archive& BOOST_SERIALIZATION_NVP(spare2);
00131         archive& BOOST_SERIALIZATION_NVP(integrity_status_flag);
00132         archive& BOOST_SERIALIZATION_NVP(alert_flag);
00133         archive& BOOST_SERIALIZATION_NVP(antispoofing_flag);
00134     }
00135 };
00136
00137
00138 /** \} */
00139 /** \} */
00140 #endif // GNSS_SDR_GPS_EPHEMERIS_H

```

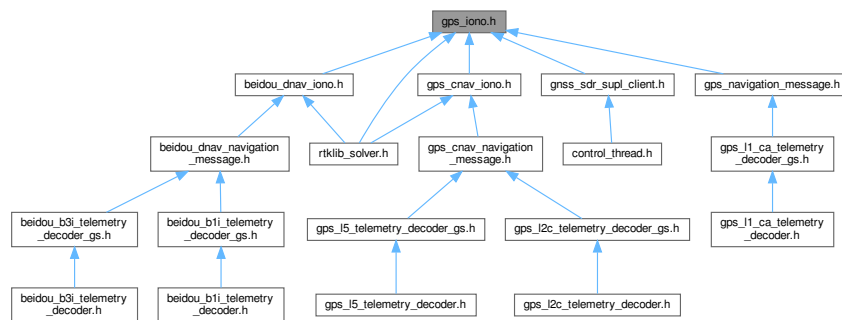
## 11.771 gps\_iono.h File Reference

Interface of a GPS IONOSPHERIC MODEL storage.

```
#include <boost/serialization/nvp.hpp>
Include dependency graph for gps_iono.h:
```



This graph shows which files directly or indirectly include this file:



## Classes

- class [Gps\\_iono](#)

*This class is a storage for the GPS IONOSPHERIC data as described in IS-GPS-200M.*

## 11.771.1 Detailed Description

Interface of a GPS IONOSPHERIC MODEL storage.

### Author

Javier Arribas, 2013. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gps\\_iono.h](#).

---

## 11.772 gps\_iono.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file gps_iono.h
00003  * \brief Interface of a GPS IONOSPHERIC MODEL storage
00004  * \author Javier Arribas, 2013. jarribas(at)cttc.es
00005  *
```

```

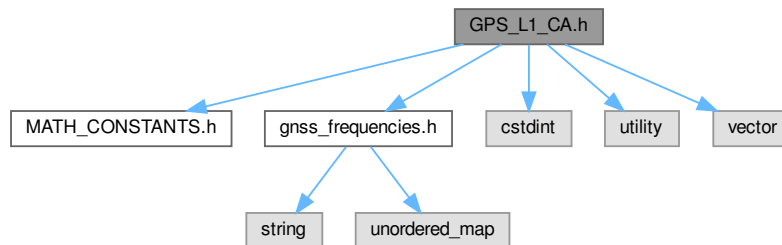
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_GPS_IONO_H
00019 #define GNSS_SDR_GPS_IONO_H
00020
00021
00022 #include <boost/serialization/nvp.hpp>
00023
00024 /** \addtogroup Core
00025  * \{ */
00026 /** \addtogroup System_Parameters
00027  * \{ */
00028
00029
00030 /*!
00031  * \brief This class is a storage for the GPS IONOSPHERIC data as described in IS-GPS-200M
00032  *
00033  * See https://www.gps.gov/technical/icwg/IS-GPS-200M.pdf Appendix II
00034  */
00035 class Gps_Iono
00036 {
00037 public:
00038     Gps_Iono() = default; //!< Default constructor
00039
00040     // Ionospheric parameters
00041     double alpha0{}; //!< Coefficient 0 of a cubic equation representing the amplitude of the
vertical delay [s]
00042     double alpha1{}; //!< Coefficient 1 of a cubic equation representing the amplitude of the
vertical delay [s/semi-circle]
00043     double alpha2{}; //!< Coefficient 2 of a cubic equation representing the amplitude of the
vertical delay [s(semi-circle)^2]
00044     double alpha3{}; //!< Coefficient 3 of a cubic equation representing the amplitude of the
vertical delay [s(semi-circle)^3]
00045     double beta0{}; //!< Coefficient 0 of a cubic equation representing the period of the model [s]
00046     double beta1{}; //!< Coefficient 1 of a cubic equation representing the period of the model
[s/semi-circle]
00047     double beta2{}; //!< Coefficient 2 of a cubic equation representing the period of the model
[s(semi-circle)^2]
00048     double beta3{}; //!< Coefficient 3 of a cubic equation representing the period of the model
[s(semi-circle)^3]
00049
00050     bool valid{}; //!< Valid flag
00051
00052     template <class Archive>
00053
00054     /*!
00055     * \brief Serialize is a boost standard method to be called by the boost XML
00056     * serialization. Here is used to save the ephemeris data on disk file.
00057     */
00058     inline void serialize(Archive& archive, const unsigned int version)
00059     {
00060         if (version)
00061         {
00062             archive& BOOST_SERIALIZATION_NVP(alpha0);
00063             archive& BOOST_SERIALIZATION_NVP(alpha1);
00064             archive& BOOST_SERIALIZATION_NVP(alpha2);
00065             archive& BOOST_SERIALIZATION_NVP(alpha3);
00066             archive& BOOST_SERIALIZATION_NVP(beta0);
00067             archive& BOOST_SERIALIZATION_NVP(beta1);
00068             archive& BOOST_SERIALIZATION_NVP(beta2);
00069             archive& BOOST_SERIALIZATION_NVP(beta3);
00070         }
00071     }
00072 };
00073
00074
00075 /** \} */
00076 /** \} */
00077 #endif // GNSS_SDR_GPS_IONO_H

```

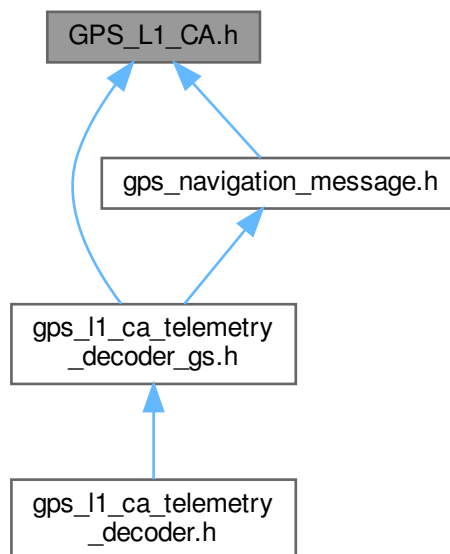
## 11.773 GPS\_L1\_CA.h File Reference

Defines system parameters for GPS L1 C/A signal and NAV data.

```
#include "MATH_CONSTANTS.h"
#include "gnss_frequencies.h"
#include <stdint>
#include <utility>
#include <vector>
Include dependency graph for GPS_L1_CA.h:
```



This graph shows which files directly or indirectly include this file:



## Functions

- `const std::vector< std::pair< int32_t, int32_t > > TOW ({{{31, 17}}})`
- `const std::vector< std::pair< int32_t, int32_t > > INTEGRITY_STATUS_FLAG ({{{23, 1}}})`
- `const std::vector< std::pair< int32_t, int32_t > > ALERT_FLAG ({{{48, 1}}})`
- `const std::vector< std::pair< int32_t, int32_t > > ANTI_SPOOFING_FLAG ({{{49, 1}}})`
- `const std::vector< std::pair< int32_t, int32_t > > SUBFRAME_ID ({{{50, 3}}})`
- `const std::vector< std::pair< int32_t, int32_t > > GPS_WEEK ({{{61, 10}}})`
- `const std::vector< std::pair< int32_t, int32_t > > CA_OR_P_ON_L2 ({{{71, 2}}})`

- `const std::vector< std::pair< int32_t, int32_t > > SV_ACCURACY {{{73, 4}}}`
- `const std::vector< std::pair< int32_t, int32_t > > SV_HEALTH {{{77, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > L2_P_DATA_FLAG {{{91, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > T_GD {{{197, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > IODC {{{83, 2}, {211, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > T_OC {{{219, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > A_F2 {{{241, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > A_F1 {{{249, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > A_F0 {{{271, 22}}}`
- `const std::vector< std::pair< int32_t, int32_t > > IODE_SF2 {{{61, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > C_RS {{{69, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DELTA_N {{{91, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > M_0 {{{107, 8}, {121, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > C_UC {{{151, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ECCENTRICITY {{{167, 8}, {181, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > C_US {{{211, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > SQRT_A {{{227, 8}, {241, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > T_OE {{{271, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > FIT_INTERVAL_FLAG {{{271, 1}}}`
- `const std::vector< std::pair< int32_t, int32_t > > AODO {{{272, 5}}}`
- `const std::vector< std::pair< int32_t, int32_t > > C_IC {{{61, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA_0 {{{77, 8}, {91, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > C_IS {{{121, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > I_0 {{{137, 8}, {151, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > C_RC {{{181, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA {{{197, 8}, {211, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > OMEGA_DOT {{{241, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > IODE_SF3 {{{271, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > I_DOT {{{279, 14}}}`
- `const std::vector< std::pair< int32_t, int32_t > > SV_DATA_ID {{{61, 2}}}`
- `const std::vector< std::pair< int32_t, int32_t > > SV_PAGE {{{63, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALPHA_0 {{{69, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALPHA_1 {{{77, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALPHA_2 {{{91, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALPHA_3 {{{99, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > BETA_0 {{{107, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > BETA_1 {{{121, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > BETA_2 {{{129, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > BETA_3 {{{137, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > A_1 {{{151, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > A_0 {{{181, 24}, {211, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > T_OT {{{219, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > WN_T {{{227, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DELTAT_LS {{{241, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > WN_LSF {{{249, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DN {{{257, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > DELTAT_LSF {{{271, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV25 {{{229, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV26 {{{241, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV27 {{{247, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV28 {{{253, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV29 {{{259, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV30 {{{271, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV31 {{{277, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV32 {{{283, 6}}}`

- `const std::vector< std::pair< int32_t, int32_t > > T_OA {{{69, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > WN_A {{{77, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV1 {{{91, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV2 {{{97, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV3 {{{103, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV4 {{{109, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV5 {{{121, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV6 {{{127, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV7 {{{133, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV8 {{{139, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV9 {{{151, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV10 {{{157, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV11 {{{163, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV12 {{{169, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV13 {{{181, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV14 {{{187, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV15 {{{193, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV16 {{{199, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV17 {{{211, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV18 {{{217, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV19 {{{223, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV20 {{{229, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV21 {{{241, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV22 {{{247, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV23 {{{253, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > HEALTH_SV24 {{{259, 6}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALM_ECC {{{69, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALM_TOA {{{91, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALM_DELTAI {{{99, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALM_OMEGADOT {{{121, 16}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALM_SVHEALTH {{{137, 8}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALM_SQUAREA {{{151, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALM_OMEGAZERO {{{181, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALM_OMEGA {{{211, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALM_MZERO {{{241, 24}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALM_AF0 {{{271, 8}, {290, 3}}}`
- `const std::vector< std::pair< int32_t, int32_t > > ALM_AF1 {{{279, 11}}}`

## Variables

- `constexpr double GPS_L1_FREQ_HZ = FREQ1`  
*L1 [Hz].*
- `constexpr double GPS_L1_CA_CODE_RATE_CPS = 1.023e6`  
*GPS L1 C/A code rate [chips/s].*
- `constexpr double GPS_L1_CA_CODE_LENGTH_CHIPS = 1023.0`  
*GPS L1 C/A code length [chips].*
- `constexpr double GPS_L1_CA_CODE_PERIOD_S = 0.001`  
*GPS L1 C/A code period [seconds].*
- `constexpr double GPS_L1_CA_CHIP_PERIOD_S = 9.7752e-07`  
*GPS L1 C/A chip period [seconds].*
- `constexpr uint32_t GPS_L1_CA_CODE_PERIOD_MS = 1U`  
*GPS L1 C/A code period [ms].*
- `constexpr uint32_t GPS_L1_CA_BIT_PERIOD_MS = 20U`  
*GPS L1 C/A bit period [ms].*



- Generated by Doxygen

- constexpr double [ALPHA\\_2\\_LSB](#) = [TWO\\_N24](#)
- constexpr double [ALPHA\\_3\\_LSB](#) = [TWO\\_N24](#)
- constexpr double [BETA\\_0\\_LSB](#) = [TWO\\_P11](#)
- constexpr double [BETA\\_1\\_LSB](#) = [TWO\\_P14](#)
- constexpr double [BETA\\_2\\_LSB](#) = [TWO\\_P16](#)
- constexpr double [BETA\\_3\\_LSB](#) = [TWO\\_P16](#)
- constexpr double [A\\_1\\_LSB](#) = [TWO\\_N50](#)
- constexpr double [A\\_0\\_LSB](#) = [TWO\\_N30](#)
- constexpr double [T\\_OT\\_LSB](#) = [TWO\\_P12](#)
- constexpr double [WN\\_T\\_LSB](#) = 1
- constexpr double [DELTAT\\_LS\\_LSB](#) = 1
- constexpr double [WN\\_LSF\\_LSB](#) = 1
- constexpr double [DN\\_LSB](#) = 1
- constexpr double [DELTAT\\_LSF\\_LSB](#) = 1
- constexpr int32\_t [T\\_OA\\_LSB](#) = [TWO\\_P12](#)
- constexpr double [ALM\\_ECC\\_LSB](#) = [TWO\\_N21](#)
- constexpr int32\_t [ALM\\_TOA\\_LSB](#) = static\_cast<int32\_t>([TWO\\_P12](#))
- constexpr double [ALM\\_DELTAI\\_LSB](#) = [TWO\\_N19](#)
- constexpr double [ALM\\_OMEGADOT\\_LSB](#) = [TWO\\_N38](#)
- constexpr double [ALM\\_SQUAREA\\_LSB](#) = [TWO\\_N11](#)
- constexpr double [ALM\\_OMEGAZERO\\_LSB](#) = [TWO\\_N23](#)
- constexpr double [ALM\\_OMEGA\\_LSB](#) = [TWO\\_N23](#)
- constexpr double [ALM\\_MZERO\\_LSB](#) = [TWO\\_N23](#)
- constexpr double [ALM\\_AF0\\_LSB](#) = [TWO\\_N20](#)
- constexpr double [ALM\\_AF1\\_LSB](#) = [TWO\\_N38](#)

### 11.773.1 Detailed Description

Defines system parameters for GPS L1 C/A signal and NAV data.

#### Author

Javier Arribas, 2011. [jarribas\(at\)cttc.es](mailto:jarribas(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [GPS\\_L1\\_CA.h](#).

---

## 11.774 GPS\_L1\_CA.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file GPS_L1_CA.h
00003  * \brief Defines system parameters for GPS L1 C/A signal and NAV data
00004  * \author Javier Arribas, 2011. jarribas(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_GPS_L1_CA_H
00019 #define GNSS_SDR_GPS_L1_CA_H
00020
00021 #include "MATH_CONSTANTS.h"
00022 #include "gnss_frequencies.h"
00023 #include <cstdint>
00024 #include <utility> // std::pair

```

[illegible]

```

00100 constexpr double A_F1_LSB = TWO_N43;
00101 const std::vector<std::pair<int32_t, int32_t>> A_F0({{271, 22}});
00102 constexpr double A_F0_LSB = TWO_N31;
00103
00104 // SUBFRAME 2
00105 const std::vector<std::pair<int32_t, int32_t>> IODE_SF2({{61, 8}});
00106 const std::vector<std::pair<int32_t, int32_t>> C_RS({{69, 16}});
00107 constexpr double C_RS_LSB = TWO_N5;
00108 const std::vector<std::pair<int32_t, int32_t>> DELTA_N({{91, 16}});
00109 constexpr double DELTA_N_LSB = PI_TWO_N43;
00110 const std::vector<std::pair<int32_t, int32_t>> M_0({{107, 8}, {121, 24}});
00111 constexpr double M_0_LSB = PI_TWO_N31;
00112 const std::vector<std::pair<int32_t, int32_t>> C_UC({{151, 16}});
00113 constexpr double C_UC_LSB = TWO_N29;
00114 const std::vector<std::pair<int32_t, int32_t>> ECCENTRICITY({{167, 8}, {181, 24}});
00115 constexpr double ECCENTRICITY_LSB = TWO_N33;
00116 const std::vector<std::pair<int32_t, int32_t>> C_US({{211, 16}});
00117 constexpr double C_US_LSB = TWO_N29;
00118 const std::vector<std::pair<int32_t, int32_t>> SQRT_A({{227, 8}, {241, 24}});
00119 constexpr double SQRT_A_LSB = TWO_N19;
00120 const std::vector<std::pair<int32_t, int32_t>> T_OE({{271, 16}});
00121 constexpr int32_t T_OE_LSB = static_cast<int32_t>(TWO_P4);
00122 const std::vector<std::pair<int32_t, int32_t>> FIT_INTERVAL_FLAG({{271, 1}});
00123 const std::vector<std::pair<int32_t, int32_t>> AODO({{272, 5}});
00124 constexpr int32_t AODO_LSB = 900;
00125
00126 // SUBFRAME 3
00127 const std::vector<std::pair<int32_t, int32_t>> C_IC({{61, 16}});
00128 constexpr double C_IC_LSB = TWO_N29;
00129 const std::vector<std::pair<int32_t, int32_t>> OMEGA_0({{77, 8}, {91, 24}});
00130 constexpr double OMEGA_0_LSB = PI_TWO_N31;
00131 const std::vector<std::pair<int32_t, int32_t>> C_IS({{121, 16}});
00132 constexpr double C_IS_LSB = TWO_N29;
00133 const std::vector<std::pair<int32_t, int32_t>> I_0({{137, 8}, {151, 24}});
00134 constexpr double I_0_LSB = PI_TWO_N31;
00135 const std::vector<std::pair<int32_t, int32_t>> C_RC({{181, 16}});
00136 constexpr double C_RC_LSB = TWO_N5;
00137 const std::vector<std::pair<int32_t, int32_t>> OMEGA({{197, 8}, {211, 24}});
00138 constexpr double OMEGA_LSB = PI_TWO_N31;
00139 const std::vector<std::pair<int32_t, int32_t>> OMEGA_DOT({{241, 24}});
00140 constexpr double OMEGA_DOT_LSB = PI_TWO_N43;
00141 const std::vector<std::pair<int32_t, int32_t>> IODE_SF3({{271, 8}});
00142 const std::vector<std::pair<int32_t, int32_t>> I_DOT({{279, 14}});
00143 constexpr double I_DOT_LSB = PI_TWO_N43;
00144
00145 // SUBFRAME 4-5
00146 const std::vector<std::pair<int32_t, int32_t>> SV_DATA_ID({{61, 2}});
00147 const std::vector<std::pair<int32_t, int32_t>> SV_PAGE({{63, 6}});
00148
00149 // SUBFRAME 4
00150 /// \todo read all pages of subframe 4
00151 // Page 18 - Ionospheric and UTC data
00152 const std::vector<std::pair<int32_t, int32_t>> ALPHA_0({{69, 8}});
00153 constexpr double ALPHA_0_LSB = TWO_N30;
00154 const std::vector<std::pair<int32_t, int32_t>> ALPHA_1({{77, 8}});
00155 constexpr double ALPHA_1_LSB = TWO_N27;
00156 const std::vector<std::pair<int32_t, int32_t>> ALPHA_2({{91, 8}});
00157 constexpr double ALPHA_2_LSB = TWO_N24;
00158 const std::vector<std::pair<int32_t, int32_t>> ALPHA_3({{99, 8}});
00159 constexpr double ALPHA_3_LSB = TWO_N24;
00160 const std::vector<std::pair<int32_t, int32_t>> BETA_0({{107, 8}});
00161 constexpr double BETA_0_LSB = TWO_P11;
00162 const std::vector<std::pair<int32_t, int32_t>> BETA_1({{121, 8}});
00163 constexpr double BETA_1_LSB = TWO_P14;
00164 const std::vector<std::pair<int32_t, int32_t>> BETA_2({{129, 8}});
00165 constexpr double BETA_2_LSB = TWO_P16;
00166 const std::vector<std::pair<int32_t, int32_t>> BETA_3({{137, 8}});
00167 constexpr double BETA_3_LSB = TWO_P16;
00168 const std::vector<std::pair<int32_t, int32_t>> A_1({{151, 24}});
00169 constexpr double A_1_LSB = TWO_N50;
00170 const std::vector<std::pair<int32_t, int32_t>> A_0({{181, 24}, {211, 8}});
00171 constexpr double A_0_LSB = TWO_N30;
00172 const std::vector<std::pair<int32_t, int32_t>> T_OT({{219, 8}});
00173 constexpr double T_OT_LSB = TWO_P12;
00174 const std::vector<std::pair<int32_t, int32_t>> WN_T({{227, 8}});
00175 constexpr double WN_T_LSB = 1;
00176 const std::vector<std::pair<int32_t, int32_t>> DELTAT_LS({{241, 8}});
00177 constexpr double DELTAT_LS_LSB = 1;
00178 const std::vector<std::pair<int32_t, int32_t>> WN_LSF({{249, 8}});
00179 constexpr double WN_LSF_LSB = 1;
00180 const std::vector<std::pair<int32_t, int32_t>> DN({{257, 8}});
00181 constexpr double DN_LSB = 1;
00182 const std::vector<std::pair<int32_t, int32_t>> DELTAT_LSF({{271, 8}});
00183 constexpr double DELTAT_LSF_LSB = 1;
00184
00185 // Page 25 - Antispoofing, SV config and SV health (PRN 25 -32)
00186 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV25({{229, 6}});

```

```

00187 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV26({{241, 6}});
00188 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV27({{247, 6}});
00189 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV28({{253, 6}});
00190 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV29({{259, 6}});
00191 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV30({{271, 6}});
00192 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV31({{277, 6}});
00193 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV32({{283, 6}});
00194
00195
00196 // SUBFRAME 5
00197 //! \todo read all pages of subframe 5
00198
00199 // page 25 - Health (PRN 1 - 24)
00200 const std::vector<std::pair<int32_t, int32_t>> T_OA({{69, 8}});
00201 constexpr int32_t T_OA_LSB = TWO_P12;
00202 const std::vector<std::pair<int32_t, int32_t>> WN_A({{77, 8}});
00203 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV1({{91, 6}});
00204 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV2({{97, 6}});
00205 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV3({{103, 6}});
00206 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV4({{109, 6}});
00207 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV5({{121, 6}});
00208 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV6({{127, 6}});
00209 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV7({{133, 6}});
00210 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV8({{139, 6}});
00211 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV9({{151, 6}});
00212 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV10({{157, 6}});
00213 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV11({{163, 6}});
00214 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV12({{169, 6}});
00215 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV13({{181, 6}});
00216 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV14({{187, 6}});
00217 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV15({{193, 6}});
00218 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV16({{199, 6}});
00219 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV17({{211, 6}});
00220 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV18({{217, 6}});
00221 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV19({{223, 6}});
00222 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV20({{229, 6}});
00223 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV21({{241, 6}});
00224 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV22({{247, 6}});
00225 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV23({{253, 6}});
00226 const std::vector<std::pair<int32_t, int32_t>> HEALTH_SV24({{259, 6}});
00227
00228
00229 // Almanac
00230 const std::vector<std::pair<int32_t, int32_t>> ALM_ECC({{69, 16}});
00231 constexpr double ALM_ECC_LSB = TWO_N21;
00232 const std::vector<std::pair<int32_t, int32_t>> ALM_TOA({{91, 8}});
00233 constexpr int32_t ALM_TOA_LSB = static_cast<int32_t>(TWO_P12);
00234 const std::vector<std::pair<int32_t, int32_t>> ALM_DELTAT({{99, 16}});
00235 constexpr double ALM_DELTAT_LSB = TWO_N19;
00236 const std::vector<std::pair<int32_t, int32_t>> ALM_OMEGADOT({{121, 16}});
00237 constexpr double ALM_OMEGADOT_LSB = TWO_N38;
00238 const std::vector<std::pair<int32_t, int32_t>> ALM_SVHEALTH({{137, 8}});
00239 const std::vector<std::pair<int32_t, int32_t>> ALM_SQUAREA({{151, 24}});
00240 constexpr double ALM_SQUAREA_LSB = TWO_N11;
00241 const std::vector<std::pair<int32_t, int32_t>> ALM_OMEGAZERO({{181, 24}});
00242 constexpr double ALM_OMEGAZERO_LSB = TWO_N23;
00243 const std::vector<std::pair<int32_t, int32_t>> ALM_OMEGA({{211, 24}});
00244 constexpr double ALM_OMEGA_LSB = TWO_N23;
00245 const std::vector<std::pair<int32_t, int32_t>> ALM_MZERO({{241, 24}});
00246 constexpr double ALM_MZERO_LSB = TWO_N23;
00247
00248 const std::vector<std::pair<int32_t, int32_t>> ALM_AF0({{271, 8}, {290, 3}});
00249 constexpr double ALM_AF0_LSB = TWO_N20;
00250 const std::vector<std::pair<int32_t, int32_t>> ALM_AF1({{279, 11}});
00251 constexpr double ALM_AF1_LSB = TWO_N38;
00252 /** \} */
00253 /** \} */
00254 #endif // GNSS_SDR_GPS_L1_CA_H

```

## 11.775 GPS\_L2C.h File Reference

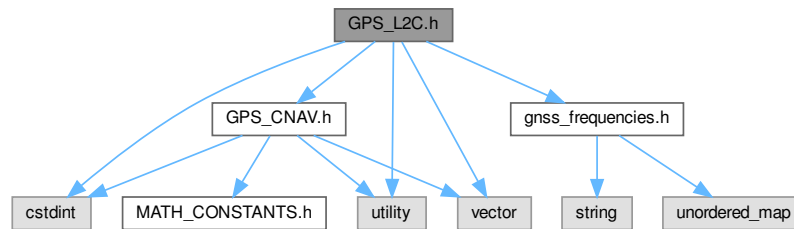
Defines system parameters for GPS L2C signal.

```

#include "GPS_CNAV.h"
#include "gnss_frequencies.h"
#include <stdint>
#include <utility>
#include <vector>

```

Include dependency graph for GPS\_L2C.h:



## Variables

- constexpr double `GPS_L2_FREQ_HZ` = `FREQ2`  
*L2 [Hz].*
- constexpr double `GPS_L2_L_PERIOD_S` = 1.5  
*GPS L2 L code period [seconds].*
- constexpr double `GPS_L2_M_CODE_RATE_CPS` = 0.5115e6  
*GPS L2 M code rate [chips/s].*
- constexpr double `GPS_L2_M_PERIOD_S` = 0.02  
*GPS L2 M code period [seconds].*
- constexpr double `GPS_L2_L_CODE_RATE_CPS` = 0.5115e6  
*GPS L2 L code rate [chips/s].*
- constexpr int32\_t `GPS_L2_M_CODE_LENGTH_CHIPS` = 10230  
*GPS L2 M code length [chips].*
- constexpr int32\_t `GPS_L2_L_CODE_LENGTH_CHIPS` = 767250  
*GPS L2 L code length [chips].*
- constexpr int32\_t `GPS_L2_CNAV_DATA_PAGE_BITS` = 300  
*GPS L2 CNAV page length, including preamble and CRC [bits].*
- constexpr int32\_t `GPS_L2_SYMBOLS_PER_BIT` = 2
- constexpr int32\_t `GPS_L2_SAMPLES_PER_SYMBOL` = 1
- constexpr int32\_t `GPS_L2_CNAV_DATA_PAGE_SYMBOLS` = 600
- constexpr int32\_t `GPS_L2_CNAV_DATA_PAGE_DURATION_S` = 12
- constexpr int32\_t `GPS_L2C_HISTORY_DEEP` = 5
- constexpr uint32\_t `GPS_L2C_OPT_ACQ_FS_SPS` = 2000000  
*Sampling frequency that maximizes the acquisition SNR while using a non-multiple of chip rate.*
- constexpr int32\_t `GPS_L2C_M_INIT_REG` [115]

## 11.775.1 Detailed Description

Defines system parameters for GPS L2C signal.

### Author

Javier Arribas, 2015. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
Definition in file [GPS\\_L2C.h](#).

---

## 11.776 GPS\_L2C.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file GPS_L2C.h
00003  * \brief Defines system parameters for GPS L2C signal
00004  * \author Javier Arribas, 2015. jarribas(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_GPS_L2C_H
00019 #define GNSS_SDR_GPS_L2C_H
00020
00021
00022 #include "GPS_CNAV.h"
00023 #include "gnss_frequencies.h"
00024 #include <stdint>
00025 #include <utility> // std::pair
00026 #include <vector>
00027
00028 /** \addtogroup Core
00029  * \{ */
00030 /** \addtogroup System_Parameters
00031  * \{ */
00032
00033
00034 // carrier and code frequencies
00035 constexpr double GPS_L2_FREQ_HZ = FREQ2; //!< L2 [Hz]
00036 constexpr double GPS_L2_L_PERIOD_S = 1.5; //!< GPS L2 L code period [seconds]
00037 constexpr double GPS_L2_M_CODE_RATE_CPS = 0.5115e6; //!< GPS L2 M code rate [chips/s]
00038 constexpr double GPS_L2_M_PERIOD_S = 0.02; //!< GPS L2 M code period [seconds]
00039 constexpr double GPS_L2_L_CODE_RATE_CPS = 0.5115e6; //!< GPS L2 L code rate [chips/s]
00040 constexpr int32_t GPS_L2_M_CODE_LENGTH_CHIPS = 10230; //!< GPS L2 M code length [chips]
00041 constexpr int32_t GPS_L2_L_CODE_LENGTH_CHIPS = 767250; //!< GPS L2 L code length [chips]
00042 constexpr int32_t GPS_L2_CNAV_DATA_PAGE_BITS = 300; //!< GPS L2 CNAV page length, including
    preamble and CRC [bits]
00043 constexpr int32_t GPS_L2_SYMBOLS_PER_BIT = 2;
00044 constexpr int32_t GPS_L2_SAMPLES_PER_SYMBOL = 1;
00045 constexpr int32_t GPS_L2_CNAV_DATA_PAGE_SYMBOLS = 600;
00046 constexpr int32_t GPS_L2_CNAV_DATA_PAGE_DURATION_S = 12;
00047
00048 constexpr int32_t GPS_L2C_HISTORY_DEEP = 5;
00049
00050 // optimum parameters
00051 constexpr uint32_t GPS_L2C_OPT_ACO_FS_SPS = 2000000; //!< Sampling frequency that maximizes the
    acquisition SNR while using a non-multiple of chip rate
00052
00053 constexpr int32_t GPS_L2C_M_INIT_REG[115] =
00054     {0742417664, 0756014035, 0002747144, 0066265724, // 1:4
00055      0601403471, 0703232733, 0124510070, 0617316361, // 5:8
00056      0047541621, 0733031046, 0713512145, 0024437606,
00057      0021264003, 0230655351, 0001314400, 0222021506,
00058      0540264026, 0205521705, 0064022144, 0120161274,
00059      0044023533, 0724744327, 0045743577, 0741201660,
00060      0700274134, 0010247261, 0713433445, 0737324162,
00061      0311627434, 0710452007, 0722462133, 0050172213,
00062      0500653703, 0755077436, 0136717361, 0756675453,
00063      0435506112, 0771353753, 0226107701, 0022025110,
00064      0402466344, 0752566114, 0702011164, 0041216771,
00065      0047457275, 0266333164, 0713167356, 0060546335,
00066      0355173035, 0617201036, 0157465571, 0767360553,
00067      0023127030, 0431343777, 0747317317, 0045706125,
00068      0002744276, 0060036467, 0217744147, 0603340174, // 57:60
00069      0326616775, 0063240065, 0111460621, // 61:63
00070      0604055104, 0157065232, 0013305707, 0603552017, // 159:162
00071      0230461355, 0603653437, 0652346475, 0743107103,
00072      0401521277, 0167335110, 0014013575, 0362051132,
00073      0617753265, 0216363634, 0755561123, 0365304033,
00074      0625025543, 0054420334, 0415473671, 0662364360,
00075      0373446602, 0417564100, 0000526452, 0226631300,
00076      0113752074, 0706134401, 0041352546, 0664630154,
00077      0276524255, 0714720530, 0714051771, 0044526647,
00078      0207164322, 0262120161, 0204244652, 0202133131,
00079      0714351204, 0657127260, 0130567507, 0670517677,
00080      0607275514, 0045413633, 0212645405, 0613700455,
00081      0706202440, 0705056276, 0020373522, 0746013617,

```

```

00082         0132720621, 0434015513, 0566721727, 0140633660});
00083
00084
00085 /** \} */
00086 /** \} */
00087 #endif // GNSS_SDR_GPS_L2C_H

```

## 11.777 GPS\_L5.h File Reference

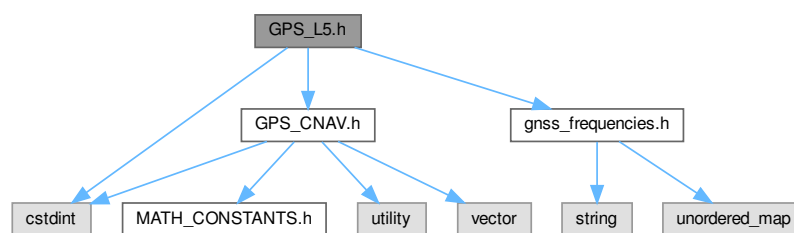
Defines system parameters for GPS L5 signal.

```
#include "GPS_CNAV.h"
```

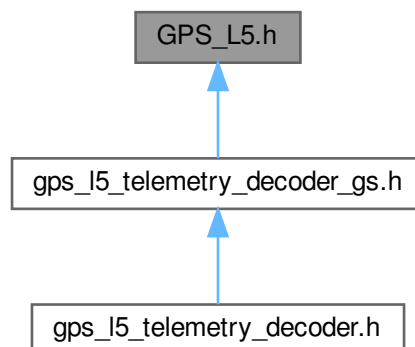
```
#include "gnss_frequencies.h"
```

```
#include <stdint>
```

Include dependency graph for GPS\_L5.h:



This graph shows which files directly or indirectly include this file:



### Variables

- constexpr double [GPS\\_L5\\_FREQ\\_HZ](#) = [FREQ5](#)  
*L5 [Hz].*
- constexpr double [GPS\\_L5I\\_CODE\\_RATE\\_CPS](#) = 10.23e6  
*GPS L5I code rate [chips/s].*
- constexpr double [GPS\\_L5I\\_PERIOD\\_S](#) = 0.001  
*GPS L5I code period [seconds].*
- constexpr double [GPS\\_L5I\\_SYMBOL\\_PERIOD\\_S](#) = 0.01



- GPS L5I symbol period [seconds].*
- constexpr double [GPS\\_L5Q\\_CODE\\_RATE\\_CPS](#) = 10.23e6
- GPS L5Q code rate [chips/s].*
- constexpr double [GPS\\_L5Q\\_PERIOD\\_S](#) = 0.001
- GPS L5Q code period [seconds].*
- constexpr int32\_t [GPS\\_L5Q\\_CODE\\_LENGTH\\_CHIPS](#) = 10230
- GPS L5Q code length [chips].*
- constexpr int32\_t [GPS\\_L5I\\_CODE\\_LENGTH\\_CHIPS](#) = 10230
- GPS L5I code length [chips].*
- constexpr int32\_t [GPS\\_L5I\\_PERIOD\\_MS](#) = 1
- GPS L5I code period [ms].*
- constexpr int32\_t [GPS\\_L5I\\_SYMBOL\\_PERIOD\\_MS](#) = 10
- GPS L5I symbol period [ms].*
- constexpr int32\_t [GPS\\_L5\\_HISTORY\\_DEEP](#) = 5
- constexpr uint32\_t [GPS\\_L5\\_OPT\\_ACQ\\_FS\\_SPS](#) = 10000000
- Sampling frequency that maximizes the acquisition SNR while using a non-multiple of chip rate.*
- constexpr int32\_t [GPS\\_L5I\\_INIT\\_REG](#) [210]
- constexpr int32\_t [GPS\\_L5Q\\_INIT\\_REG](#) [210]
- constexpr int32\_t [GPS\\_L5\\_CNAV\\_DATA\\_PAGE\\_BITS](#) = 300
- GPS L5 CNAV page length, including preamble and CRC [bits].*
- constexpr int32\_t [GPS\\_L5\\_SYMBOLS\\_PER\\_BIT](#) = 2
- constexpr int32\_t [GPS\\_L5\\_SAMPLES\\_PER\\_SYMBOL](#) = 10
- constexpr int32\_t [GPS\\_L5\\_CNAV\\_DATA\\_PAGE\\_SYMBOLS](#) = 600
- constexpr int32\_t [GPS\\_L5\\_CNAV\\_DATA\\_PAGE\\_DURATION\\_S](#) = 6
- constexpr int32\_t [GPS\\_L5I\\_NH\\_CODE\\_LENGTH](#) = 10
- constexpr int32\_t [GPS\\_L5I\\_NH\\_CODE](#) [10] = {0, 0, 0, 0, 1, 1, 0, 1, 0, 1}
- constexpr int32\_t [GPS\\_L5Q\\_NH\\_CODE\\_LENGTH](#) = 20
- constexpr int32\_t [GPS\\_L5Q\\_NH\\_CODE](#) [20] = {0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0}
- constexpr char [GPS\\_L5I\\_NH\\_CODE\\_STR](#) [11] = "0000110101"
- constexpr char [GPS\\_L5Q\\_NH\\_CODE\\_STR](#) [21] = "00000100110101001110"

### 11.777.1 Detailed Description

Defines system parameters for GPS L5 signal.

#### Author

Javier Arribas, 2017. [jarribas\(at\)cttc.es](mailto:jarribas(at)cttc.es)

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [GPS\\_L5.h](#).

---

## 11.778 GPS\_L5.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file GPS_L5.h
00003  * \brief Defines system parameters for GPS L5 signal
00004  * \author Javier Arribas, 2017. jarribas\(at\)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
```

```

00015  */
00016
00017
00018 #ifndef GNSS_SDR_GPS_L5_H
00019 #define GNSS_SDR_GPS_L5_H
00020
00021 #include "GPS_CNAV.h"
00022 #include "gnss_frequencies.h"
00023 #include <stdint>
00024
00025 /** \addtogroup Core
00026  * \{ */
00027 /** \addtogroup System_Parameters
00028  * \{ */
00029
00030
00031 // carrier and code frequencies
00032 constexpr double GPS_L5_FREQ_HZ = FREQ5;          //!< L5 [Hz]
00033 constexpr double GPS_L5I_CODE_RATE_CPS = 10.23e6;  //!< GPS L5I code rate [chips/s]
00034 constexpr double GPS_L5I_PERIOD_S = 0.001;         //!< GPS L5I code period [seconds]
00035 constexpr double GPS_L5I_SYMBOL_PERIOD_S = 0.01;   //!< GPS L5I symbol period [seconds]
00036 constexpr double GPS_L5Q_CODE_RATE_CPS = 10.23e6;  //!< GPS L5Q code rate [chips/s]
00037 constexpr double GPS_L5Q_PERIOD_S = 0.001;         //!< GPS L5Q code period [seconds]
00038 constexpr int32_t GPS_L5Q_CODE_LENGTH_CHIPS = 10230;  //!< GPS L5Q code length [chips]
00039 constexpr int32_t GPS_L5I_CODE_LENGTH_CHIPS = 10230;  //!< GPS L5I code length [chips]
00040 constexpr int32_t GPS_L5I_PERIOD_MS = 1;            //!< GPS L5I code period [ms]
00041 constexpr int32_t GPS_L5I_SYMBOL_PERIOD_MS = 10;     //!< GPS L5I symbol period [ms]
00042
00043 constexpr int32_t GPS_L5_HISTORY_DEEP = 5;
00044
00045 // optimum parameters
00046 constexpr uint32_t GPS_L5_OPT_ACQ_FS_SPS = 10000000;  //!< Sampling frequency that maximizes the
    acquisition SNR while using a non-multiple of chip rate
00047
00048 constexpr int32_t GPS_L5I_INIT_REG[210] =
00049     {266, 365, 804, 1138,
00050      1509, 1559, 1756, 2084,
00051      2170, 2303, 2527, 2687,
00052      2930, 3471, 3940, 4132,
00053      4332, 4924, 5343, 5443,
00054      5641, 5816, 5898, 5918,
00055      5955, 6243, 6345, 6477,
00056      6518, 6875, 7168, 7187,
00057      7329, 7577, 7720, 7777,
00058      8057, 5358, 3550, 3412,
00059      819,
00060      4608, 3698, 962, 3001,
00061      4441, 4937, 3717, 4730,
00062      7291, 2279, 7613, 5723,
00063      7030, 1475, 2593, 2904,
00064      2056, 2757, 3756, 6205,
00065      5053, 6437,
00066      7789, 2311, 7432, 5155,
00067      1593, 5841, 5014, 1545,
00068      3016, 4875, 2119, 229,
00069      7634, 1406, 4506, 1819,
00070      7580, 5446, 6053, 7958,
00071      5267, 2956, 3544, 1277,
00072      2996, 1758, 3360, 2718,
00073      3754, 7440, 2781, 6756,
00074      7314, 208, 5252, 696,
00075      527, 1399, 5879, 6868,
00076      217, 7681, 3788, 1337,
00077      2424, 4243, 5686, 1955,
00078      4791, 492, 1518, 6566,
00079      5349, 506, 113, 1953,
00080      2797, 934, 3023, 3632,
00081      1330, 4909, 4867, 1183,
00082      3990, 6217, 1224, 1733,
00083      2319, 3928, 2380, 841,
00084      5049, 7027, 1197, 7208,
00085      8000, 152, 6762, 3745,
00086      4723, 5502, 4796, 123,
00087      8142, 5091, 7875, 330,
00088      5272, 4912, 374, 2045,
00089      6616, 6321, 7605, 2570,
00090      2419, 1234, 1922, 4317,
00091      5110, 825, 958, 1089,
00092      7813, 6058, 7703, 6702,
00093      1714, 6371, 2281, 1986,
00094      6282, 3201, 3760, 1056,
00095      6233, 1150, 2823, 6250,
00096      645, 2401, 1639, 2946,
00097      7091, 923, 7045, 6493,
00098      1706, 5836, 926, 6086,
00099      950, 5905, 3240, 6675,
00100      3197, 1555, 3589, 4555,

```

```

00101         5671, 6948, 4664, 2086,
00102         5950, 5521, 1515};
00103
00104
00105 constexpr int32_t GPS_L5Q_INIT_REG[210] =
00106     {
00107         1701, 323, 5292, 2020,
00108         5429, 7136, 1041, 5947,
00109         4315, 148, 535, 1939,
00110         5206, 5910, 3595, 5135,
00111         6082, 6990, 3546, 1523,
00112         4548, 4484, 1893, 3961,
00113         7106, 5299, 4660, 276,
00114         4389, 3783, 1591, 1601,
00115         749, 1387, 1661, 3210,
00116         708,
00117         4226, 5604, 6375, 3056,
00118         1772, 3662, 4401, 5218,
00119         2838, 6913, 1685, 1194,
00120         6963, 5001, 6694, 991,
00121         7489, 2441, 639, 2097,
00122         2498, 6470, 2399, 242,
00123         3768, 1186,
00124         5246, 4259, 5907, 3870,
00125         3262, 7387, 3069, 2999,
00126         7993, 7849, 4157, 5031,
00127         5986, 4833, 5739, 7846,
00128         898, 2022, 7446, 6404,
00129         155, 7862, 7795, 6121,
00130         4840, 6585, 429, 6020,
00131         200, 1664, 1499, 7298,
00132         1305, 7323, 7544, 4438,
00133         2485, 3387, 7319, 1853,
00134         5781, 1874, 7555, 2132,
00135         6441, 6722, 1192, 2588,
00136         2188, 297, 1540, 4138,
00137         5231, 4789, 659, 871,
00138         6837, 1393, 7383, 611,
00139         4920, 5416, 1611, 2474,
00140         118, 1382, 1092, 7950,
00141         7223, 1769, 4721, 1252,
00142         5147, 2165, 7897, 4054,
00143         3498, 6571, 2858, 8126,
00144         7017, 1901, 181, 1114,
00145         5195, 7479, 4186, 3904,
00146         7128, 1396, 4513, 5967,
00147         2580, 2575, 7961, 2598,
00148         4508, 2090, 3685, 7748,
00149         684, 913, 5558, 2894,
00150         5858, 6432, 3813, 3573,
00151         7523, 5280, 3376, 7424,
00152         2918, 5793, 1747, 7079,
00153         2921, 2490, 4119, 3373,
00154         977, 681, 4273, 5419,
00155         5626, 1266, 5804, 2414,
00156         6444, 4757, 427, 5452,
00157         5182, 6606, 6531, 4268,
00158         3115, 6835, 862, 4856,
00159         2765, 37, 1943, 7977,
00160         2512, 4451, 4071};
00161
00162 constexpr int32_t GPS_L5_CNAV_DATA_PAGE_BITS = 300;    //!< GPS L5 CNAV page length, including preamble
00163               and CRC [bits]
00164 constexpr int32_t GPS_L5_SYMBOLS_PER_BIT = 2;
00165 constexpr int32_t GPS_L5_SAMPLES_PER_SYMBOL = 10;
00166 constexpr int32_t GPS_L5_CNAV_DATA_PAGE_SYMBOLS = 600;
00167 constexpr int32_t GPS_L5_CNAV_DATA_PAGE_DURATION_S = 6;
00168 constexpr int32_t GPS_L5I_NH_CODE_LENGTH = 10;
00169 constexpr int32_t GPS_L5I_NH_CODE[10] = {0, 0, 0, 0, 1, 1, 0, 1, 0, 1};
00170 constexpr int32_t GPS_L5Q_NH_CODE_LENGTH = 20;
00171 constexpr int32_t GPS_L5Q_NH_CODE[20] = {0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0};
00172 constexpr char GPS_L5I_NH_CODE_STR[11] = "0000110101";
00173 constexpr char GPS_L5Q_NH_CODE_STR[21] = "00000100110101001110";
00174
00175 /** @} */
00176 /** @} */
00177 #endif    // GNSS_SDR_GPS_L5_H

```

## 11.779 gps\_navigation\_message.h File Reference

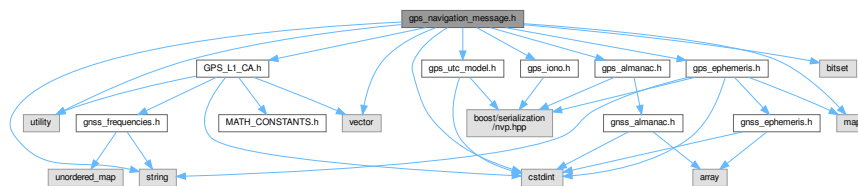
Interface of a GPS NAV Data message decoder.

```

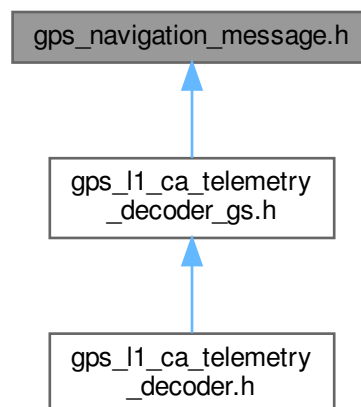
#include "GPS_L1_CA.h"
#include "gps_almanac.h"
#include "gps_ephemeris.h"
#include "gps_iono.h"
#include "gps_utc_model.h"
#include <bitset>
#include <stdint>
#include <map>
#include <string>
#include <utility>
#include <vector>

```

Include dependency graph for `gps_navigation_message.h`:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Gps\\_Navigation\\_Message](#)

*This class decodes a GPS NAV Data message as described in IS-GPS-200M.*

## 11.779.1 Detailed Description

Interface of a GPS NAV Data message decoder.

## Author

Javier Arribas, 2011. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gps\\_navigation\\_message.h](#).

---

## 11.780 gps\_navigation\_message.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file gps_navigation_message.h
00003  * \brief Interface of a GPS NAV Data message decoder
00004  * \author Javier Arribas, 2011. jarribas(at)cttc.es
00005  *
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018
00019 #ifndef GNSS_SDR_GPS_NAVIGATION_MESSAGE_H
00020 #define GNSS_SDR_GPS_NAVIGATION_MESSAGE_H
00021
00022
00023 #include "GPS_L1_CA.h"
00024 #include "gps_almanac.h"
00025 #include "gps_ephemeris.h"
00026 #include "gps_iono.h"
00027 #include "gps_utc_model.h"
00028 #include <bitset>
00029 #include <cstdint>
00030 #include <map>
00031 #include <string>
00032 #include <utility> // for pair
00033 #include <vector>
00034
00035 /** \addtogroup Core
00036  * \{ */
00037 /** \addtogroup System_Parameters
00038  * \{ */
00039
00040
00041  /*!
00042  * \brief This class decodes a GPS NAV Data message as described in IS-GPS-200M
00043  *
00044  * See https://www.gps.gov/technical/icwg/IS-GPS-200M.pdf Appendix II
00045  */
00046  class Gps_Navigation_Message
00047  {
00048  public:
00049      /*!
00050       * Default constructor
00051       */
00052      Gps_Navigation_Message();
00053
00054      /*!
00055       * \brief Obtain a GPS SV Ephemeris class filled with current SV data
00056       */
00057      Gps_Ephemeris get_ephemeris() const;
00058
00059      /*!
00060       * \brief Obtain a GPS ionospheric correction parameters class filled with current SV data
00061       */
00062      Gps_Iono get_iono();
00063
00064      /*!
00065       * \brief Obtain a GPS almanac class filled with current SV data
00066       */
00067      Gps_Almanac get_almanac();
00068
00069      /*!
00070       * \brief Obtain a GPS Almanac model parameters class filled with current SV data
00071       */
00072      Gps_Utc_Model get_utc_model();

```

```

00073
00074     /*!
00075      * \brief Decodes the GPS NAV message
00076      */
00077     int32_t subframe_decoder(const char* subframe);
00078
00079     /*!
00080      * \brief Computes the Coordinated Universal Time (UTC) and
00081      * returns it in [s] (IS-GPS-200M, 20.3.3.5.2.4)
00082      */
00083     double utc_time(double gpstime_corrected) const;
00084
00085     /*!
00086      * \brief Gets Time of Week, in seconds
00087      */
00088     inline int32_t get_TOW() const
00089     {
00090         return d_TOW;
00091     }
00092
00093     /*!
00094      * \brief Sets Time of Week, in seconds
00095      */
00096     inline int32_t get_GPS_week() const
00097     {
00098         return i_GPS_week;
00099     }
00100
00101     /*!
00102      * \brief Sets satellite PRN number
00103      */
00104     inline void set_satellite_PRN(uint32_t prn)
00105     {
00106         i_satellite_PRN = prn;
00107     }
00108
00109     /*!
00110      * \brief Gets satellite PRN number
00111      */
00112     inline uint32_t get_satellite_PRN() const
00113     {
00114         return i_satellite_PRN;
00115     }
00116
00117     /*!
00118      * \brief Sets channel ID
00119      */
00120     inline void set_channel(int32_t channel_id)
00121     {
00122         i_channel_ID = channel_id;
00123     }
00124
00125     /*!
00126      * \brief Gets flag_iono_valid
00127      */
00128     inline bool get_flag_iono_valid() const
00129     {
00130         return flag_iono_valid;
00131     }
00132
00133     /*!
00134      * \brief Gets flag_utc_model_valid
00135      */
00136     inline bool get_flag_utc_model_valid() const
00137     {
00138         return flag_utc_model_valid;
00139     }
00140
00141     bool satellite_validation();
00142     bool almanac_validation() const;
00143
00144 private:
00145     uint64_t read_navigation_unsigned(const std::bitset<GPS_SUBFRAME_BITS>& bits, const
std::vector<std::pair<int32_t, int32_t>& parameter) const;
00146     int64_t read_navigation_signed(const std::bitset<GPS_SUBFRAME_BITS>& bits, const
std::vector<std::pair<int32_t, int32_t>& parameter) const;
00147     bool read_navigation_bool(const std::bitset<GPS_SUBFRAME_BITS>& bits, const
std::vector<std::pair<int32_t, int32_t>& parameter) const;
00148     void print_gps_word_bytes(uint32_t GPS_word) const;
00149
00150     std::map<int32_t, int32_t> almanacHealth; //!< Map that stores the health information stored in
the almanac
00151
00152     std::map<int32_t, std::string> satelliteBlock; //!< Map that stores to which block the PRN
belongs https://www.navcen.uscg.gov/?Do=constellationStatus
00153
00154     // broadcast orbit 1

```

```

00155     int32_t d_TOW{};           // Time of GPS Week of the ephemeris set (taken from subframes TOW) [s]
00156     int32_t d_TOW_SF1{};      // Time of GPS Week from HOW word of Subframe 1 [s]
00157     int32_t d_TOW_SF2{};      // Time of GPS Week from HOW word of Subframe 2 [s]
00158     int32_t d_TOW_SF3{};      // Time of GPS Week from HOW word of Subframe 3 [s]
00159     int32_t d_TOW_SF4{};      // Time of GPS Week from HOW word of Subframe 4 [s]
00160     int32_t d_TOW_SF5{};      // Time of GPS Week from HOW word of Subframe 5 [s]
00161     int32_t d_IODE_SF2{};
00162     int32_t d_IODE_SF3{};
00163     double d_Crs{};           // Amplitude of the Sine Harmonic Correction Term to the Orbit Radius [m]
00164     double d_Delta_n{};       // Mean Motion Difference From Computed Value [semi-circles/s]
00165     double d_M_0{};           // Mean Anomaly at Reference Time [semi-circles]
00166     // broadcast orbit 2
00167     double d_Cuc{};           // Amplitude of the Cosine Harmonic Correction Term to the Argument of
Latitude [rad]
00168     double d_e_eccentricity{}; // Eccentricity [dimensionless]
00169     double d_Cus{};           // Amplitude of the Sine Harmonic Correction Term to the Argument of
Latitude [rad]
00170     double d_sqrt_A{};        // Square Root of the Semi-Major Axis [sqrt(m)]
00171     // broadcast orbit 3
00172     int32_t d_Toe{};           // Ephemeris data reference time of week (Ref. 20.3.3.4.3 IS-GPS-200M) [s]
00173     int32_t d_Toc{};           // clock data reference time (Ref. 20.3.3.3.1 IS-GPS-200M) [s]
00174     double d_Cic{};           // Amplitude of the Cosine Harmonic Correction Term to the Angle of
Inclination [rad]
00175     double d_OMEGA0{};         // Longitude of Ascending Node of Orbit Plane at Weekly Epoch [semi-circles]
00176     double d_Cis{};           // Amplitude of the Sine Harmonic Correction Term to the Angle of Inclination
[rad]
00177     // broadcast orbit 4
00178     double d_i_0{};           // Inclination Angle at Reference Time [semi-circles]
00179     double d_Crc{};           // Amplitude of the Cosine Harmonic Correction Term to the Orbit Radius [m]
00180     double d_OMEGA{};         // Argument of Perigee [semi-circles]
00181     double d_OMEGA_DOT{};     // Rate of Right Ascension [semi-circles/s]
00182     // broadcast orbit 5
00183     double d_IDOT{};           // Rate of Inclination Angle [semi-circles/s]
00184     int32_t i_code_on_L2{};    // If 1, P code ON in L2; if 2, C/A code ON in L2;
00185     int32_t i_GPS_week{};      // GPS week number, aka WN [week]
00186     bool b_L2_P_data_flag{};   // When true, indicates that the NAV data stream was commanded OFF on
the P-code of the L2 channel
00187     // broadcast orbit 6
00188     int32_t i_SV_accuracy{};   // User Range Accuracy (URA) index of the SV (reference paragraph 6.2.1)
for the standard positioning service user (Ref 20.3.3.3.1.3 IS-GPS-200M)
00189     int32_t i_SV_health{};
00190     double d_TGD{};           // Estimated Group Delay Differential: L1-L2 correction term only for the
benefit of "L1 P(Y)" or "L2 P(Y)" s users [s]
00191     int32_t d_IODC{};         // Issue of Data, Clock
00192     // broadcast orbit 7
00193     int32_t i_AODO{};         // Age of Data Offset (AODO) term for the navigation message
correction table (NMCT) contained in subframe 4 (reference paragraph 20.3.3.5.1.9) [s]
00194     bool b_fit_interval_flag{}; // indicates the curve-fit interval used by the CS (Block
II/IIA/IIR/IIR-M/IIF) and SS (Block IIIA) in determining the ephemeris parameters, as follows: 0 = 4
hours, 1 = greater than 4 hours.
00195     double d_spare1{};
00196     double d_spare2{};
00197     double d_A_f0{};          // Coefficient 0 of code phase offset model [s]
00198     double d_A_f1{};          // Coefficient 1 of code phase offset model [s/s]
00199     double d_A_f2{};          // Coefficient 2 of code phase offset model [s/s^2]
00200
00201     // Almanac
00202     int32_t i_ToA{};           // Almanac reference time [s]
00203     int32_t i_WN_A{};         // Modulo 256 of the GPS week number to which the almanac reference time
(i_ToA) is referenced
00204     int32_t SV_Health{};      // Almanac SV health
00205     uint32_t a_PRN{};          // Almanac PRN
00206     double a_delta_i{};        // Inclination Angle at Reference Time (relative to i_0 = 0.30 semi-circles)
00207     double a_M_0{};           // Mean Anomaly at Reference Time [semi-circles]
00208     double a_ecc{};           // Eccentricity [dimensionless]
00209     double a_sqrt_A{};         // Square Root of the Semi-Major Axis [sqrt(m)]
00210     double a_OMEGA_0{};        // Longitude of Ascending Node of Orbit Plane at Weekly Epoch [semi-circles]
00211     double a_omega{};          // Argument of Perigee [semi-circles]
00212     double a_OMEGA_dot{};       // Rate of Right Ascension [semi-circles/s]
00213     double a_af0{};           // Coefficient 0 of code phase offset model [s]
00214     double a_af1{};           // Coefficient 1 of code phase offset model [s/s]
00215
00216     // satellite identification info
00217     int32_t i_channel_ID{};
00218     uint32_t i_satellite_PRN{};
00219
00220     // Ionospheric parameters
00221     double d_alpha0{};         // Coefficient 0 of a cubic equation representing the amplitude of the
vertical delay [s]
00222     double d_alpha1{};         // Coefficient 1 of a cubic equation representing the amplitude of the
vertical delay [s/semi-circle]
00223     double d_alpha2{};         // Coefficient 2 of a cubic equation representing the amplitude of the
vertical delay [s(semi-circle)^2]
00224     double d_alpha3{};         // Coefficient 3 of a cubic equation representing the amplitude of the
vertical delay [s(semi-circle)^3]
00225     double d_beta0{};          // Coefficient 0 of a cubic equation representing the period of the model [s]
00226     double d_beta1{};          // Coefficient 1 of a cubic equation representing the period of the model

```

```

    [s/semi-circle]
00227     double d_beta2{};    // Coefficient 2 of a cubic equation representing the period of the model
    [s(semi-circle)^2]
00228     double d_beta3{};    // Coefficient 3 of a cubic equation representing the period of the model
    [s(semi-circle)^3]
00229
00230     // UTC parameters
00231     double d_A0{};    // Constant of a model that relates GPS and UTC time (ref. 20.3.3.5.2.4
IS-GPS-200M) [s]
00232     double d_A1{};    // 1st order term of a model that relates GPS and UTC time (ref. 20.3.3.5.2.4
IS-GPS-200M) [s/s]
00233
00234     int32_t d_t_OT{};    // Reference time for UTC data (reference 20.3.4.5 and 20.3.3.5.2.4
IS-GPS-200M) [s]
00235     int32_t i_WN_T{};    // UTC reference week number [weeks]
00236     int32_t d_DeltaT_LS{}; // delta time due to leap seconds [s]. Number of leap seconds since
6-Jan-1980 as transmitted by the GPS almanac.
00237     int32_t i_WN_LSF{};    // Week number at the end of which the leap second becomes effective
[weeks]
00238     int32_t i_DN{};    // Day number (DN) at the end of which the leap second becomes effective
[days]
00239     int32_t d_DeltaT_LSF{}; // Scheduled future or recent past (relative to NAV message upload) value
of the delta time due to leap seconds [s]
00240
00241     // Flags
00242     bool b_valid_ephemeris_set_flag{}; // flag indicating that this ephemeris set have passed the
validation check
00243     bool flag_iono_valid{};    // If set, it indicates that the ionospheric parameters are
filled (page 18 has arrived and decoded)
00244     bool flag_utc_model_valid{}; // If set, it indicates that the UTC model parameters are
filled
00245     bool flag_almanac_valid{};    // If set, it indicates that the almanac is filled
00246     bool flag_almanac_week_valid{}; // If set, it indicates that the almanac week is valid
00247     /* If true, enhanced level of integrity assurance.
00248     *
00249     * If false, indicates that the conveying signal is provided with the legacy level of integrity
assurance.
00250     * That is, the probability that the instantaneous URE of the conveying signal exceeds 4.42 times
the upper bound
00251     * value of the current broadcast URA index, for more than 5.2 seconds, without an accompanying
alert, is less
00252     * than 1E-5 per hour. If true, indicates that the conveying signal is provided with an enhanced
level of
00253     * integrity assurance. That is, the probability that the instantaneous URE of the conveying
signal exceeds 5.73
00254     * times the upper bound value of the current broadcast URA index, for more than 5.2 seconds,
without an
00255     * accompanying alert, is less than 1E-8 per hour.
00256     */
00257     bool b_integrity_status_flag{};
00258     bool b_alert_flag{};    // If true, indicates that the SV URA may be worse than indicated in
d_SV_accuracy, use that SV at our own risk.
00259     bool b_antispoofing_flag{}; // If true, the AntiSpoofing mode is ON in that SV
00260 };
00261
00262 /** \} */
00263 /** \} */
00264 /** \} */
00265 #endif // GNSS_SDR_GPS_NAVIGATION_MESSAGE_H

```

## 11.781 gps\_utc\_model.h File Reference

Interface of a GPS UTC MODEL storage.

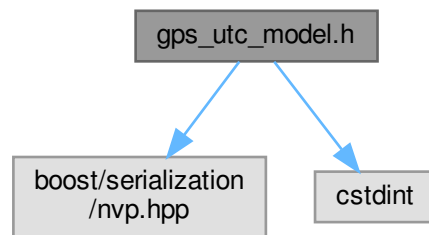
```

#include <boost/serialization/nvp.hpp>
#include <cstdint>

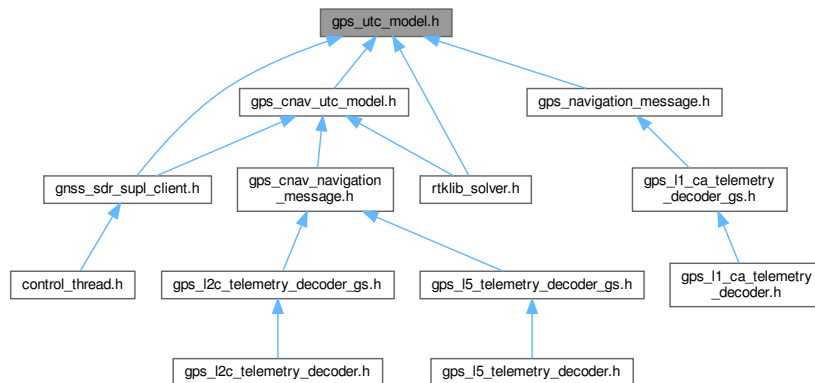
```



Include dependency graph for gps\_utc\_model.h:



This graph shows which files directly or indirectly include this file:



## Classes

- class [Gps\\_Utc\\_Model](#)

*This class is a storage for the GPS UTC MODEL data as described in IS-GPS-200M.*

### 11.781.1 Detailed Description

Interface of a GPS UTC MODEL storage.

#### Author

Javier Arribas, 2013. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [gps\\_utc\\_model.h](#).

---

## 11.782 gps\_utc\_model.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file gps_utc_model.h
  
```

```

00003  * \brief Interface of a GPS UTC MODEL storage
00004  * \author Javier Arribas, 2013. jarribas(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_GPS_UTC_MODEL_H
00019 #define GNSS_SDR_GPS_UTC_MODEL_H
00020
00021 #include <boost/serialization/nvp.hpp>
00022 #include <cstdint>
00023
00024 /** \addtogroup Core
00025  * \{ */
00026 /** \addtogroup System_Parameters
00027  * \{ */
00028
00029
00030 /*!
00031  * \brief This class is a storage for the GPS UTC MODEL data as described in IS-GPS-200M
00032  *
00033  * See https://www.gps.gov/technical/icwg/IS-GPS-200M.pdf Appendix II
00034  */
00035 class Gps_Utc_Model
00036 {
00037 public:
00038     /*!
00039      * Default constructor
00040      */
00041     Gps_Utc_Model() = default;
00042
00043     // UTC parameters
00044     double A0{};          //!< Constant of a model that relates GPS and UTC time (ref. 20.3.3.5.2.4
IS-GPS-200M) [s]
00045     double A1{};          //!< 1st order term of a model that relates GPS and UTC time (ref.
20.3.3.5.2.4 IS-GPS-200M) [s/s]
00046     double A2{};          //!< 2nd order term of a model that relates GPS and UTC time (ref.
20.3.3.5.2.4 IS-GPS-200M) [s/s]
00047     int32_t tot{};        //!< Reference time for UTC data (reference 20.3.4.5 and 20.3.3.5.2.4
IS-GPS-200M) [s]
00048     int32_t WN_T{};       //!< UTC reference week number [weeks]
00049     int32_t DeltaT_LS{};  //!< Delta time due to leap seconds [s]. Number of leap seconds since
6-Jan-1980 as transmitted by the GPS almanac.
00050     int32_t WN_LSF{};     //!< Week number at the end of which the leap second becomes effective
[weeks]
00051     int32_t DN{};         //!< Day number (DN) at the end of which the leap second becomes effective
[days]
00052     int32_t DeltaT_LSF{}; //!< Scheduled future or recent past (relative to NAV message upload) value
of the delta time due to leap seconds [s]
00053
00054     bool valid{};
00055
00056     template <class Archive>
00057     /*
00058      * \brief Serialize is a boost standard method to be called by the boost XML
00059      * serialization. Here is used to save the ephemeris data on disk file.
00060      */
00061     inline void serialize(Archive& archive, const uint32_t version)
00062     {
00063         if (version)
00064         {
00065             archive& BOOST_SERIALIZATION_NVP(A0);
00066             archive& BOOST_SERIALIZATION_NVP(A1);
00067             archive& BOOST_SERIALIZATION_NVP(A2);
00068             archive& BOOST_SERIALIZATION_NVP(tot);
00069             archive& BOOST_SERIALIZATION_NVP(WN_T);
00070             archive& BOOST_SERIALIZATION_NVP(DeltaT_LS);
00071             archive& BOOST_SERIALIZATION_NVP(WN_LSF);
00072             archive& BOOST_SERIALIZATION_NVP(DN);
00073             archive& BOOST_SERIALIZATION_NVP(DeltaT_LSF);
00074             archive& BOOST_SERIALIZATION_NVP(valid);
00075         }
00076     };
00077
00078
00079
00080 /** \} */
00081 /** \} */

```

## 11.783 MATH\_CONSTANTS.h File Reference

- constexpr double **GNSS\_OMEGA\_EARTH\_DOT** = 7.2921151467e-5  
Default Earth rotation rate, [rad/s].
- constexpr double **SPEED\_OF\_LIGHT\_M\_S** = 299792458.0  
Speed of light in vacuum [m/s].
- constexpr double **SPEED\_OF\_LIGHT\_M\_MS** = 299792.4580  
Speed of light in vacuum [m/ms].
- constexpr double **GPS\_GM** = 3.986005e14  
Universal gravitational constant times the mass of the Earth, [ $m^3/s^2$ ] IS-GPS-200M, 20.3.3.3.3.1.
- constexpr double **GPS\_F** = -4.442807633e-10  
Constant, [ $s/(m)^{1/2}$ ], IS-GPS-200M, 20.3.3.3.3.1.
- constexpr double **GALILEO\_GM** = 3.986004418e14  
Geocentric gravitational constant [ $m^3/s^2$ ], OS SIS ICD v2.0, pag. 44.
- constexpr double **GALILEO\_F** = -4.442807309e-10  
Constant, [ $s/(m)^{1/2}$ ]. OS SIS ICD v2.0, pag. 47.
- constexpr double **GLONASS\_OMEGA\_EARTH\_DOT** = 7.292115e-5  
Earth rotation rate, [rad/s] ICD L1, L2 GLONASS Edition 5.1 2008 pag. 55.
- constexpr double **GLONASS\_GM** = 398600.44e9  
Universal gravitational constant times the mass of the Earth, [ $m^3/s^2$ ].
- constexpr double **BEIDOU\_OMEGA\_EARTH\_DOT** = 7.2921150e-5  
Earth rotation rate, [rad/s] as defined in BDS-SIS-ICD-B11-3.0 2019-02, pag. 3.
- constexpr double **BEIDOU\_GM** = 3.986004418e14  
Universal gravitational constant times the mass of the Earth, [ $m^3/s^2$ ] as defined in CGCS2000.
- constexpr double **BEIDOU\_F** = -4.442807309e-10  
Constant, [ $s/(m)^{1/2}$ ]  $F = -2(GM)^{.5}/C^2$ .
- constexpr double **GNSS\_PI** = 3.1415926535898  
 $\pi$  constant as defined for GNSS
- constexpr double **HALF\_PI** = **GNSS\_PI** / 2.0  
 $\pi/2$
- constexpr double **TWO\_PI** = 2.0 \* **GNSS\_PI**  
 $2 * \pi$
- constexpr double **TWO\_P3** = 8.0  
 $2^3$
- constexpr double **TWO\_P4** = 16.0  
 $2^4$
- constexpr double **TWO\_P8** = 256.0  
 $2^8$

- constexpr double [TWO\\_P11](#) = 2048.0  
 $2^{11}$
- constexpr double [TWO\\_P12](#) = 4096.0  
 $2^{12}$
- constexpr double [TWO\\_P14](#) = 16384.0  
 $2^{14}$
- constexpr double [TWO\\_P16](#) = 65536.0  
 $2^{16}$
- constexpr double [TWO\\_P19](#) = 524288.0  
 $2^{19}$
- constexpr double [TWO\\_P31](#) = 2147483648.0  
 $2^{31}$
- constexpr double [TWO\\_P32](#) = 4294967296.0  
 $2^{32}$
- constexpr double [TWO\\_P56](#) = 7.205759403792794e+016  
 $2^{56}$
- constexpr double [TWO\\_P57](#) = 1.441151880758559e+017  
 $2^{57}$
- constexpr double [TWO\\_N2](#) = 0.25  
 $2^{-2}$
- constexpr double [TWO\\_N5](#) = 0.03125  
 $2^{-5}$
- constexpr double [TWO\\_N6](#) = 0.015625  
 $2^{-6}$
- constexpr double [TWO\\_N8](#) = 0.00390625  
 $2^{-8}$
- constexpr double [TWO\\_N9](#) = 0.001953125  
 $2^{-9}$
- constexpr double [TWO\\_N10](#) = 0.0009765625  
 $2^{-10}$
- constexpr double [TWO\\_N11](#) = 4.882812500000000e-004  
 $2^{-11}$
- constexpr double [TWO\\_N14](#) = 0.00006103515625  
 $2^{-14}$
- constexpr double [TWO\\_N15](#) = 3.051757812500000e-005  
 $2^{-15}$
- constexpr double [TWO\\_N16](#) = 1.525878906250000e-005  
 $2^{-16}$
- constexpr double [TWO\\_N17](#) = 7.629394531250000e-006  
 $2^{-17}$
- constexpr double [TWO\\_N18](#) = 3.814697265625000e-006  
 $2^{-18}$
- constexpr double [TWO\\_N19](#) = 1.907348632812500e-006  
 $2^{-19}$
- constexpr double [TWO\\_N20](#) = 9.536743164062500e-007  
 $2^{-20}$
- constexpr double [TWO\\_N21](#) = 4.768371582031250e-007  
 $2^{-21}$
- constexpr double [TWO\\_N22](#) = 2.384185791015625e-007  
 $2^{-22}$
- constexpr double [TWO\\_N23](#) = 1.192092895507810e-007

- $2^{-23}$ 
  - `constexpr double TWO_N24` = 5.960464477539063e-008
- $2^{-24}$ 
  - `constexpr double TWO_N25` = 2.980232238769531e-008
- $2^{-25}$ 
  - `constexpr double TWO_N26` = 1.490116119384765e-009
- $2^{-26}$ 
  - `constexpr double TWO_N27` = 7.450580596923828e-009
- $2^{-27}$ 
  - `constexpr double TWO_N29` = 1.862645149230957e-009
- $2^{-29}$ 
  - `constexpr double TWO_N30` = 9.313225746154785e-010
- $2^{-30}$ 
  - `constexpr double TWO_N31` = 4.656612873077393e-010
- $2^{-31}$ 
  - `constexpr double TWO_N32` = 2.328306436538696e-010
- $2^{-32}$ 
  - `constexpr double TWO_N33` = 1.164153218269348e-010
- $2^{-33}$ 
  - `constexpr double TWO_N34` = 5.82076609134674e-011
- $2^{-34}$ 
  - `constexpr double TWO_N35` = 2.91038304567337e-011
- $2^{-35}$ 
  - `constexpr double TWO_N38` = 3.637978807091713e-012
- $2^{-38}$ 
  - `constexpr double TWO_N39` = 1.818989403545856e-012
- $2^{-39}$ 
  - `constexpr double TWO_N40` = 9.094947017729280e-013
- $2^{-40}$ 
  - `constexpr double TWO_N43` = 1.136868377216160e-013
- $2^{-43}$ 
  - `constexpr double TWO_N44` = 5.684341886080802e-14
- $2^{-44}$ 
  - `constexpr double TWO_N46` = 1.4210854715202e-014
- $2^{-46}$ 
  - `constexpr double TWO_N48` = 3.552713678800501e-15
- $2^{-46}$ 
  - `constexpr double TWO_N50` = 8.881784197001252e-016
- $2^{-50}$ 
  - `constexpr double TWO_N51` = 4.44089209850063e-016
- $2^{-51}$ 
  - `constexpr double TWO_N55` = 2.775557561562891e-017
- $2^{-55}$ 
  - `constexpr double TWO_N57` = 6.938893903907228e-18
- $2^{-57}$ 
  - `constexpr double TWO_N59` = 1.73472347597681e-018
- $2^{-59}$ 
  - `constexpr double TWO_N60` = 8.673617379884036e-19
- $2^{-60}$ 
  - `constexpr double TWO_N66` = 1.3552527156068805425093160010874271392822265625e-20
- $2^{-66}$

- `constexpr double TWO_N68 = 3.388131789017201e-21`  
 $2^{-68}$
- `constexpr double PI_TWO_N19 = 5.992112452678286e-006`  
 $\pi \cdot 2^{-19}$ .
- `constexpr double PI_TWO_N43 = 3.571577341960839e-013`  
 $\pi \cdot 2^{-43}$ .
- `constexpr double PI_TWO_N31 = 1.462918079267160e-009`  
 $\pi \cdot 2^{-31}$ .
- `constexpr double PI_TWO_N38 = 1.142904749427469e-011`  
 $\pi \cdot 2^{-38}$ .
- `constexpr double PI_TWO_N23 = 3.745070282923929e-007`  
 $\pi \cdot 2^{-23}$ .
- `constexpr double D2R = GNSS_PI / 180.0`  
*deg to rad*
- `constexpr double R2D = 180.0 / GNSS_PI`  
*rad to deg*
- `constexpr double SC2RAD = GNSS_PI`  
*semi-circle to radian (IS-GPS)*
- `constexpr double AS2R = D2R / 3600.0`  
*arc sec to radian*
- `constexpr double AU = 149597870691.0`  
*1 Astronomical Unit AU (m) distance from Earth to the Sun.*

### 11.783.1 Detailed Description

Defines useful mathematical constants and their scaled versions.

Author

Javier Arribas, 2013. jarribas(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [MATH\\_CONSTANTS.h](#).

---

## 11.784 MATH\_CONSTANTS.h

[Go to the documentation of this file.](#)

```
00001 /*!
00002  * \file MATH_CONSTANTS.h
00003  * \brief Defines useful mathematical constants and their scaled versions
00004  * \author Javier Arribas, 2013. jarribas(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_MATH_CONSTANTS_H
00018 #define GNSS_SDR_MATH_CONSTANTS_H
00019
00020 /** \addtogroup Core
00021  * \{ */
00022 /** \addtogroup System_Parameters
00023  * \{ */
00024
00025
00026 constexpr double GNSS_OMEGA_EARTH_DOT = 7.2921151467e-5; //!< Default Earth rotation rate, [rad/s]
00027 constexpr double SPEED_OF_LIGHT_M_S = 299792458.0;      //!< Speed of light in vacuum [m/s]
```

```

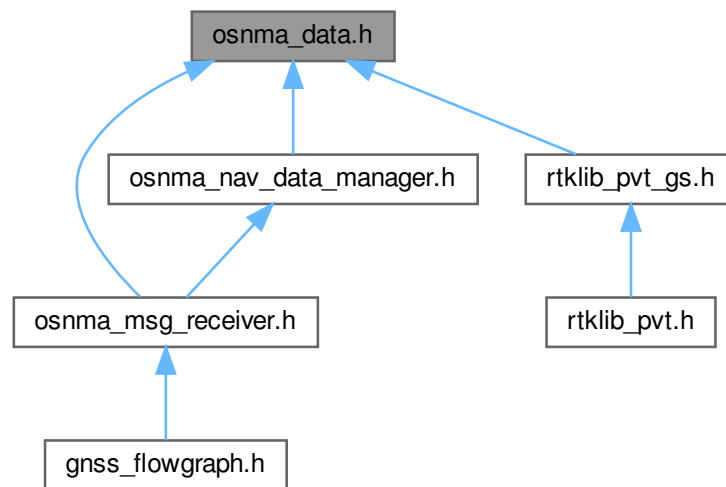
00028 constexpr double SPEED_OF_LIGHT_M_MS = 299792.4580;          //!< Speed of light in vacuum [m/ms]
00029
00030 // Physical constants for GPS
00031 constexpr double GPS_GM = 3.986005e14;          //!< Universal gravitational constant times the mass of
the Earth, [m^3/s^2] IS-GPS-200M, 20.3.3.3.1
00032 constexpr double GPS_F = -4.442807633e-10;      //!< Constant, [s/(m)^(1/2)], IS-GPS-200M, 20.3.3.3.1
00033
00034 // Physical constants for Galileo
00035 constexpr double GALILEO_GM = 3.986004418e14;    //!< Geocentric gravitational constant[m^3/s^2], OS
SIS ICD v2.0, pag. 44
00036 constexpr double GALILEO_F = -4.442807309e-10;    //!< Constant, [s/(m)^(1/2)]. OS SIS ICD v2.0, pag. 47
00037
00038 // Physical constants for GLONASS
00039 constexpr double GLONASS_OMEGA_EARTH_DOT = 7.292115e-5;  //!< Earth rotation rate, [rad/s] ICD L1, L2
GLONASS Edition 5.1 2008 pag. 55
00040 constexpr double GLONASS_GM = 398600.44e9;          //!< Universal gravitational constant times
the mass of the Earth, [m^3/s^2]
00041
00042 // Physical constants for Beidou
00043 constexpr double BEIDOU_OMEGA_EARTH_DOT = 7.2921150e-5;  //!< Earth rotation rate, [rad/s] as defined
in BDS-SIS-ICD-B1I-3.0 2019-02, pag. 3
00044 constexpr double BEIDOU_GM = 3.986004418e14;          //!< Universal gravitational constant times
the mass of the Earth, [m^3/s^2] as defined in CGCS2000
00045 constexpr double BEIDOU_F = -4.442807309e-10;          //!< Constant, [s/(m)^(1/2)] F=-2(GM)^.5/C^2
00046
00047 constexpr double GNSS_PI = 3.1415926535898;          //!< pi constant as defined for GNSS
00048 constexpr double HALF_PI = GNSS_PI / 2.0;           //!< pi/2
00049 constexpr double TWO_PI = 2.0 * GNSS_PI;             //!< 2 * pi
00050
00051
00052 // Constants for scaling the ephemeris found in the data message
00053 // the format is the following: TWO_N5 -> 2^-5, TWO_P4 -> 2^4, PI_TWO_N43 -> Pi*2^-43, etc etc
00054 // Additionally some of the PI*2^N terms are used in the tracking stuff
00055 // TWO_PX ==> 2^X
00056 // TWO_NX ==> 2^-X
00057 // PI_TWO_NX ==> Pi*2^-X
00058
00059 constexpr double TWO_P3 = 8.0;                       //!< 2^3
00060 constexpr double TWO_P4 = 16.0;                      //!< 2^4
00061 constexpr double TWO_P8 = 256.0;                    //!< 2^8
00062 constexpr double TWO_P11 = 2048.0;                   //!< 2^11
00063 constexpr double TWO_P12 = 4096.0;                  //!< 2^12
00064 constexpr double TWO_P14 = 16384.0;                 //!< 2^14
00065 constexpr double TWO_P16 = 65536.0;                 //!< 2^16
00066 constexpr double TWO_P19 = 524288.0;                //!< 2^19
00067 constexpr double TWO_P31 = 2147483648.0;            //!< 2^31
00068 constexpr double TWO_P32 = 4294967296.0;            //!< 2^32
00069 constexpr double TWO_P56 = 7.205759403792794e+016;  //!< 2^56
00070 constexpr double TWO_P57 = 1.441151880758559e+017;  //!< 2^57
00071
00072 constexpr double TWO_N2 = 0.25;                     //!< 2^-2
00073 constexpr double TWO_N5 = 0.03125;                  //!< 2^-5
00074 constexpr double TWO_N6 = 0.015625;                 //!< 2^-6
00075 constexpr double TWO_N8 = 0.00390625;              //!< 2^-8
00076 constexpr double TWO_N9 = 0.001953125;             //!< 2^-9
00077 constexpr double TWO_N10 = 0.0009765625;           //!< 2^-10
00078 constexpr double TWO_N11 = 4.8828125000000000e-004; //!< 2^-11
00079 constexpr double TWO_N14 = 0.00006103515625;       //!< 2^-14
00080 constexpr double TWO_N15 = 3.0517578125000000e-005; //!< 2^-15
00081 constexpr double TWO_N16 = 1.5258789062500000e-005; //!< 2^-16
00082 constexpr double TWO_N17 = 7.6293945312500000e-006; //!< 2^-17
00083 constexpr double TWO_N18 = 3.8146972656250000e-006; //!< 2^-18
00084 constexpr double TWO_N19 = 1.9073486328125000e-006; //!< 2^-19
00085 constexpr double TWO_N20 = 9.5367431640625000e-007; //!< 2^-20
00086 constexpr double TWO_N21 = 4.7683715820312500e-007; //!< 2^-21
00087 constexpr double TWO_N22 = 2.384185791015625e-007; //!< 2^-22
00088 constexpr double TWO_N23 = 1.192092895507810e-007; //!< 2^-23
00089 constexpr double TWO_N24 = 5.960464477539063e-008; //!< 2^-24
00090 constexpr double TWO_N25 = 2.980232238769531e-008; //!< 2^-25
00091 constexpr double TWO_N26 = 1.490116119384765e-009; //!< 2^-26
00092 constexpr double TWO_N27 = 7.450580596923828e-009; //!< 2^-27
00093 constexpr double TWO_N29 = 1.862645149230957e-009; //!< 2^-29
00094 constexpr double TWO_N30 = 9.313225746154785e-010; //!< 2^-30
00095 constexpr double TWO_N31 = 4.656612873077393e-010; //!< 2^-31
00096 constexpr double TWO_N32 = 2.328306436538696e-010; //!< 2^-32
00097 constexpr double TWO_N33 = 1.164153218269348e-010; //!< 2^-33
00098 constexpr double TWO_N34 = 5.82076609134674e-011;  //!< 2^-34
00099 constexpr double TWO_N35 = 2.91038304567337e-011;  //!< 2^-35
00100 constexpr double TWO_N38 = 3.637978807091713e-012;  //!< 2^-38
00101 constexpr double TWO_N39 = 1.818989403545856e-012;  //!< 2^-39
00102 constexpr double TWO_N40 = 9.094947017729280e-013;  //!< 2^-40
00103 constexpr double TWO_N43 = 1.136868377216160e-013;  //!< 2^-43
00104 constexpr double TWO_N44 = 5.684341886080802e-014;  //!< 2^-44
00105 constexpr double TWO_N46 = 1.4210854715202e-014;     //!< 2^-46
00106 constexpr double TWO_N48 = 3.552713678800501e-015;  //!< 2^-46
00107
00108 constexpr double TWO_N50 = 8.881784197001252e-016;    //!< 2^-50

```





This graph shows which files directly or indirectly include this file:



## Classes

- class [DSM\\_nma\\_header](#)
- class [DSM\\_dsm\\_header](#)
- class [MACK\\_header](#)
- class [MACK\\_tag\\_info](#)
- class [MACK\\_tag\\_and\\_info](#)
- class [DSM\\_PKR\\_message](#)
- class [DSM\\_KROOT\\_message](#)
- class [MACK\\_message](#)
- class [OSNMA\\_NavData](#)
- class [OSNMA\\_data](#)

*This class handles OSNMA data See [https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo\\_OSNMA\\_User\\_ICD\\_for\\_Test\\_Phase\\_v1.0.pdf](https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo_OSNMA_User_ICD_for_Test_Phase_v1.0.pdf).*

- class [Tag](#)

## 11.785.1 Detailed Description

Class for Galileo OSNMA data storage.

### Author

Carles Fernandez-Prades, 2020-2023 cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2023 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [osnma\\_data.h](#).

---

## 11.786 osnma\_data.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002   * \file osnma_data.h
00003   * \brief Class for Galileo OSNMA data storage
  
```

```

00004  * \author Carles Fernandez-Prades, 2020-2023 cfernandez(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2023 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017
00018 #ifndef GNSS_SDR_OSNMA_DATA_H
00019 #define GNSS_SDR_OSNMA_DATA_H
00020
00021 #include "galileo_inav_message.h" // for OSNMA_msg
00022 #include <array>
00023 #include <cstdint>
00024 #include <memory>
00025 #include <string>
00026 #include <vector>
00027
00028 /** \addtogroup Core
00029  * \{ */
00030 /** \addtogroup System_Parameters
00031  * \{ */
00032
00033 class DSM_nma_header
00034 {
00035 public:
00036     DSM_nma_header() = default;
00037     uint8_t nmas{};
00038     uint8_t cid{};
00039     uint8_t cpks{};
00040     bool reserved{};
00041 };
00042
00043
00044 class DSM_dsm_header
00045 {
00046 public:
00047     DSM_dsm_header() = default;
00048     uint8_t dsm_id{};
00049     uint8_t dsm_block_id{};
00050 };
00051
00052
00053 class MACK_header
00054 {
00055 public:
00056     MACK_header() = default;
00057     uint64_t tag0{};
00058     uint16_t macseq{};
00059     uint8_t cop{};
00060 };
00061
00062
00063 class MACK_tag_info
00064 {
00065 public:
00066     MACK_tag_info() = default;
00067     uint8_t PRN_d{};
00068     uint8_t ADKD{};
00069     uint8_t cop{};
00070 };
00071
00072
00073 class MACK_tag_and_info
00074 {
00075 public:
00076     MACK_tag_and_info() = default;
00077     uint64_t tag; // C: 20-40 bits
00078     MACK_tag_info tag_info;
00079     uint32_t counter; // CTR
00080 };
00081
00082
00083 class DSM_PKR_message
00084 {
00085 public:
00086     DSM_PKR_message() = default;
00087
00088     std::array<uint8_t, 128> itn{}; // bitset<1024>
00089     std::vector<uint8_t> npk;
00090     std::vector<uint8_t> p_dp;

```

```

00091     uint8_t nb_dp{};
00092     uint8_t mid{};
00093     uint8_t npkt{};
00094     uint8_t npktid{};
00095 };
00096
00097
00098 class DSM_KROOT_message
00099 {
00100 public:
00101     DSM_KROOT_message() = default;
00102
00103     std::vector<uint8_t> kroot;
00104     std::vector<uint8_t> ds;
00105     std::vector<uint8_t> p_dk;
00106     uint64_t alpha{};
00107     uint16_t wn_k{};
00108     uint8_t nb_dk{};
00109     uint8_t pkid{};
00110     uint8_t cidkr{};
00111     uint8_t reserved1{};
00112     uint8_t hf{};
00113     uint8_t mf{};
00114     uint8_t ks{}; // key size, in bits
00115     uint8_t ts{};
00116     uint8_t mac1t{};
00117     uint8_t reserved{};
00118     uint8_t towh_k{};
00119     bool verified{false};
00120 };
00121
00122
00123 class MACK_message
00124 {
00125 public:
00126     MACK_message() = default;
00127     MACK_header header;
00128     std::vector<MACK_tag_and_info> tag_and_info;
00129     std::vector<uint8_t> key;
00130     uint32_t TOW; // TODO duplicated variable, also in OSNMA_NavData
00131     uint32_t WN;
00132     uint32_t PRNA;
00133 };
00134
00135
00136 class OSNMA_NavData
00137 {
00138 public:
00139     OSNMA_NavData() : nav_data_id(id_counter++) {}
00140     const uint32_t nav_data_id;
00141     std::string get_utc_data() const;
00142     std::string get_ephemeris_data() const;
00143     uint32_t get_verified_bits() const { return verified_bits; }
00144     uint32_t get_prn_d() const { return PRNd; }
00145     uint32_t get_IOD_nav() const { return IOD_nav; }
00146     uint32_t get_last_received_TOW() const { return d_last_received_TOW; }
00147     uint32_t get_tow_sf0() const { return d_TOW_sf0; }
00148     bool have_this_bits(std::string nav_data);
00149     bool get_verified_status() const { return verified; }
00150     bool add_nav_data(const std::string& nav_data);
00151     void set_tow_sf0(int value) { d_TOW_sf0 = value; }
00152     void set_ephemeris_data(std::string value) { d_ephemeris_iono = value; }
00153     void set_utc_data(std::string value) { d_utc = value; }
00154     void update_last_received_timestamp(uint32_t TOW);
00155     void set_prn_d(uint32_t value) { PRNd = value; }
00156     void set_last_received_TOW(uint32_t TOW) { d_last_received_TOW = TOW; }
00157     void set_update_verified_bits(uint32_t morebits) { verified_bits += morebits; }
00158     void set_verified_status(bool value) { verified = value; }
00159     void set_IOD_nav(uint32_t value) { IOD_nav = value; }
00160
00161 private:
00162     static uint32_t id_counter;
00163     std::string d_ephemeris_iono{" "};
00164     std::string d_utc{" "};
00165     uint32_t d_TOW_sf0{0};
00166     uint32_t d_last_received_TOW{0};
00167     uint32_t PRNd{0};
00168     uint32_t verified_bits{0};
00169     uint32_t IOD_nav{0};
00170     bool verified{false};
00171 };
00172
00173
00174 /*!
00175  * \brief This class handles OSNMA data
00176  * See
00177     https://www.gsc-europa.eu/sites/default/files/sites/all/files/Galileo\_OSNMA\_User\_ICD\_for\_Test\_Phase\_v1.0.pdf

```

```

00177  */
00178  class OSNMA_data
00179  {
00180  public:
00181      OSNMA_data() = default;
00182      DSM_nma_header d_nma_header;
00183      DSM_dsm_header d_dsm_header;
00184      DSM_PKR_message d_dsm_pkr_message;
00185      DSM_KROOT_message d_dsm_kroot_message;
00186      DSM_KROOT_message d_dsm_kroot_new_message;
00187      MACK_message d_mack_message;
00188      OSNMA_NavData d_nav_data;
00189  };
00190
00191
00192  class Tag
00193  {
00194  public:
00195      enum e_verification_status
00196      {
00197          SUCCESS,
00198          FAIL,
00199          UNVERIFIED
00200      };
00201      Tag(const MACK_tag_and_info& MTI, uint32_t TOW, uint32_t WN, uint32_t PRNa, uint8_t CTR) //
standard tag constructor, for tags within Tag&Info field
00202      : tag_id(id_counter++),
00203        TOW(TOW), // TODO missing for build_message WN for GST computation, CTR, NMAS,
OSNMA_NavData missing
00204        WN(WN),
00205        PRNa(PRNa),
00206        CTR(CTR),
00207        status(UNVERIFIED),
00208        received_tag(MTI.tag),
00209        computed_tag(0),
00210        PRN_d(MTI.tag_info.PRN_d),
00211        ADKD(MTI.tag_info.ADKD),
00212        cop(MTI.tag_info.cop),
00213        skipped(0)
00214      {
00215      }
00216      explicit Tag(const MACK_message& mack) // constructor for Tag0
00217      : tag_id(id_counter++),
00218        TOW(mack.TOW), // TODO missing for build_message WN for GST computation, CTR, NMAS,
OSNMA_NavData missing
00219        WN(mack.WN),
00220        PRNa(mack.PRNa),
00221        CTR(1),
00222        status(UNVERIFIED),
00223        received_tag(mack.header.tag0),
00224        computed_tag(0),
00225        PRN_d(mack.PRNa), // Tag0 are self-authenticating
00226        ADKD(0),
00227        cop(mack.header.cop),
00228        skipped(0)
00229      {
00230      }
00231      const uint32_t tag_id;
00232      static uint32_t id_counter;
00233      uint32_t TOW;
00234      uint32_t WN;
00235      uint32_t PRNa;
00236      uint8_t CTR;
00237      e_verification_status status;
00238      uint64_t received_tag;
00239      uint64_t computed_tag;
00240      uint8_t PRN_d;
00241      uint8_t ADKD;
00242      uint8_t cop;
00243      uint32_t skipped;
00244      std::string nav_data;
00245  };
00246
00247  /** \} */
00248  /** \} */
00249
00250  #endif // GNSS_SDR_OSNMA_DATA_H

```

## 11.787 osnma\_dsm\_reader.h File Reference

Class for reading OSNMA DSM messages.

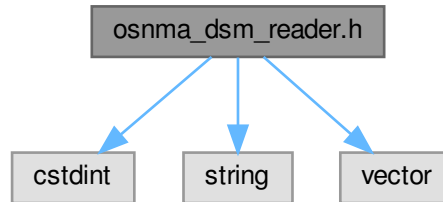
```

#include <cstdint>
#include <string>

```

```
#include <vector>
```

Include dependency graph for osnma\_dsm\_reader.h:



## Classes

- class [OSNMA\\_DSM\\_Reader](#)

### 11.787.1 Detailed Description

Class for reading OSNMA DSM messages.

#### Author

Carles Fernandez-Prades, 2023 cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2023 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [osnma\\_dsm\\_reader.h](#).

---

## 11.788 osnma\_dsm\_reader.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002  * \file osnma_dsm_reader.h
00003  * \brief Class for reading OSNMA DSM messages
00004  * \author Carles Fernandez-Prades, 2023 cfernandez(at)cttc.es
00005  *
00006  * -----
00007  *
00008  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009  * This file is part of GNSS-SDR.
00010  *
00011  * Copyright (C) 2010-2023 (see AUTHORS file for a list of contributors)
00012  * SPDX-License-Identifier: GPL-3.0-or-later
00013  *
00014  * -----
00015  */
00016
00017 #ifndef GNSS_SDR_OSNMA_DSM_READER_H
00018 #define GNSS_SDR_OSNMA_DSM_READER_H
00019
00020 #include <cstdint>
00021 #include <string>
00022 #include <vector>
00023
00024 /** \addtogroup Core
00025  * \{ */
00026 /** \addtogroup System_Parameters
00027  * \{ */
00028
00029 class OSNMA_DSM_Reader
00030 {
00031 public:
00032     OSNMA_DSM_Reader() = default;
00033     uint8_t get_nmas(uint8_t nma_header) const;
00034     uint8_t get_cid(uint8_t nma_header) const;
  
```

```

00035     uint8_t get_cpks(uint8_t nma_header) const;
00036     bool get_nma_header_reserved(uint8_t nma_header) const;
00037
00038     uint8_t get_dsm_id(uint8_t dsm_header) const;
00039     uint8_t get_dsm_block_id(uint8_t dsm_header) const;
00040
00041     uint8_t get_number_blocks_index(uint8_t dsm_msg_0) const;
00042     uint8_t get_pkid(const std::vector<uint8_t>& dsm_msg) const;
00043     uint8_t get_cidkr(const std::vector<uint8_t>& dsm_msg) const;
00044     uint8_t get_dsm_reserved1(const std::vector<uint8_t>& dsm_msg) const;
00045     uint8_t get_hf(const std::vector<uint8_t>& dsm_msg) const;
00046     uint8_t get_mf(const std::vector<uint8_t>& dsm_msg) const;
00047     uint8_t get_ks(const std::vector<uint8_t>& dsm_msg) const;
00048     uint8_t get_ts(const std::vector<uint8_t>& dsm_msg) const;
00049     uint8_t get_mac1t(const std::vector<uint8_t>& dsm_msg) const;
00050     uint8_t get_dsm_reserved(const std::vector<uint8_t>& dsm_msg) const;
00051     uint16_t get_wn_k(const std::vector<uint8_t>& dsm_msg) const;
00052     uint8_t get_towh_k(const std::vector<uint8_t>& dsm_msg) const;
00053     uint64_t get_alpha(const std::vector<uint8_t>& dsm_msg) const;
00054     uint16_t get_l_dk_bits(uint8_t nb_dk) const;
00055     uint16_t get_lk_bits(uint8_t ks) const;
00056     std::vector<uint8_t> get_kroot(const std::vector<uint8_t>& dsm_msg, uint16_t bytes_lk) const;
00057     std::string get_hash_function(uint8_t hf) const;
00058     std::string get_nmas_status(uint8_t nmas) const;
00059     std::string get_cpks_status(uint8_t cpks) const;
00060
00061     uint8_t get_mid(const std::vector<uint8_t>& dsm_msg) const;
00062     uint8_t get_npkt(const std::vector<uint8_t>& dsm_msg) const;
00063     uint8_t get_npktid(const std::vector<uint8_t>& dsm_msg) const;
00064
00065 private:
00066     static constexpr std::uint8_t mask_nmas{0xC0};
00067     static constexpr std::uint8_t mask_cid{0x30};
00068     static constexpr std::uint8_t mask_cpks{0x0E};
00069     static constexpr std::uint8_t mask_nma_header_reserved{0x01};
00070     static constexpr std::uint8_t mask_dsm_id{0xF0};
00071     static constexpr std::uint8_t mask_dsm_block_id{0x0F};
00072     static constexpr std::uint8_t mask_dsm_number_blocks{0xF0};
00073     static constexpr std::uint8_t mask_dsm_pkid{0x0F};
00074     static constexpr std::uint8_t mask_dsm_cidkr{0xC0};
00075     static constexpr std::uint8_t mask_dsm_reserved1{0x30};
00076     static constexpr std::uint8_t mask_dsm_hf{0x0C};
00077     static constexpr std::uint8_t mask_dsm_mf{0x03};
00078     static constexpr std::uint8_t mask_dsm_ks{0xF0};
00079     static constexpr std::uint8_t mask_dsm_ts{0x0F};
00080     static constexpr std::uint8_t mask_dsm_reserved{0xF0};
00081     static constexpr std::uint8_t mask_dsm_wk_k_msbyte{0x0F};
00082     static constexpr std::uint8_t mask_dsm_mid{0x0F};
00083     static constexpr std::uint8_t mask_dsm_npkt{0xF0};
00084     static constexpr std::uint8_t mask_dsm_npktid{0x0F};
00085 };
00086
00087 /** \} */
00088 /** \} */
00089 #endif // GNSS_SDR_OSNMA_DSM_READER_H

```

## 11.789 reed\_solomon.h File Reference

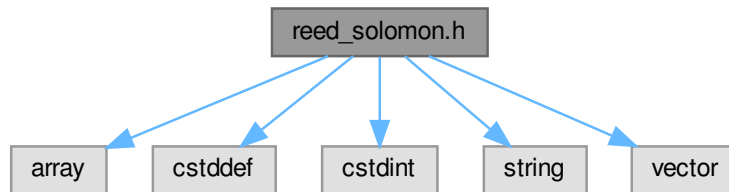
Class implementing a Reed-Solomon encoder/decoder for blocks of 255 symbols and symbols of 8 bits.

```

#include <array>
#include <cstdint>
#include <cstdint>
#include <string>
#include <vector>

```

Include dependency graph for reed\_solomon.h:



## Classes

- class [ReedSolomon](#)

*Class implementing a Reed-Solomon encoder and decoder  $RS(255, K, d)$  where  $k=255-nroots$  is the information vector length and  $d=nroots+1$  is the minimum Hamming distance, with symbols of 8 bits. It allows shortened RS codes.*

## 11.789.1 Detailed Description

Class implementing a Reed-Solomon encoder/decoder for blocks of 255 symbols and symbols of 8 bits.

### Author

Carles Fernandez, 2021. cfernandez(at)cttc.es

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.  
 Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [reed\\_solomon.h](#).

---

## 11.790 reed\_solomon.h

[Go to the documentation of this file.](#)

```

00001 /*!
00002  * \file reed_solomon.h
00003  * \brief Class implementing a Reed-Solomon encoder/decoder for blocks of 255
00004  * symbols and symbols of 8 bits.
00005  * \author Carles Fernandez, 2021. cfernandez(at)cttc.es
00006  *
00007  * -----
00008  *
00009  * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00010  * This file is part of GNSS-SDR.
00011  *
00012  * Copyright (C) 2010-2021 (see AUTHORS file for a list of contributors)
00013  * SPDX-License-Identifier: GPL-3.0-or-later
00014  *
00015  * -----
00016  */
00017
00018
00019 #ifndef GNSS_SDR_REED_SOLOMON_H
00020 #define GNSS_SDR_REED_SOLOMON_H
00021
00022 #include <array>
00023 #include <cstdint>
00024 #include <cstdint>
00025 #include <string>
00026 #include <vector>
00027
00028
00029 /** \addtogroup Core
00030  * \{ */
00031 /** \addtogroup System_Parameters
00032  * \{ */
00033
  
```

```

00034  /*!
00035  * \brief
00036  * Class implementing a Reed-Solomon encoder and decoder RS(255,K,d) where
00037  * k=255-nroots is the information vector length and d=nroots+1 is the minimum
00038  * Hamming distance, with symbols of 8 bits. It allows shortened RS codes.
00039  */
00040  class ReedSolomon
00041  {
00042  public:
00043      /*!
00044       * \brief Default constructor.
00045       * Constructs a Reed Solomon object. The encode_with_generator_poly
00046       * and encode_with_generator_matrix methods are available for testing
00047       * purposes.
00048       *
00049       * gnss_signal: empty or "E6B" sets the Galileo E6B RS parameters.
00050       * "E1B" sets the Galileo E1B (INAV) RS parameters.
00051       */
00052      explicit ReedSolomon(const std::string& gnss_signal = std::string());
00053
00054      /*!
00055       * \brief Custom constructor for RS(255, 255-nroots, nroots+1). Parameters:
00056       *
00057       * nroots - the number of roots in the RS code generator polynomial,
00058       * which is the same as the number of parity symbols in a block.
00059       *
00060       * minpoly - primitive polynomial.
00061       *
00062       * prim - the primitive root of the generator polynomial.
00063       *
00064       * fcr - first consecutive root of the Reed-Solomon generator polynomial.
00065       *
00066       * pad - the number of pad symbols in a block. If not defined, it defaults
00067       * to 0.
00068       *
00069       * shortening - value of the shortening parameter. Defaults to 0.
00070       * If different to 0, it defines a shortened RS encoder/decoder.
00071       *
00072       * genpoly_coeff - a vector of (nroots+1) elements containing the generator
00073       * polynomial coefficients. Only used for encoding. Defaults to empty.
00074       * If defined, the encode_with_generator_poly method can be used.
00075       *
00076       * gen_matrix - a (255-shortening)x(255-nroots-shortening) matrix containing
00077       * the elements of the generator matrix. Only used for encoding. Defaults
00078       * to empty. If defined, the encode_with_generator_matrix method can be
00079       * used.
00080       */
00081      ReedSolomon(int nroots,
00082                  int minpoly,
00083                  int prim,
00084                  int fcr,
00085                  int pad = 0,
00086                  int shortening = 0,
00087                  const std::vector<uint8_t>& genpoly_coeff = std::vector<uint8_t>{},
00088                  const std::vector<std::vector<uint8_t>& gen_matrix = std::vector<std::vector<uint8_t>>{});
00089
00090      /*!
00091       * \brief Decode an encoded block.
00092       *
00093       * The decoded symbols are at the first 255-nroots-shortening elements
00094       * of the data_to_decode vector.
00095       *
00096       * The second parameter is optional, and contains a vector of erasure
00097       * positions to be passed to the decoding algorithm. Defaults to empty.
00098       *
00099       * Returns the number of corrected errors, or -1 if decoding failed.
00100       */
00101      int decode(std::vector<uint8_t>& data_to_decode,
00102                const std::vector<int>& erasure_positions = std::vector<int>{}) const;
00103
00104      /*!
00105       * \brief Encode data with the generator matrix (for testing purposes)
00106       *
00107       * Returns the encoded vector. It is set to all zeros if the generator
00108       * matrix is not defined.
00109       */
00110      std::vector<uint8_t> encode_with_generator_matrix(const std::vector<uint8_t>& data_to_encode)
00111      const;
00112
00113      /*!
00114       * \brief Encode data with the generator polynomial (for testing purposes)
00115       *
00116       * Returns the encoded vector. It is set to all zeros if the generator
00117       * polynomial is not defined.
00118       */
00119      std::vector<uint8_t> encode_with_generator_poly(const std::vector<uint8_t>& data_to_encode) const;

```



```

00120 private:
00121     static const int d_symbols_per_block = 255; // the total number of symbols in a RS block.
00122     static const int d_symsize = 8;             // symbol size, in bits.
00123
00124     int mod255(int x) const;
00125     int rs_min(int a, int b) const;
00126     int decode_rs_8(uint8_t* data, const int* eras_pos, int no_eras) const;
00127
00128     uint8_t galois_mul(uint8_t a, uint8_t b) const;
00129     uint8_t galois_add(uint8_t a, uint8_t b) const;
00130     uint8_t galois_mul_table(uint8_t a, uint8_t b) const;
00131
00132     void encode_rs_8(const uint8_t* data, uint8_t* parity) const;
00133     void init_log_tables(); // initialize d_log_table and d_antilog
00134     void init_alpha_tables(); // initialize d_alpha_to, d_index_of
00135
00136     std::array<uint8_t, 256> d_alpha_to{}; // used for decoding
00137     std::array<uint8_t, 256> d_index_of{}; // used for decoding
00138     std::array<uint8_t, 256> d_log_table{}; // used for encoding
00139     std::array<uint8_t, 256> d_antilog{}; // used for encoding
00140
00141     std::vector<std::vector<uint8_t>> d_genmatrix; // used for encoding
00142     std::vector<uint8_t> d_genpoly_coeff; // used for encoding
00143     std::vector<uint8_t> d_genpoly_index; // used for encoding
00144
00145     size_t d_data_in_block{}; // number of information symbols in a block
00146     size_t d_rows_G{}; // number of rows of the generator matrix
00147     size_t d_columns_G{}; // number of rows of the generator matrix
00148     size_t d_info_symbols_shortened{}; // number of info symbols in the shortened code
00149     size_t d_data_symbols_shortened{}; // number of data symbols in the shortened code
00150
00151     int d_nroots{}; // number of parity symbols in a block
00152     int d_prim{}; // The primitive root of the generator poly
00153     int d_pad{}; // the number of pad symbols in a block
00154     int d_iprim{}; // prim-th root of 1, index form
00155     int d_fcr{}; // first consecutive root
00156     int d_shortening{}; // shortening parameter
00157
00158     uint8_t d_min_poly{}; // primitive polynomial
00159     uint8_t d_a0{}; // auxiliary variable
00160 };
00161
00162 /** \} */
00163 /** \} */
00164 #endif // GNSS_SDR_REED_SOLOMON_H

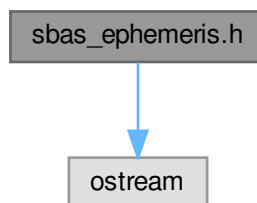
```

## 11.791 sbas\_ephemeris.h File Reference

Interface of a SBAS REFERENCE LOCATION storage.

```
#include <ostream>
```

Include dependency graph for sbas\_ephemeris.h:



### Classes

- class [Sbas\\_Ephemeris](#)

*This class stores SBAS SV ephemeris data.*

## 11.791.1 Detailed Description

Interface of a SBAS REFERENCE LOCATION storage.

Author

Daniel Fehr, 2013. daniel.co(at)bluewin.ch

---

GNSS-SDR is a Global Navigation Satellite System software-defined receiver. This file is part of GNSS-SDR.

Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors) SPDX-License-Identifier: GPL-3.0-or-later  
 Definition in file [sbas\\_ephemeris.h](#).

---

## 11.792 sbas\_ephemeris.h

[Go to the documentation of this file.](#)

```

00001  /*!
00002   * \file sbas_ephemeris.h
00003   * \brief Interface of a SBAS REFERENCE LOCATION storage
00004   * \author Daniel Fehr, 2013. daniel.co(at)bluewin.ch
00005   *
00006   * -----
00007   *
00008   * GNSS-SDR is a Global Navigation Satellite System software-defined receiver.
00009   * This file is part of GNSS-SDR.
00010   *
00011   * Copyright (C) 2010-2020 (see AUTHORS file for a list of contributors)
00012   * SPDX-License-Identifier: GPL-3.0-or-later
00013   *
00014   * -----
00015   */
00016
00017
00018 #ifndef GNSS_SDR_SBAS_EPHEMERIS_H
00019 #define GNSS_SDR_SBAS_EPHEMERIS_H
00020
00021 #include <ostream>
00022
00023 /** \addtogroup Core
00024  * \{ */
00025 /** \addtogroup System_Parameters
00026  * \{ */
00027
00028
00029 /*!
00030  * \brief This class stores SBAS SV ephemeris data
00031  *
00032  */
00033 class Sbas_Ephemeris
00034 {
00035 public:
00036     Sbas_Ephemeris() = default;
00037
00038     int i_prn{};          //!< PRN number
00039     int i_t0{};          //!< Reference epoch time (GPST)
00040     double d_tof{};      //!< Time of message frame (GPST)
00041     int i_sv_ura{};      //!< SV accuracy (URA index), not standardized
00042     bool b_sv_do_not_use{}; //!< Health status (false:do not use / true:usable)
00043     double d_pos[3]{};   //!< Satellite position (m) (ECEF)
00044     double d_vel[3]{};   //!< Satellite velocity (m/s) (ECEF)
00045     double d_acc[3]{};   //!< Satellite acceleration (m/s^2) (ECEF)
00046     double d_af0{};      //!< Satellite clock-offset (s)
00047     double d_af1{};      //!< Satellite drift (s/s)
00048
00049     void print(std::ostream &out);
00050 };
00051
00052
00053 /** \} */
00054 /** \} */
00055 #endif // GNSS_SDR_SBAS_EPHEMERIS_H

```

# Index

- ~An\_Packet\_Printer
  - An\_Packet\_Printer, [358](#)
- ~ArraySignalConditioner
  - ArraySignalConditioner, [362](#)
- ~Channel
  - Channel, [428](#)
- ~ControlThread
  - ControlThread, [466](#)
- ~Exponential\_Smoothing
  - Exponential\_Smoothing, [527](#)
- ~FIRFilter
  - FIRFilter, [560](#)
- ~FPGA\_Acquisition
  - FPGA\_Acquisition, [577](#)
- ~FPGA\_DMA
  - FPGA\_DMA, [581](#)
- ~FPGA\_Multicorrelator\_8sc
  - FPGA\_Multicorrelator\_8sc, [585](#)
- ~FPGA\_Switch
  - FPGA\_Switch, [590](#)
- ~FPGA\_buffer\_monitor
  - FPGA\_buffer\_monitor, [580](#)
- ~FPGA\_dynamic\_bit\_selection
  - FPGA\_dynamic\_bit\_selection, [583](#)
- ~FPGA\_spidev
  - FPGA\_spidev, [588](#)
- ~GNSSFlowgraph
  - GNSSFlowgraph, [913](#)
- ~GalileoE1DIIPIIVemlTrackingFPGA
  - GalileoE1DIIPIIVemlTrackingFPGA, [672](#)
- ~GalileoE1PcpsAmbiguousAcquisitionFPGA
  - GalileoE1PcpsAmbiguousAcquisitionFPGA, [689](#)
- ~GalileoE5aDIIPIITrackingFPGA
  - GalileoE5aDIIPIITrackingFPGA, [720](#)
- ~GalileoE5aPcpsAcquisitionFPGA
  - GalileoE5aPcpsAcquisitionFPGA, [738](#)
- ~GalileoE5bPcpsAcquisition
  - GalileoE5bPcpsAcquisition, [753](#)
- ~GalileoE5bPcpsAcquisitionFPGA
  - GalileoE5bPcpsAcquisitionFPGA, [760](#)
- ~Gnss\_Crypto
  - Gnss\_Crypto, [868](#)
- ~Gnss\_Satellite
  - Gnss\_Satellite, [880](#)
- ~Gnss\_Synchro
  - Gnss\_Synchro, [900](#)
- ~GpsL1CaDIIPIITrackingFPGA
  - GpsL1CaDIIPIITrackingFPGA, [972](#)
- ~GpsL1CaPcpsAcquisitionFPGA
  - GpsL1CaPcpsAcquisitionFPGA, [1000](#)
- ~GpsL5DIIPIITrackingFPGA
  - GpsL5DIIPIITrackingFPGA, [1063](#)
- ~GpsL5iPcpsAcquisitionFPGA
  - GpsL5iPcpsAcquisitionFPGA, [1074](#)
- ~Nav\_Message\_Packet
  - Nav\_Message\_Packet, [1187](#)
- ~Nmea\_Printer
  - Nmea\_Printer, [1197](#)
- ~Rinex\_Printer
  - Rinex\_Printer, [1334](#)
- ~Rtcm\_Printer
  - Rtcm\_Printer, [1351](#)
- ~Serdes\_Galileo\_Eph
  - Serdes\_Galileo\_Eph, [1412](#)
- ~Serdes\_Gnss\_Synchro
  - Serdes\_Gnss\_Synchro, [1414](#)
- ~Serdes\_Gps\_Eph
  - Serdes\_Gps\_Eph, [1416](#)
- ~Serdes\_Monitor\_Pvt
  - Serdes\_Monitor\_Pvt, [1418](#)
- ~Serdes\_Nav\_Message
  - Serdes\_Nav\_Message, [1419](#)
- ~SignalConditioner
  - SignalConditioner, [1427](#)
- ~SignalSourceInterface
  - GNSS block interfaces, [207](#)
- ~beidou\_b1i\_telemetry\_decoder\_gs
  - beidou\_b1i\_telemetry\_decoder\_gs, [371](#)
- ~beidou\_b3i\_telemetry\_decoder\_gs
  - beidou\_b3i\_telemetry\_decoder\_gs, [373](#)
- ~channel\_msg\_receiver\_cc
  - channel\_msg\_receiver\_cc, [433](#)
- ~channel\_status\_msg\_receiver
  - channel\_status\_msg\_receiver, [434](#)
- ~dll\_pll\_veml\_tracking\_fpga
  - dll\_pll\_veml\_tracking\_fpga, [508](#)
- ~galileo\_e5a\_noncoherentIQ\_acquisition\_caf\_cc
  - galileo\_e5a\_noncoherentIQ\_acquisition\_caf\_cc, [611](#)
- ~galileo\_e6\_has\_msg\_receiver
  - galileo\_e6\_has\_msg\_receiver, [616](#)
- ~galileo\_pcps\_8ms\_acquisition\_cc
  - galileo\_pcps\_8ms\_acquisition\_cc, [648](#)
- ~galileo\_tow\_map
  - galileo\_tow\_map, [657](#)
- ~glonass\_l1\_ca\_telemetry\_decoder\_gs
  - glonass\_l1\_ca\_telemetry\_decoder\_gs, [818](#)
- ~glonass\_l2\_ca\_telemetry\_decoder\_gs

- glonass\_l2\_ca\_telemetry\_decoder\_gs, 825
- ~gnss\_synchro\_monitor
  - gnss\_synchro\_monitor, 906
- ~nav\_message\_monitor
  - nav\_message\_monitor, 1185
- ~osnma\_msg\_receiver
  - osnma\_msg\_receiver, 1233
- ~pcps\_acquisition\_fine\_doppler\_cc
  - pcps\_acquisition\_fine\_doppler\_cc, 1249
- ~pcps\_acquisition\_fpga
  - pcps\_acquisition\_fpga, 1254
- ~pcps\_assisted\_acquisition\_cc
  - pcps\_assisted\_acquisition\_cc, 1259
- ~pcps\_cccwsr\_acquisition\_cc
  - pcps\_cccwsr\_acquisition\_cc, 1264
- ~pcps\_openc1\_acquisition\_cc
  - pcps\_openc1\_acquisition\_cc, 1269
- ~pcps\_quicksync\_acquisition\_cc
  - pcps\_quicksync\_acquisition\_cc, 1275
- ~pcps\_tong\_acquisition\_cc
  - pcps\_tong\_acquisition\_cc, 1281
- ~rtklib\_pvt\_gs
  - rtklib\_pvt\_gs, 1367
- A
  - alm\_t, 354
  - eph\_t, 520
- A0
  - Galileo\_Utc\_Model, 658
  - Gps\_Utc\_Model, 963
- A0\_6\_LSB
  - core\_system\_parameters, 247
- A0\_GAL
  - Beidou\_Dnav\_Utc\_Model, 392
- A0\_GLO
  - Beidou\_Dnav\_Utc\_Model, 392
- A0\_GPS
  - Beidou\_Dnav\_Utc\_Model, 392
- A0\_UTC
  - Beidou\_Dnav\_Utc\_Model, 392
- A1
  - Galileo\_Utc\_Model, 658
  - Gps\_Utc\_Model, 963
- A1\_6\_LSB
  - core\_system\_parameters, 247
- A1\_GAL
  - Beidou\_Dnav\_Utc\_Model, 392
- A1\_GLO
  - Beidou\_Dnav\_Utc\_Model, 392
- A1\_GPS
  - Beidou\_Dnav\_Utc\_Model, 393
- A1\_UTC
  - Beidou\_Dnav\_Utc\_Model, 393
- A2
  - Gps\_Utc\_Model, 963
- A\_0\_LSB
  - core\_system\_parameters, 247
- A\_0G
  - Galileo\_Utc\_Model, 659
- A\_0G\_10\_LSB
  - core\_system\_parameters, 247
- A\_1\_LSB
  - core\_system\_parameters, 247
- A\_1\_LSB\_GAL
  - core\_system\_parameters, 247
- A\_1G
  - Galileo\_Utc\_Model, 659
- A\_1G\_10\_LSB
  - core\_system\_parameters, 247
- A\_F0\_LSB
  - core\_system\_parameters, 247
- A\_F1\_LSB
  - core\_system\_parameters, 247
- A\_F2\_LSB
  - core\_system\_parameters, 247
- ABS
  - telemetry\_decoder\_libswiftcnv, 190
- ABSL\_DECLARE\_FLAG
  - gnss\_sdr\_flags, 114–117
- acc
  - geph\_t, 789
  - lexeph\_t, 1149
  - seph\_t, 1410
- Acq\_Conf, 324
  - bit\_transition\_flag, 325
  - blocking, 325
  - blocking\_on\_standby, 325
  - chips\_per\_second, 325
  - doppler\_max, 326
  - doppler\_min, 326
  - doppler\_step, 326
  - doppler\_step2, 326
  - dump, 326
  - dump\_channel, 326
  - dump\_filename, 326
  - enable\_monitor\_output, 326
  - fs\_in, 326
  - it\_size, 326
  - item\_type, 326
  - make\_2\_steps, 326
  - max\_dwells, 327
  - ms\_per\_code, 327
  - num\_doppler\_bins\_step2, 327
  - pfa, 327
  - pfa2, 327
  - resampled\_fs, 327
  - resampler\_latency\_samples, 327
  - resampler\_ratio, 327
  - sampled\_ms, 327
  - samples\_per\_chip, 327
  - samples\_per\_code, 327
  - samples\_per\_ms, 327
  - use\_automatic\_resampler, 328
  - use\_CFAR\_algorithm\_flag, 328
- acq\_conf.h, 1663, 1664
- Acq\_Conf\_Fpga, 329
  - all\_fft\_codes, 330

- code\_length, 330
- code\_length\_chips, 330
- code\_rate\_cps, 330
- device\_name, 330
- doppler\_max, 330
- doppler\_step, 330
- doppler\_step2, 331
- downsampling\_factor, 331
- downsampling\_filter\_delay, 331
- downsampling\_filter\_num, 331
- enable\_zero\_padding, 331
- excludelimit, 331
- fft\_size, 331
- fs\_in, 331
- make\_2\_steps, 331
- max\_num\_acqs, 331
- num\_doppler\_bins\_step2, 331
- repeat\_satellite, 331
- resampled\_fs, 332
- total\_block\_exp, 332
- acq\_conf\_fpga.h, 1665, 1666
- Acq\_delay\_samples
  - Gnss\_Synchro, 902
- Acq\_doppler\_hz
  - Gnss\_Synchro, 902
- Acq\_doppler\_step
  - Gnss\_Synchro, 902
- Acq\_samplestamp\_samples
  - Gnss\_Synchro, 902
- Acquisition, 75
- acquisition
  - Channel, 429
- acquisition\_adapters, 76
  - pcps\_acquisition\_fine\_doppler\_cc\_sptr, 77
- acquisition\_gr\_blocks, 77
  - galileo\_e5a\_noncoherentIQ\_acquisition\_caf\_cc\_sptr, 79
  - galileo\_pcps\_8ms\_acquisition\_cc\_sptr, 79
  - pcps\_acquisition\_fine\_doppler\_cc\_sptr, 79
  - pcps\_acquisition\_fpga\_sptr, 79
  - pcps\_acquisition\_sptr, 79
  - pcps\_assisted\_acquisition\_cc\_sptr, 80
  - pcps\_cccwsr\_acquisition\_cc\_sptr, 80
  - pcps\_openc1\_acquisition\_cc\_sptr, 80
  - pcps\_quicksync\_acquisition\_cc\_sptr, 80
  - pcps\_tong\_acquisition\_cc\_sptr, 80
- acquisition\_interface.h, 2312, 2313
- acquisition\_libs, 80
- acquisition\_manager
  - GNSSFlowgraph, 913
- AcquisitionInterface, 332
  - init, 334
  - mag, 334
  - reset, 334
  - set\_channel, 335
  - set\_channel\_fsm, 335
  - set\_doppler\_center, 335
  - set\_doppler\_max, 335
  - set\_doppler\_step, 335
  - set\_gnss\_synchro, 336
  - set\_local\_code, 336
  - set\_resampler\_latency, 336
  - set\_state, 336
  - set\_threshold, 336
  - stop\_acquisition, 337
- ad9361\_manager.h, 2069, 2070
- ad936x\_custom\_signal\_source.h, 1983, 1984
- ad936x\_iio\_custom, 337
  - n\_channels, 338
- ad936x\_iio\_custom.h, 2072, 2073
- ad936x\_iio\_samples, 338
  - buffer, 339
  - n\_bytes, 339
  - n\_channels, 339
  - n\_interleaved\_iq\_samples, 339
  - step\_bytes, 339
- ad936x\_iio\_samples.h, 2075, 2077
  - IIO\_DEFAULTAD936XAPIFIFOSIZE\_SAMPLES, 2076
  - IIO\_INPUTRAMFIFOSIZE, 2076
- ad936x\_iio\_source, 339
  - start, 341
  - stop, 341
- ad936x\_iio\_source.h, 2041, 2042
- ad936x\_iio\_source\_sptr
  - signal\_source\_gr\_blocks, 181
- Ad936xCustomSignalSource, 341
  - connect, 344
  - disconnect, 344
  - get\_left\_block, 344
  - get\_right\_block, 344
  - item\_size, 345
- addr
  - ftp\_t, 595
  - tcp\_t, 1477
- ADKD
  - MACK\_tag\_info, 1162
  - Tag, 1473
- Adot
  - eph\_t, 520
  - Gps\_CNAV\_Ephemeris, 928
- adrv9361\_z7035\_signal\_source\_fpga.h, 1986, 1987
- Adrv9361z7035SignalSourceFPGA, 346
  - connect, 348
  - disconnect, 348
  - get\_left\_block, 348
  - get\_right\_block, 348
  - item\_size, 348
- advance\_word
  - IONGSMSChunkUnpackingCtx< WT >, 1119
- af0
  - Gnss\_Almanac, 863
  - Gnss\_Ephemeris, 874
  - lexeph\_t, 1149
  - seph\_t, 1410
- af0\_10

- Galileo\_Almanac\_Helper, 601
- AF0\_10\_LSB
  - core\_system\_parameters, 247
- AF0\_4\_LSB
  - core\_system\_parameters, 248
- af0\_8
  - Galileo\_Almanac\_Helper, 601
- AF0\_8\_LSB
  - core\_system\_parameters, 248
- af0\_9
  - Galileo\_Almanac\_Helper, 601
- AF0\_9\_LSB
  - core\_system\_parameters, 248
- af0red
  - Galileo\_Reduced\_CED, 652
- af1
  - Gnss\_Almanac, 863
  - Gnss\_Ephemeris, 874
  - lexeph\_t, 1149
  - seph\_t, 1410
- af1\_10
  - Galileo\_Almanac\_Helper, 601
- AF1\_10\_LSB
  - core\_system\_parameters, 248
- AF1\_4\_LSB
  - core\_system\_parameters, 248
- af1\_8
  - Galileo\_Almanac\_Helper, 602
- AF1\_8\_LSB
  - core\_system\_parameters, 248
- af1\_9
  - Galileo\_Almanac\_Helper, 602
- AF1\_9\_LSB
  - core\_system\_parameters, 248
- af1red
  - Galileo\_Reduced\_CED, 652
- af2
  - Gnss\_Ephemeris, 874
- AF2\_4\_LSB
  - core\_system\_parameters, 248
- age
  - geph\_t, 789
  - sol\_t, 1438
- Agnss\_Ref\_Location, 349
  - Agnss\_Ref\_Location, 349
  - lat, 350
  - lon, 350
  - serialize, 349
  - uncertainty, 350
  - valid, 350
- agnss\_ref\_location.h, 2399, 2400
- Agnss\_Ref\_Time, 350
  - Agnss\_Ref\_Time, 351
  - microseconds, 351
  - seconds, 351
  - serialize, 351
  - tow, 351
  - valid, 351
- week, 351
- agnss\_ref\_time.h, 2401, 2402
- ai
  - sbsfcrr\_t, 1396
- ai0
  - Galileo\_Iono, 643
- AI0\_5\_LSB
  - core\_system\_parameters, 248
- ai1
  - Galileo\_Iono, 643
- AI1\_5\_LSB
  - core\_system\_parameters, 248
- ai2
  - Galileo\_Iono, 643
- AI2\_5\_LSB
  - core\_system\_parameters, 248
- alert
  - cnav\_msg\_t, 449
  - lexmsg\_t, 1151
- alert\_flag
  - Beidou\_Dnav\_Ephemeris, 382
  - Gps\_CNAV\_Ephemeris, 928
  - Gps\_Ephemeris, 942
- Algorithms Common Library, 90
- algorithms\_libs, 90
  - at, 97
  - back, 97
  - beidou\_b1i\_code\_gen\_complex, 97
  - beidou\_b1i\_code\_gen\_complex\_sampled, 97
  - beidou\_b1i\_code\_gen\_float, 97
  - beidou\_b1i\_code\_gen\_int, 97
  - beidou\_b3i\_code\_gen\_complex, 98
  - beidou\_b3i\_code\_gen\_complex\_sampled, 98
  - beidou\_b3i\_code\_gen\_float, 98
  - beidou\_b3i\_code\_gen\_int, 98
  - byte\_x2\_to\_complex\_byte\_sptr, 95
  - cart2geo, 98
  - cart2utm, 99
  - clear, 99
  - clksin, 99
  - clsin, 99
  - complex\_byte\_to\_float\_x2\_sptr, 95
  - complex\_exp\_gen, 100
  - complex\_exp\_gen\_conj, 100
  - complex\_float\_to\_complex\_byte\_sptr, 95
  - conjugate\_cc\_sptr, 96
  - conjugate\_ic\_sptr, 96
  - conjugate\_sc\_sptr, 96
  - cshort\_to\_float\_x2\_sptr, 96
  - findUtmZone, 100
  - front, 100
  - galileo\_e1\_code\_gen\_complex\_sampled, 100, 101
  - galileo\_e1\_code\_gen\_float\_sampled, 101
  - galileo\_e1\_code\_gen\_sinboc11\_float, 101
  - galileo\_e5\_a\_code\_gen\_complex\_primary, 102
  - galileo\_e5\_a\_code\_gen\_complex\_sampled, 102
  - galileo\_e5\_b\_code\_gen\_complex\_primary, 102
  - galileo\_e5\_b\_code\_gen\_complex\_sampled, 102

- galileo\_e6\_b\_code\_gen\_complex\_primary, [102](#)
- galileo\_e6\_b\_code\_gen\_complex\_sampled, [103](#)
- galileo\_e6\_b\_code\_gen\_float\_primary, [103](#)
- galileo\_e6\_c\_code\_gen\_complex\_primary, [103](#)
- galileo\_e6\_c\_code\_gen\_complex\_sampled, [103](#)
- galileo\_e6\_c\_code\_gen\_float\_primary, [103](#)
- galileo\_e6\_c\_secondary\_code, [103](#)
- galileo\_e6\_c\_secondary\_code\_gen\_complex, [104](#)
- galileo\_e6\_c\_secondary\_code\_gen\_float, [104](#)
- Geo\_to\_ECEF, [104](#)
- get, [104](#)
- glonass\_l1\_ca\_code\_gen\_complex, [104](#)
- glonass\_l1\_ca\_code\_gen\_complex\_sampled, [105](#)
- glonass\_l2\_ca\_code\_gen\_complex, [105](#)
- glonass\_l2\_ca\_code\_gen\_complex\_sampled, [105](#)
- Gnss\_circular\_deque, [105](#)
- gps\_l1\_ca\_code\_gen\_complex, [106](#)
- gps\_l1\_ca\_code\_gen\_complex\_sampled, [106](#)
- gps\_l1\_ca\_code\_gen\_float, [106](#)
- gps\_l1\_ca\_code\_gen\_int, [106](#)
- gps\_l2c\_m\_code\_gen\_complex, [107](#)
- gps\_l2c\_m\_code\_gen\_complex\_sampled, [107](#)
- gps\_l2c\_m\_code\_gen\_float, [107](#)
- gps\_l5i\_code\_gen\_complex, [107](#)
- gps\_l5i\_code\_gen\_complex\_sampled, [107](#)
- gps\_l5i\_code\_gen\_float, [107](#)
- gps\_l5q\_code\_gen\_complex, [108](#)
- gps\_l5q\_code\_gen\_complex\_sampled, [108](#)
- gps\_l5q\_code\_gen\_float, [108](#)
- Gravity\_ECEF, [108](#)
- great\_circle\_distance, [108](#)
- hex\_to\_binary\_converter, [108](#)
- hex\_to\_binary\_string, [109](#)
- item\_type\_converter\_t, [96](#)
- item\_type\_is\_complex, [109](#)
- item\_type\_size, [109](#)
- item\_type\_valid, [109](#)
- make\_vector\_converter, [109](#)
- pop\_front, [110](#)
- push\_back, [110](#)
- pv\_Geo\_to\_ECEF, [110](#)
- resampler, [111](#)
- reset, [111](#)
- short\_x2\_to\_cshort\_sptr, [96](#)
- size, [112](#)
- Skew\_symmetric, [112](#)
- togeod, [112](#)
- topocent, [112](#)
- algorithms\_libs\_rtklib, [118](#)
- ARMODE\_CONT, [139](#)
- ARMODE\_FIXHOLD, [139](#)
- ARMODE\_INST, [139](#)
- ARMODE\_OFF, [140](#)
- ARMODE\_PPPAR, [140](#)
- ARMODE\_PPPAR\_ILS, [140](#)
- ARMODE\_TCAR, [140](#)
- ARMODE\_WLNL, [140](#)
- CHISQR, [140](#)
- closesocket, [130](#)
- COMMENTH, [130](#)
- dev\_t, [130](#)
- DTTOL, [141](#)
- EFACT\_BDS, [141](#)
- EFACT\_GAL, [141](#)
- EFACT\_GLO, [141](#)
- EFACT\_GPS, [141](#)
- EFACT\_IRN, [142](#)
- EFACT\_QZS, [142](#)
- EFACT\_SBS, [142](#)
- ENABDS, [130](#)
- ENAGLO, [130](#)
- eph\_to\_rtklib, [139](#)
- EPHOPT\_BRDC, [142](#)
- EPHOPT\_LEX, [142](#)
- EPHOPT\_PREC, [142](#)
- EPHOPT\_SBAS, [143](#)
- EPHOPT\_SSRAPC, [143](#)
- EPHOPT\_SSRCOM, [143](#)
- ERR\_BRDCI, [143](#)
- ERR\_CBIAS, [143](#)
- ERR\_SAAS, [143](#)
- fatalfunc\_t, [139](#)
- FE\_WGS84, [144](#)
- FILEPATHSEP, [144](#)
- FTP\_CMD, [131](#)
- FTP\_TIMEOUT, [144](#)
- GAP\_RESION, [144](#)
- HION, [144](#)
- IB\_PPP, [131](#)
- IB\_RTK, [131](#)
- IC\_PPP, [131](#)
- II\_RTK, [131](#)
- IL\_RTK, [132](#)
- INIT\_ZWD, [144](#)
- initlock, [132](#)
- INT\_SWAP\_STAT, [145](#)
- INT\_SWAP\_TRAC, [145](#)
- IONOOPT\_BRDC, [145](#)
- IONOOPT\_EST, [145](#)
- IONOOPT\_IFLC, [145](#)
- IONOOPT\_LEX, [145](#)
- IONOOPT\_OFF, [146](#)
- IONOOPT\_QZS, [146](#)
- IONOOPT\_SBAS, [146](#)
- IONOOPT\_STEC, [146](#)
- IONOOPT\_TEC, [146](#)
- IT\_PPP, [132](#)
- IT\_RTK, [132](#)
- LAM\_CARR, [146](#)
- lock\_t, [132](#)
- MAXAC, [147](#)
- MAXANT, [147](#)
- MAXBAND, [147](#)
- MAXCLI, [147](#)
- MAXDToe, [147](#)
- MAXDToe\_BDS, [147](#)

MAXDTOE\_GAL, 148  
MAXDTOE\_GLO, 148  
MAXDTOE\_QZS, 148  
MAXDTOE\_S, 148  
MAXDTOE\_SBS, 148  
MAXERRMSG, 148  
MAXEXFILE, 149  
MAXFREQ, 149  
MAXGDOP, 149  
MAXLEAPS, 149  
MAXNGEO, 149  
MAXNIGP, 149  
MAXOBS, 150  
MAXOBSBUF, 150  
MAXOBSTYPE, 150  
MAXPRNBDS, 150  
MAXPRNGAL, 150  
MAXPRNGLO, 150  
MAXPRNGPS, 151  
MAXPRNIRN, 151  
MAXPRNLEO, 151  
MAXPRNQZS, 151  
MAXPRNSBS, 151  
MAXRAWLEN, 151  
MAXRCV, 152  
MAXSAT, 152  
MAXSBSAGEF, 152  
MAXSBSAGEL, 152  
MAXSBSMSG, 152  
MAXSBSURA, 152  
MAXSOLBUF, 153  
MAXSOLMSG, 153  
MAXSOLQ, 153  
MAXSTA, 153  
MAXSTATMSG, 153  
MAXSTRMSG, 153  
MAXSTRPATH, 154  
MAXSTRRTK, 154  
MINPRNBDS, 154  
MINPRNGAL, 154  
MINPRNGLO, 154  
MINPRNGPS, 154  
MINPRNIRN, 155  
MINPRNLEO, 155  
MINPRNQZS, 155  
MINPRNQZS\_S, 155  
MINPRNSBS, 155  
MSG\_DISCONN, 133  
NB\_RTK, 133  
NEXOBS, 155  
NF\_RTK, 133  
NFREQ, 156  
NFREQGLO, 156  
NI\_RTK, 133  
NL\_RTK, 133  
NP\_PPP, 133  
NP\_RTK, 134  
NR\_PPP, 134  
NR\_RTK, 134  
NSATBDS, 156  
NSATGAL, 156  
NSATGLO, 156  
NSATGPS, 156  
NSATIRN, 157  
NSATLEO, 157  
NSATQZS, 157  
NSATSBS, 157  
NSYS, 157  
NSYSBDS, 157  
NSYSGAL, 157  
NSYSGLO, 158  
NSYSGPS, 158  
NSYSIRN, 158  
NSYSLEO, 158  
NSYSQZS, 158  
NT\_RTK, 134  
NTRIP\_AGENT, 134  
NTRIP\_CLI\_PORT, 135  
NTRIP\_MAXRSP, 135  
NTRIP\_MAXSTR, 135  
NTRIP\_RSP\_ERROR, 135  
NTRIP\_RSP\_HTTP, 135  
NTRIP\_RSP\_OK\_CLI, 135  
NTRIP\_RSP\_OK\_SVR, 135  
NTRIP\_RSP\_SRCTBL, 135  
NTRIP\_RSP\_TBLEND, 136  
NTRIP\_SVR\_PORT, 136  
NX\_PPP, 136  
NX\_RTK, 136  
PMODE\_DGPS, 158  
PMODE\_FIXED, 158  
PMODE\_KINEMA, 159  
PMODE\_MOVEB, 159  
PMODE\_PPP\_FIXED, 159  
PMODE\_PPP\_KINEMA, 159  
PMODE\_PPP\_STATIC, 159  
PMODE\_SINGLE, 159  
PMODE\_STATIC, 160  
POLYCRC24Q, 160  
POLYCRC32, 160  
POSOPT\_RINEX, 160  
PRN\_HWBIA, 160  
PRN\_HWBIAS, 160  
RE\_WGS84, 161  
REL\_HUMI, 161  
rtk\_lock, 136  
rtk\_unlock, 136  
SERIBUFFSIZE, 161  
socket\_t, 137  
SOLF\_ENU, 161  
SOLF\_GSIF, 161  
SOLF\_LLH, 161  
SOLF\_NMEA, 162  
SOLF\_STAT, 162  
SOLF\_XYZ, 162  
SOLQ\_DGPS, 162



- SOLQ\_DR, [162](#)
- SOLQ\_FIX, [162](#)
- SOLQ\_FLOAT, [163](#)
- SOLQ\_NONE, [163](#)
- SOLQ\_PPP, [163](#)
- SOLQ\_SBAS, [163](#)
- SOLQ\_SINGLE, [163](#)
- STR\_FILE, [137](#)
- STR\_FTP, [137](#)
- STR\_HTTP, [137](#)
- STR\_MODE\_R, [137](#)
- STR\_MODE\_RW, [137](#)
- STR\_MODE\_W, [137](#)
- STR\_NONE, [138](#)
- STR\_NTRIPCLI, [138](#)
- STR\_NTRIPSVR, [138](#)
- STR\_SERIAL, [138](#)
- STR\_TCPCLI, [138](#)
- STR\_TCPSVR, [138](#)
- STR\_UDP, [138](#)
- STRFMT\_NMEA, [163](#)
- STRFMT\_RNXCLK, [164](#)
- STRFMT\_RTCM2, [164](#)
- STRFMT\_RTCM3, [164](#)
- STRFMT\_SBAS, [164](#)
- STRFMT\_SP3, [164](#)
- SYS\_ALL, [164](#)
- SYS\_BDS, [164](#)
- SYS\_GAL, [165](#)
- SYS\_GLO, [165](#)
- SYS\_GPS, [165](#)
- SYS\_IRN, [165](#)
- SYS\_LEO, [165](#)
- SYS\_NONE, [165](#)
- SYS\_QZS, [166](#)
- SYS\_SBS, [166](#)
- TIMES\_GPST, [166](#)
- TIMES\_JST, [166](#)
- TIMES\_UTC, [166](#)
- TIMETAGH\_LEN, [166](#)
- TINTACT, [167](#)
- TROPOPT\_COR, [167](#)
- TROPOPT\_CORG, [167](#)
- TROPOPT\_EST, [167](#)
- TROPOPT\_ESTG, [167](#)
- TROPOPT\_OFF, [167](#)
- TROPOPT\_SAAS, [168](#)
- TROPOPT\_SBAS, [168](#)
- TTOL\_MOVEB, [168](#)
- VAR\_ACC, [168](#)
- VAR\_GRA, [168](#)
- VAR\_HOLDAMB, [168](#)
- VAR\_HWBIAS, [168](#)
- VAR\_POS, [169](#)
- VAR\_VEL, [169](#)
- VER\_RTKLIB, [138](#)
- alias
  - tlcd\_t, [1488](#)
- all\_fft\_codes
  - Acq\_Conf\_Fpga, [330](#)
- Allocator, [352](#)
  - Allocator, [352](#)
  - buffer\_, [353](#)
  - countwords\_, [353](#)
  - operator(), [352](#)
- alm
  - nav\_t, [1192](#)
- ALM\_AF0\_LSB
  - core\_system\_parameters, [248](#)
- ALM\_AF1\_LSB
  - core\_system\_parameters, [249](#)
- ALM\_DELTAI\_LSB
  - core\_system\_parameters, [249](#)
- ALM\_ECC\_LSB
  - core\_system\_parameters, [249](#)
- ALM\_MZERO\_LSB
  - core\_system\_parameters, [249](#)
- ALM\_OMEGA\_LSB
  - core\_system\_parameters, [249](#)
- ALM\_OMEGADOT\_LSB
  - core\_system\_parameters, [249](#)
- ALM\_OMEGAZERO\_LSB
  - core\_system\_parameters, [249](#)
- ALM\_SQUAREA\_LSB
  - core\_system\_parameters, [249](#)
- alm\_t, [353](#)
  - A, [354](#)
  - e, [354](#)
  - f0, [354](#)
  - f1, [354](#)
  - i0, [354](#)
  - M0, [354](#)
  - omg, [354](#)
  - OMG0, [354](#)
  - OMGd, [355](#)
  - sat, [355](#)
  - svconf, [355](#)
  - svh, [355](#)
  - toa, [355](#)
  - toas, [355](#)
  - week, [355](#)
- ALM\_TOA\_LSB
  - core\_system\_parameters, [249](#)
- alpha
  - DSM\_KROOT\_message, [514](#)
- alpha0
  - Gps\_Iono, [946](#)
- alpha1
  - Gps\_Iono, [946](#)
- alpha2
  - Gps\_Iono, [946](#)
- alpha3
  - Gps\_Iono, [946](#)
- ALPHA\_0
  - core\_system\_parameters, [246](#)
- ALPHA\_0\_LSB

- core\_system\_parameters, 249
- ALPHA\_1\_LSB
  - core\_system\_parameters, 249
- ALPHA\_2\_LSB
  - core\_system\_parameters, 249
- ALPHA\_3\_LSB
  - core\_system\_parameters, 250
- always\_output\_gs
  - Obs\_Conf, 1215
- ambc
  - rtk\_t, 1358
- ambc\_t, 356
  - epoch, 356
  - fixcnt, 356
  - flags, 356
  - LC, 356
  - LCv, 357
  - n, 357
- an\_dump\_devname
  - Pvt\_Conf, 1309
- an\_output\_enabled
  - Pvt\_Conf, 1309
- An\_Packet\_Printer, 357
  - ~An\_Packet\_Printer, 358
  - An\_Packet\_Printer, 358
  - close\_serial, 358
  - print\_packet, 358
- an\_packet\_printer.h, 1890, 1891
- an\_packet\_t, 358
  - data, 359
  - header, 359
- an\_rate\_ms
  - Pvt\_Conf, 1309
- antdel
  - prcopt\_t, 1298
- antdes
  - sta\_t, 1464
- antispoofing\_flag
  - Beidou\_Dnav\_Ephemeris, 382
  - Gps\_CNAV\_Ephemeris, 928
  - Gps\_Ephemeris, 942
- antsetup
  - sta\_t, 1464
- antsno
  - sta\_t, 1465
- anttype
  - prcopt\_t, 1298
- AODC
  - Beidou\_Dnav\_Ephemeris, 383
- AODE
  - Beidou\_Dnav\_Ephemeris, 383
- AODO
  - Beidou\_Dnav\_Ephemeris, 383
  - Gps\_Ephemeris, 942
- AODO\_LSB
  - core\_system\_parameters, 250
- applied\_actions
  - ControlThread, 466
- apply\_action
  - GNSSFlowgraph, 913
- apply\_has\_corrections
  - eph\_t, 520
- AR\_ratio\_factor
  - Monitor\_Pvt, 1172
- AR\_ratio\_threshold
  - Monitor\_Pvt, 1172
- armaxiter
  - prcopt\_t, 1298
- ARMODE\_CONT
  - algorithms\_libs\_rtklib, 139
- ARMODE\_FIXHOLD
  - algorithms\_libs\_rtklib, 139
- ARMODE\_INST
  - algorithms\_libs\_rtklib, 139
- ARMODE\_OFF
  - algorithms\_libs\_rtklib, 140
- ARMODE\_PPPAR
  - algorithms\_libs\_rtklib, 140
- ARMODE\_PPPAR\_ILS
  - algorithms\_libs\_rtklib, 140
- ARMODE\_TCAR
  - algorithms\_libs\_rtklib, 140
- ARMODE\_WLNL
  - algorithms\_libs\_rtklib, 140
- array\_signal\_conditioner.h, 1677
- ArraySignalConditioner, 359
  - ~ArraySignalConditioner, 362
  - ArraySignalConditioner, 362
  - connect, 362
  - data\_type\_adapter, 362
  - disconnect, 362
  - get\_left\_block, 362
  - get\_right\_block, 363
  - implementation, 363
  - input\_filter, 363
  - item\_size, 363
  - resampler, 363
  - role, 363
- AS2R
  - core\_system\_parameters, 250
- AS\_status
  - Gps\_Almanac, 922
- asInt
  - pvt\_libs, 172
- assist\_acquisition\_doppler
  - Channel, 429
- asString
  - pvt\_libs, 172
- at
  - algorithms\_libs, 97
- AU
  - core\_system\_parameters, 250
- az
  - solstat\_t, 1446
- azel
  - ssat\_t, 1459

- stec\_t, [1466](#)
- Azimuth
  - Gps\_Acq\_Assist, [917](#)
- b\_io\_context
  - core\_libs, [209](#)
  - core\_monitor, [211](#)
  - pvt\_libs, [172](#)
  - signal\_source\_gr\_blocks, [181](#)
  - tracking\_libs, [200](#)
- b\_sv\_do\_not\_use
  - Sbas\_Ephemeris, [1387](#)
- back
  - algorithms\_libs, [97](#)
- bancroft\_init
  - prcpt\_t, [1298](#)
- baseline
  - prcpt\_t, [1298](#)
- bayesian\_estimation.h, [2271](#), [2273](#)
- Bayesian\_estimator, [363](#)
- bdsmodear
  - prcpt\_t, [1298](#)
- beamformer, [364](#)
- beamformer.h, [1711](#), [1712](#)
- beamformer\_filter.h, [1700](#), [1701](#)
- beamformer\_sptr
  - input\_filter\_gr\_blocks, [89](#)
- BeamformerFilter, [366](#)
  - connect, [368](#)
  - disconnect, [368](#)
  - get\_left\_block, [368](#)
  - get\_right\_block, [368](#)
  - implementation, [368](#)
  - item\_size, [368](#)
  - role, [368](#)
- Beidou\_B1I.h, [2402](#), [2404](#)
- beidou\_b1i\_code\_gen\_complex
  - algorithms\_libs, [97](#)
- beidou\_b1i\_code\_gen\_complex\_sampled
  - algorithms\_libs, [97](#)
- beidou\_b1i\_code\_gen\_float
  - algorithms\_libs, [97](#)
- beidou\_b1i\_code\_gen\_int
  - algorithms\_libs, [97](#)
- BEIDOU\_B1I\_CODE\_LENGTH\_CHIPS
  - core\_system\_parameters, [250](#)
- BEIDOU\_B1I\_CODE\_PERIOD\_MS
  - core\_system\_parameters, [250](#)
- BEIDOU\_B1I\_CODE\_PERIOD\_S
  - core\_system\_parameters, [250](#)
- BEIDOU\_B1I\_CODE\_RATE\_CPS
  - core\_system\_parameters, [250](#)
- BEIDOU\_B1I\_D2\_SECONDARY\_CODE\_STR
  - core\_system\_parameters, [250](#)
- beidou\_b1i\_dll\_pll\_tracking.h, [2179](#), [2180](#)
- BEIDOU\_B1I\_FREQ\_HZ
  - core\_system\_parameters, [250](#)
- BEIDOU\_B1I\_GEO\_PREAMBLE\_LENGTH\_SYMBOLS
  - core\_system\_parameters, [250](#)
- BEIDOU\_B1I\_GEO\_PREAMBLE\_SYMBOLS\_STR
  - core\_system\_parameters, [251](#)
- BEIDOU\_B1I\_GEO\_TELEMETRY\_SYMBOLS\_PER\_BIT
  - core\_system\_parameters, [251](#)
- beidou\_b1i\_pcps\_acquisition.h, [1539](#)
- BEIDOU\_B1I\_PREAMBLE\_DURATION\_MS
  - core\_system\_parameters, [251](#)
- BEIDOU\_B1I\_PREAMBLE\_DURATION\_S
  - core\_system\_parameters, [251](#)
- BEIDOU\_B1I\_PREAMBLE\_LENGTH\_BITS
  - core\_system\_parameters, [251](#)
- BEIDOU\_B1I\_PREAMBLE\_LENGTH\_SYMBOLS
  - core\_system\_parameters, [251](#)
- BEIDOU\_B1I\_SECONDARY\_CODE\_LENGTH
  - core\_system\_parameters, [251](#)
- BEIDOU\_B1I\_SECONDARY\_CODE\_STR
  - core\_system\_parameters, [251](#)
- beidou\_b1i\_signal\_replica.h, [1719](#)
- beidou\_b1i\_telemetry\_decoder.h, [2113](#)
- beidou\_b1i\_telemetry\_decoder\_gs, [369](#)
  - ~beidou\_b1i\_telemetry\_decoder\_gs, [371](#)
  - general\_work, [371](#)
  - set\_channel, [371](#)
  - set\_satellite, [371](#)
- beidou\_b1i\_telemetry\_decoder\_gs.h, [2135](#), [2136](#)
- beidou\_b1i\_telemetry\_decoder\_gs\_sptr
  - telemetry\_decoder\_gr\_blocks, [188](#)
- BEIDOU\_B1I\_TELEMETRY\_RATE\_BITS\_SECOND
  - core\_system\_parameters, [251](#)
- BEIDOU\_B1I\_TELEMETRY\_RATE\_SYMBOLS\_SECOND
  - core\_system\_parameters, [251](#)
- BEIDOU\_B1I\_TELEMETRY\_SYMBOL\_PERIOD\_MS
  - core\_system\_parameters, [251](#)
- BEIDOU\_B1I\_TELEMETRY\_SYMBOLS\_PER\_BIT
  - core\_system\_parameters, [251](#)
- Beidou\_B3I.h, [2405](#), [2407](#)
- beidou\_b3i\_code\_gen\_complex
  - algorithms\_libs, [98](#)
- beidou\_b3i\_code\_gen\_complex\_sampled
  - algorithms\_libs, [98](#)
- beidou\_b3i\_code\_gen\_float
  - algorithms\_libs, [98](#)
- beidou\_b3i\_code\_gen\_int
  - algorithms\_libs, [98](#)
- BEIDOU\_B3I\_CODE\_LENGTH\_CHIPS
  - core\_system\_parameters, [252](#)
- BEIDOU\_B3I\_CODE\_PERIOD\_MS
  - core\_system\_parameters, [252](#)
- BEIDOU\_B3I\_CODE\_PERIOD\_S
  - core\_system\_parameters, [252](#)
- BEIDOU\_B3I\_CODE\_RATE\_CPS
  - core\_system\_parameters, [252](#)
- BEIDOU\_B3I\_D2\_SECONDARY\_CODE\_STR
  - core\_system\_parameters, [252](#)
- beidou\_b3i\_dll\_pll\_tracking.h, [2181](#), [2182](#)
- BEIDOU\_B3I\_FREQ\_HZ
  - core\_system\_parameters, [252](#)
- BEIDOU\_B3I\_GEO\_PREAMBLE\_LENGTH\_SYMBOLS

- core\_system\_parameters, 252
- BEIDOU\_B3I\_GEO\_PREAMBLE\_SYMBOLS\_STR
  - core\_system\_parameters, 252
- BEIDOU\_B3I\_GEO\_TELEMETRY\_SYMBOLS\_PER\_BIT
  - core\_system\_parameters, 252
- beidou\_b3i\_pcps\_acquisition.h, 1542
- BEIDOU\_B3I\_PREAMBLE\_DURATION\_MS
  - core\_system\_parameters, 252
- BEIDOU\_B3I\_PREAMBLE\_DURATION\_S
  - core\_system\_parameters, 252
- BEIDOU\_B3I\_PREAMBLE\_LENGTH\_BITS
  - core\_system\_parameters, 253
- BEIDOU\_B3I\_PREAMBLE\_LENGTH\_SYMBOLS
  - core\_system\_parameters, 253
- BEIDOU\_B3I\_SECONDARY\_CODE\_LENGTH
  - core\_system\_parameters, 253
- BEIDOU\_B3I\_SECONDARY\_CODE\_STR
  - core\_system\_parameters, 253
- beidou\_b3i\_signal\_replica.h, 1720, 1721
- beidou\_b3i\_telemetry\_decoder.h, 2114, 2115
- beidou\_b3i\_telemetry\_decoder\_gs, 371
  - ~beidou\_b3i\_telemetry\_decoder\_gs, 373
  - general\_work, 373
  - set\_channel, 373
  - set\_satellite, 373
- beidou\_b3i\_telemetry\_decoder\_gs.h, 2137, 2139
- beidou\_b3i\_telemetry\_decoder\_gs\_sptr
  - telemetry\_decoder\_gr\_blocks, 188
- BEIDOU\_B3I\_TELEMETRY\_RATE\_BITS\_SECOND
  - core\_system\_parameters, 253
- BEIDOU\_B3I\_TELEMETRY\_RATE\_SYMBOLS\_SECOND
  - core\_system\_parameters, 253
- BEIDOU\_B3I\_TELEMETRY\_SYMBOL\_PERIOD\_MS
  - core\_system\_parameters, 253
- BEIDOU\_B3I\_TELEMETRY\_SYMBOLS\_PER\_BIT
  - core\_system\_parameters, 253
- Beidou\_DNAV.h, 2407, 2412
- Beidou\_Dnav\_Almanac, 374
  - Beidou\_Dnav\_Almanac, 376
  - serialize, 377
  - SV\_health, 377
- beidou\_dnav\_almanac.h, 2416, 2417
- beidou\_dnav\_almanac\_map
  - Rtklib\_Solver, 1372
- BEIDOU\_DNAV\_BDT2GPST\_LEAP\_SEC\_OFFSET
  - core\_system\_parameters, 253
- BEIDOU\_DNAV\_BDT2GPST\_WEEK\_NUM\_OFFSET
  - core\_system\_parameters, 253
- Beidou\_Dnav\_Ephemeris, 377
  - alert\_flag, 382
  - antispoofing\_flag, 382
  - AODC, 383
  - AODE, 383
  - AODO, 383
  - Beidou\_Dnav\_Ephemeris, 382
  - fit\_interval\_flag, 383
  - integrity\_status\_flag, 383
  - nav\_type, 383
  - satelliteBlock, 383
  - serialize, 382
  - sig\_type, 384
  - spare1, 384
  - spare2, 384
  - SV\_accuracy, 384
  - SV\_health, 384
  - TGD1, 384
  - TGD2, 384
- beidou\_dnav\_ephemeris.h, 2418, 2420
- beidou\_dnav\_ephemeris\_map
  - Rtklib\_Solver, 1372
- Beidou\_Dnav\_Iono, 384
  - Beidou\_Dnav\_Iono, 387
- beidou\_dnav\_iono
  - Rtklib\_Solver, 1372
- beidou\_dnav\_iono.h, 2421, 2423
- Beidou\_Dnav\_Navigation\_Message, 387
  - Beidou\_Dnav\_Navigation\_Message, 389
  - d1\_subframe\_decoder, 389
  - d2\_subframe\_decoder, 389
  - get\_ephemeris, 389
  - get\_flag\_CRC\_test, 389
  - get\_flag\_new\_SOW\_available, 389
  - get\_iono, 389
  - get\_SOW, 389
  - get\_utc\_model, 389
  - have\_new\_almanac, 390
  - have\_new\_ephemeris, 390
  - have\_new\_iono, 390
  - have\_new\_utc\_model, 390
  - set\_flag\_new\_SOW\_available, 390
  - set\_satellite\_PRN, 390
  - set\_signal\_type, 390
  - utc\_time, 390
- beidou\_dnav\_navigation\_message.h, 2423, 2424
- BEIDOU\_DNAV\_PREAMBLE
  - core\_system\_parameters, 253
- BEIDOU\_DNAV\_PREAMBLE\_LENGTH\_BITS
  - core\_system\_parameters, 254
- BEIDOU\_DNAV\_PREAMBLE\_LENGTH\_SYMBOLS
  - core\_system\_parameters, 254
- BEIDOU\_DNAV\_PREAMBLE\_PERIOD\_SYMBOLS
  - core\_system\_parameters, 254
- BEIDOU\_DNAV\_SUBFRAME\_DATA\_BITS
  - core\_system\_parameters, 254
- BEIDOU\_DNAV\_SUBFRAME\_SYMBOLS
  - core\_system\_parameters, 254
- Beidou\_Dnav\_Utc\_Model, 390
  - A0\_GAL, 392
  - A0\_GLO, 392
  - A0\_GPS, 392
  - A0\_UTC, 392
  - A1\_GAL, 392
  - A1\_GLO, 392
  - A1\_GPS, 393
  - A1\_UTC, 393
  - DeltaT\_LS, 393

- DeltaT\_LSF, 393
- DN, 393
- serialize, 392
- valid, 393
- WN\_LSF, 393
- beidou\_dnav\_utc\_model
  - Rtklib\_Solver, 1372
- beidou\_dnav\_utc\_model.h, 2428, 2430
- BEIDOU\_DNAV\_WORD\_LENGTH\_BITS
  - core\_system\_parameters, 254
- BEIDOU\_DNAV\_WORDS\_SUBFRAME
  - core\_system\_parameters, 254
- BEIDOU\_F
  - core\_system\_parameters, 254
- BEIDOU\_GM
  - core\_system\_parameters, 254
- BEIDOU\_OMEGA\_EARTH\_DOT
  - core\_system\_parameters, 254
- BeidouB1iDIIPIITracking, 393
  - connect, 396
  - disconnect, 396
  - get\_left\_block, 396
  - get\_right\_block, 396
  - implementation, 396
  - item\_size, 396
  - role, 396
  - set\_channel, 397
  - set\_gnss\_synchro, 397
  - start\_tracking, 397
  - stop\_tracking, 397
- BeidouB1iPcpsAcquisition, 397
  - connect, 401
  - disconnect, 401
  - get\_left\_block, 401
  - get\_right\_block, 401
  - implementation, 401
  - init, 401
  - item\_size, 401
  - mag, 401
  - reset, 401
  - role, 401
  - set\_channel, 401
  - set\_channel\_fsm, 402
  - set\_doppler\_max, 402
  - set\_doppler\_step, 402
  - set\_gnss\_synchro, 402
  - set\_local\_code, 402
  - set\_resampler\_latency, 402
  - set\_state, 402
  - set\_threshold, 402
  - stop\_acquisition, 403
- BeidouB1iTelemetryDecoder, 403
  - connect, 406
  - disconnect, 406
  - get\_left\_block, 406
  - get\_right\_block, 406
  - implementation, 406
  - item\_size, 406
  - reset, 406
  - role, 407
  - set\_channel, 407
  - set\_satellite, 407
- BeidouB3iDIIPIITracking, 407
  - connect, 410
  - disconnect, 410
  - get\_left\_block, 410
  - get\_right\_block, 410
  - implementation, 410
  - item\_size, 410
  - role, 410
  - set\_channel, 411
  - set\_gnss\_synchro, 411
  - start\_tracking, 411
  - stop\_tracking, 411
- BeidouB3iPcpsAcquisition, 411
  - connect, 414
  - disconnect, 414
  - get\_left\_block, 415
  - get\_right\_block, 415
  - implementation, 415
  - init, 415
  - item\_size, 415
  - mag, 415
  - reset, 415
  - role, 415
  - set\_channel, 415
  - set\_channel\_fsm, 415
  - set\_doppler\_max, 416
  - set\_doppler\_step, 416
  - set\_gnss\_synchro, 416
  - set\_local\_code, 416
  - set\_resampler\_latency, 416
  - set\_state, 416
  - set\_threshold, 416
  - stop\_acquisition, 416
- BeidouB3iTelemetryDecoder, 417
  - connect, 420
  - disconnect, 420
  - get\_left\_block, 420
  - get\_right\_block, 420
  - implementation, 420
  - item\_size, 420
  - reset, 420
  - role, 421
  - set\_channel, 421
  - set\_satellite, 421
- beta0
  - Gps\_Iono, 946
- beta1
  - Gps\_Iono, 947
- beta2
  - Gps\_Iono, 947
- beta3
  - Gps\_Iono, 947
- BETA\_0\_LSB
  - core\_system\_parameters, 254

- BETA\_1\_LSB
  - core\_system\_parameters, [255](#)
- BETA\_2\_LSB
  - core\_system\_parameters, [255](#)
- BETA\_3\_LSB
  - core\_system\_parameters, [255](#)
- BGD\_E1\_E5A\_5\_LSB
  - core\_system\_parameters, [255](#)
- BGD\_E1\_E5B\_5\_LSB
  - core\_system\_parameters, [255](#)
- BGD\_E1E5a
  - Galileo\_Ephemeris, [621](#)
- BGD\_E1E5b
  - Galileo\_Ephemeris, [621](#)
- bias
  - fcdb\_t, [530](#)
- bin\_to\_binary\_data
  - Rtcm, [1341](#)
- bin\_to\_double
  - Rtcm, [1341](#)
- bin\_to\_hex
  - Rtcm, [1341](#)
- bin\_to\_uint
  - Rtcm, [1341](#)
- binary\_data\_to\_bin
  - Rtcm, [1341](#)
- bit\_selection
  - Fpga\_dynamic\_bit\_selection, [583](#)
- bit\_synchronization\_time\_limit\_s
  - Dll\_Pll\_Conf, [493](#)
  - Dll\_Pll\_Conf\_Fpga, [499](#)
  - Kf\_Conf, [1135](#)
- bit\_transition\_flag
  - Acq\_Conf, [325](#)
- bite
  - sbsigpband\_t, [1399](#)
- bits
  - sbsigpband\_t, [1399](#)
- bits.h, [2160](#), [2161](#)
- BITS\_IN\_OCTET
  - core\_system\_parameters, [255](#)
- bits\_to\_item\_size
  - ion\_gsms\_chunk\_data.h, [2094](#)
- bitshift
  - IONGSMSChunkUnpackingCtx< WT >, [1119](#)
- blocking
  - Acq\_Conf, [325](#)
- blocking\_on\_standby
  - Acq\_Conf, [325](#)
- bstar
  - tlcd\_t, [1488](#)
- buff
  - ntrip\_t, [1213](#)
  - raw\_t, [1324](#)
  - rtcm\_t, [1354](#)
  - rtksvr\_t, [1376](#)
  - solbuf\_t, [1441](#)
- buffer
  - ad936x\_iio\_samples, [339](#)
  - fpga\_dma-proxy.h, [2081](#)
- buffer\_
  - Allocator, [353](#)
  - Deleter, [482](#)
- buffsize
  - rtksvr\_t, [1376](#)
- bw\_hz
  - stream\_cfg, [1467](#)
- byte\_to\_short.h, [1680](#), [1681](#)
- byte\_x2\_to\_complex\_byte, [421](#)
- byte\_x2\_to\_complex\_byte.h, [1722](#), [1723](#)
- byte\_x2\_to\_complex\_byte\_sptr
  - algorithms\_libs, [95](#)
- ByteToShort, [422](#)
  - connect, [425](#)
  - disconnect, [425](#)
  - get\_left\_block, [425](#)
  - get\_right\_block, [425](#)
  - implementation, [425](#)
  - item\_size, [425](#)
  - role, [425](#)
- c0
  - v27\_poly\_t, [1529](#)
- c1
  - v27\_poly\_t, [1529](#)
- C\_IC\_4\_LSB
  - core\_system\_parameters, [255](#)
- C\_IC\_LSB
  - core\_system\_parameters, [255](#)
- C\_IS\_4\_LSB
  - core\_system\_parameters, [255](#)
- C\_IS\_LSB
  - core\_system\_parameters, [255](#)
- C\_RC\_3\_LSB
  - core\_system\_parameters, [255](#)
- C\_RC\_LSB
  - core\_system\_parameters, [255](#)
- C\_RS\_3\_LSB
  - core\_system\_parameters, [256](#)
- C\_RS\_LSB
  - core\_system\_parameters, [256](#)
- C\_UC\_3\_LSB
  - core\_system\_parameters, [256](#)
- C\_UC\_LSB
  - core\_system\_parameters, [256](#)
- C\_US\_3\_LSB
  - core\_system\_parameters, [256](#)
- C\_US\_LSB
  - core\_system\_parameters, [256](#)
- ca\_codes
  - Dll\_Pll\_Conf\_Fpga, [499](#)
- CalculateSlope
  - tracking\_libs, [201](#)
- CalculateSlopeAbs
  - tracking\_libs, [201](#)
- carrier\_aiding
  - Dll\_Pll\_Conf, [493](#)

- Dll\_Pll\_Conf\_Fpga, [500](#)
- carrier\_disc\_sd\_rads
  - Kf\_Conf, [1135](#)
- Carrier\_Doppler\_hz
  - Gnss\_Synchro, [902](#)
- carrier\_freq\_hz
  - TrackingCmd, [1498](#)
- carrier\_freq\_rate\_hz\_s
  - TrackingCmd, [1498](#)
- carrier\_freq\_rate\_sd\_hz\_s
  - Kf\_Conf, [1135](#)
- carrier\_freq\_sd\_hz
  - Kf\_Conf, [1136](#)
- carrier\_lock\_detector
  - tracking\_libs, [201](#)
- carrier\_lock\_test\_smoother\_alpha
  - Dll\_Pll\_Conf, [493](#)
  - Dll\_Pll\_Conf\_Fpga, [500](#)
  - Kf\_Conf, [1136](#)
- carrier\_lock\_test\_smoother\_samples
  - Dll\_Pll\_Conf, [494](#)
  - Dll\_Pll\_Conf\_Fpga, [500](#)
  - Kf\_Conf, [1136](#)
- carrier\_lock\_th
  - Dll\_Pll\_Conf, [494](#)
  - Dll\_Pll\_Conf\_Fpga, [500](#)
  - Kf\_Conf, [1136](#)
- Carrier\_phase\_rads
  - Gnss\_Synchro, [902](#)
- carrier\_phase\_sd\_rad
  - Kf\_Conf, [1136](#)
- Carrier\_wipeoff\_multicorrelator\_resampler
  - Fpga\_Multicorrelator\_8sc, [585](#)
- cart2geo
  - algorithms\_libs, [98](#)
- cart2utm
  - algorithms\_libs, [99](#)
- cbias
  - nav\_t, [1192](#)
  - ssr\_t, [1462](#)
- CED\_af0red\_LSB
  - core\_system\_parameters, [256](#)
- CED\_af1red\_LSB
  - core\_system\_parameters, [256](#)
- CED\_DeltaAred\_LSB
  - core\_system\_parameters, [256](#)
- CED\_DeltaI0red\_LSB
  - core\_system\_parameters, [256](#)
- CED\_exred\_LSB
  - core\_system\_parameters, [256](#)
- CED\_eyred\_LSB
  - core\_system\_parameters, [256](#)
- CED\_lambda0red\_LSB
  - core\_system\_parameters, [257](#)
- CED\_Omega0red\_LSB
  - core\_system\_parameters, [257](#)
- cell\_mask
  - Galileo\_HAS\_data, [632](#)
- cell\_mask\_availability\_flag
  - Galileo\_HAS\_data, [632](#)
- cellmask
  - msm\_h\_t, [1177](#)
- cerr
  - exterr\_t, [529](#)
- Channel, [81](#), [426](#)
  - ~Channel, [428](#)
  - acquisition, [429](#)
  - assist\_acquisition\_doppler, [429](#)
  - Channel, [428](#)
  - connect, [429](#)
  - disconnect, [429](#)
  - get\_left\_block, [429](#)
  - get\_left\_block\_acq, [429](#)
  - get\_left\_block\_trk, [429](#)
  - get\_right\_block, [429](#)
  - get\_right\_block\_acq, [429](#)
  - get\_right\_block\_trk, [429](#)
  - get\_signal, [430](#)
  - implementation, [430](#)
  - item\_size, [430](#)
  - role, [430](#)
  - set\_signal, [430](#)
  - start\_acquisition, [430](#)
  - stop\_channel, [430](#)
  - telemetry, [430](#)
  - tracking, [430](#)
- channel.h, [1671](#), [1672](#)
- channel\_adapters, [81](#)
- Channel\_Event, [431](#)
  - channel\_id, [431](#)
  - event\_type, [431](#)
- channel\_event.h, [2326](#), [2328](#)
- channel\_event\_sptr
  - core\_libs, [209](#)
- channel\_fsm.h, [1673](#), [1674](#)
- Channel\_ID
  - Gnss\_Synchro, [902](#)
- channel\_id
  - Channel\_Event, [431](#)
- channel\_interface.h, [2314](#), [2315](#)
- channel\_libs, [82](#)
  - channel\_msg\_receiver\_cc\_sptr, [83](#)
- channel\_msg\_receiver\_cc, [431](#)
  - ~channel\_msg\_receiver\_cc, [433](#)
- channel\_msg\_receiver\_cc.h, [1675](#), [1676](#)
- channel\_msg\_receiver\_cc\_sptr
  - channel\_libs, [83](#)
- channel\_status\_msg\_receiver, [433](#)
  - ~channel\_status\_msg\_receiver, [434](#)
  - get\_current\_status\_map, [434](#)
  - get\_current\_status\_pvt, [435](#)
- channel\_status\_msg\_receiver.h, [2328](#), [2329](#)
- channel\_status\_msg\_receiver\_sptr
  - core\_libs, [209](#)
- ChannelFsm, [435](#)
- ChannelInterface, [436](#)



- get\_left\_block, 439
- get\_left\_block\_acq, 439
- get\_left\_block\_trk, 439
- get\_right\_block, 439
- get\_right\_block\_acq, 439
- get\_right\_block\_trk, 439
- set\_signal, 439
- start\_acquisition, 440
- stop\_channel, 440
- check\_buffer\_overflow\_and\_monitor\_buffer\_status
  - Fpga\_buffer\_monitor, 580
- check\_CRC
  - Rtcm, 1341
- chips\_per\_second
  - Acq\_Conf, 325
- CHISQR
  - algorithms\_libs\_rtklib, 140
- Cic
  - Gnss\_Ephemeris, 874
- cic
  - eph\_t, 520
- cid
  - DSM\_nma\_header, 516
- cidkr
  - DSM\_KROOT\_message, 514
- Cis
  - Gnss\_Ephemeris, 874
- cis
  - eph\_t, 520
- cl\_fft\_plan, 441
  - context, 442
  - dim, 442
  - format, 442
  - kernel\_info, 442
  - kernel\_string, 442
  - last\_batch\_size, 442
  - max\_localmem\_fft\_size, 442
  - max\_radix, 442
  - max\_work\_item\_per\_workgroup, 442
  - min\_mem\_coalesce\_width, 442
  - n, 443
  - num\_kernels, 443
  - num\_local\_mem\_banks, 443
  - program, 443
  - temp\_buffer\_needed, 443
  - tempmemobj, 443
  - tempmemobj\_imag, 443
  - tempmemobj\_real, 443
  - twist\_kernel, 443
- CL\_SILENCE\_DEPRECATED
  - pcps\_openc1\_acquisition\_cc.h, 1650
- CLAMP\_DIFF
  - telemetry\_decoder\_libswiftcnv, 190
- clear
  - algorithms\_libs, 99
  - Concurrent\_Queue< Data >, 458
- clear\_ephemeris
  - Rtklib\_Pvt, 1363
- rtklib\_pvt\_gs, 1367
- clFFT.h, 1771, 1773
  - clFFT\_DataFormat, 1773
  - clFFT\_Dimension, 1773
  - clFFT\_Direction, 1773
  - clFFT\_Plan, 1773
- clFFT\_Complex, 444
  - imag, 444
  - real, 444
- clFFT\_DataFormat
  - clFFT.h, 1773
- clFFT\_Dim3, 445
  - x, 445
  - y, 445
  - z, 445
- clFFT\_Dimension
  - clFFT.h, 1773
- clFFT\_Direction
  - clFFT.h, 1773
- clFFT\_Plan
  - clFFT.h, 1773
- clFFT\_SplitComplex, 446
  - imag, 446
  - real, 446
- cli
  - tcpsvr\_t, 1481
- clk
  - pclk\_t, 1241
- clk\_ext
  - msm\_h\_t, 1177
- clk\_str
  - msm\_h\_t, 1177
- clksin
  - algorithms\_libs, 99
- clock\_correction\_m
  - HAS\_clock\_corrections, 1089
- clock\_fullset\_flag
  - mt1\_header, 1179
- clock\_subset\_flag
  - mt1\_header, 1179
- close\_device
  - Fpga\_Acquisition, 577
- close\_serial
  - An\_Packet\_Printer, 358
- closesocket
  - algorithms\_libs\_rtklib, 130
- clsin
  - algorithms\_libs, 99
- CN0\_dB\_hz
  - Gnss\_Synchro, 902
- cn0\_m2m4\_estimator
  - tracking\_libs, 201
- cn0\_min
  - Dll\_Pll\_Conf, 494
  - Dll\_Pll\_Conf\_Fpga, 500
  - Kf\_Conf, 1136
- cn0\_samples
  - Dll\_Pll\_Conf, 494



- Dll\_Pll\_Conf\_Fpga, [500](#)
- Kf\_Conf, [1136](#)
- cn0\_smoother\_alpha
  - Dll\_Pll\_Conf, [494](#)
  - Dll\_Pll\_Conf\_Fpga, [500](#)
  - Kf\_Conf, [1136](#)
- cn0\_smoother\_samples
  - Dll\_Pll\_Conf, [494](#)
  - Dll\_Pll\_Conf\_Fpga, [500](#)
  - Kf\_Conf, [1136](#)
- cn0\_svn\_estimator
  - tracking\_libs, [201](#)
- CNAV\_A0\_LSB
  - core\_system\_parameters, [257](#)
- CNAV\_A1\_LSB
  - core\_system\_parameters, [257](#)
- CNAV\_A2\_LSB
  - core\_system\_parameters, [257](#)
- CNAV\_A\_DOT\_LSB
  - core\_system\_parameters, [257](#)
- CNAV\_A\_REF
  - core\_system\_parameters, [257](#)
- CNAV\_AF0\_LSB
  - core\_system\_parameters, [257](#)
- CNAV\_AF1\_LSB
  - core\_system\_parameters, [257](#)
- CNAV\_AF2\_LSB
  - core\_system\_parameters, [257](#)
- CNAV\_ALPHA0\_LSB
  - core\_system\_parameters, [257](#)
- CNAV\_ALPHA1\_LSB
  - core\_system\_parameters, [257](#)
- CNAV\_ALPHA2\_LSB
  - core\_system\_parameters, [258](#)
- CNAV\_ALPHA3\_LSB
  - core\_system\_parameters, [258](#)
- CNAV\_BETA0\_LSB
  - core\_system\_parameters, [258](#)
- CNAV\_BETA1\_LSB
  - core\_system\_parameters, [258](#)
- CNAV\_BETA2\_LSB
  - core\_system\_parameters, [258](#)
- CNAV\_BETA3\_LSB
  - core\_system\_parameters, [258](#)
- CNAV\_CIC\_LSB
  - core\_system\_parameters, [258](#)
- CNAV\_CIS\_LSB
  - core\_system\_parameters, [258](#)
- CNAV\_CRC\_LSB
  - core\_system\_parameters, [258](#)
- CNAV\_CRS\_LSB
  - core\_system\_parameters, [258](#)
- CNAV\_CUC\_LSB
  - core\_system\_parameters, [258](#)
- CNAV\_CUS\_LSB
  - core\_system\_parameters, [258](#)
- CNAV\_DELTA\_A\_LSB
  - core\_system\_parameters, [259](#)
- CNAV\_DELTA\_N0\_DOT\_LSB
  - core\_system\_parameters, [259](#)
- CNAV\_DELTA\_N0\_LSB
  - core\_system\_parameters, [259](#)
- CNAV\_DELTA\_OMEGA\_DOT\_LSB
  - core\_system\_parameters, [259](#)
- CNAV\_DELTA\_TLS\_LSB
  - core\_system\_parameters, [259](#)
- CNAV\_DELTA\_TLSF\_LSB
  - core\_system\_parameters, [259](#)
- CNAV\_DN\_LSB
  - core\_system\_parameters, [259](#)
- CNAV\_E\_ECCENTRICITY\_LSB
  - core\_system\_parameters, [259](#)
- CNAV\_I0\_DOT\_LSB
  - core\_system\_parameters, [259](#)
- CNAV\_I0\_LSB
  - core\_system\_parameters, [259](#)
- CNAV\_ISCL1\_LSB
  - core\_system\_parameters, [259](#)
- CNAV\_ISCL2\_LSB
  - core\_system\_parameters, [259](#)
- CNAV\_ISCL5I\_LSB
  - core\_system\_parameters, [260](#)
- CNAV\_ISCL5Q\_LSB
  - core\_system\_parameters, [260](#)
- CNAV\_M0\_LSB
  - core\_system\_parameters, [260](#)
- cnav\_msg.h, [2162](#), [2163](#)
- cnav\_msg\_decoder\_t, [446](#)
  - part1, [448](#)
  - part2, [448](#)
- cnav\_msg\_t, [448](#)
  - alert, [449](#)
  - msg\_id, [449](#)
  - prn, [449](#)
  - raw\_msg, [449](#)
  - tow, [449](#)
- CNAV\_OMEGA0\_LSB
  - core\_system\_parameters, [260](#)
- CNAV\_OMEGA\_DOT\_REF
  - core\_system\_parameters, [260](#)
- CNAV\_OMEGA\_LSB
  - core\_system\_parameters, [260](#)
- CNAV\_TGD\_LSB
  - core\_system\_parameters, [260](#)
- CNAV\_TOC\_LSB
  - core\_system\_parameters, [260](#)
- CNAV\_TOE1\_LSB
  - core\_system\_parameters, [260](#)
- CNAV\_TOE2\_LSB
  - core\_system\_parameters, [260](#)
- CNAV\_TOP1\_LSB
  - core\_system\_parameters, [260](#)
- CNAV\_TOP2\_LSB
  - core\_system\_parameters, [260](#)
- CNAV\_TOT\_LSB
  - core\_system\_parameters, [261](#)

CNAV\_TOW\_LSB  
     core\_system\_parameters, 261  
 cnav\_v27\_part\_t, 449  
     crc\_ok, 451  
     dec, 451  
     decisions, 451  
     decoded, 451  
     init, 451  
     invert, 451  
     message\_lock, 451  
     n\_crc\_fail, 451  
     n\_decoded, 452  
     n\_symbols, 452  
     preamble\_seen, 452  
     symbols, 452  
 CNAV\_WN\_LSF\_LSB  
     core\_system\_parameters, 261  
 CNAV\_WN\_OT\_LSB  
     core\_system\_parameters, 261  
 code  
     eph\_t, 520  
     obsd\_t, 1218  
     pcv\_t, 1284  
 code\_bias  
     Galileo\_HAS\_data, 632  
 code\_bias\_flag  
     mt1\_header, 1179  
 code\_bias\_m  
     HAS\_obs\_corrections, 1089  
 code\_disc\_sd\_chips  
     Kf\_Conf, 1136  
 code\_freq\_chips  
     TrackingCmd, 1498  
 CODE\_L1A  
     core\_system\_parameters, 261  
 CODE\_L1B  
     core\_system\_parameters, 261  
 CODE\_L1C  
     core\_system\_parameters, 261  
 CODE\_L1E  
     core\_system\_parameters, 261  
 CODE\_L1I  
     core\_system\_parameters, 261  
 CODE\_L1L  
     core\_system\_parameters, 261  
 CODE\_L1M  
     core\_system\_parameters, 261  
 CODE\_L1N  
     core\_system\_parameters, 262  
 CODE\_L1P  
     core\_system\_parameters, 262  
 CODE\_L1Q  
     core\_system\_parameters, 262  
 CODE\_L1S  
     core\_system\_parameters, 262  
 CODE\_L1W  
     core\_system\_parameters, 262  
 CODE\_L1X  
     core\_system\_parameters, 262  
 CODE\_L1Y  
     core\_system\_parameters, 262  
 CODE\_L1Z  
     core\_system\_parameters, 262  
 CODE\_L2C  
     core\_system\_parameters, 262  
 CODE\_L2D  
     core\_system\_parameters, 262  
 CODE\_L2I  
     core\_system\_parameters, 263  
 CODE\_L2L  
     core\_system\_parameters, 263  
 CODE\_L2M  
     core\_system\_parameters, 263  
 CODE\_L2N  
     core\_system\_parameters, 263  
 CODE\_L2P  
     core\_system\_parameters, 263  
 CODE\_L2Q  
     core\_system\_parameters, 263  
 CODE\_L2S  
     core\_system\_parameters, 263  
 CODE\_L2W  
     core\_system\_parameters, 263  
 CODE\_L2X  
     core\_system\_parameters, 263  
 CODE\_L2Y  
     core\_system\_parameters, 263  
 CODE\_L3I  
     core\_system\_parameters, 264  
 CODE\_L3Q  
     core\_system\_parameters, 264  
 CODE\_L3X  
     core\_system\_parameters, 264  
 CODE\_L5A  
     core\_system\_parameters, 264  
 CODE\_L5B  
     core\_system\_parameters, 264  
 CODE\_L5C  
     core\_system\_parameters, 264  
 CODE\_L5I  
     core\_system\_parameters, 264  
 CODE\_L5Q  
     core\_system\_parameters, 264  
 CODE\_L5X  
     core\_system\_parameters, 264  
 CODE\_L6A  
     core\_system\_parameters, 264  
 CODE\_L6B  
     core\_system\_parameters, 265  
 CODE\_L6C  
     core\_system\_parameters, 265  
 CODE\_L6I  
     core\_system\_parameters, 265  
 CODE\_L6L  
     core\_system\_parameters, 265  
 CODE\_L6Q

- core\_system\_parameters, [265](#)
- CODE\_L6S
  - core\_system\_parameters, [265](#)
- CODE\_L6X
  - core\_system\_parameters, [265](#)
- CODE\_L6Z
  - core\_system\_parameters, [265](#)
- CODE\_L7I
  - core\_system\_parameters, [265](#)
- CODE\_L7Q
  - core\_system\_parameters, [265](#)
- CODE\_L7X
  - core\_system\_parameters, [266](#)
- CODE\_L8I
  - core\_system\_parameters, [266](#)
- CODE\_L8Q
  - core\_system\_parameters, [266](#)
- CODE\_L8X
  - core\_system\_parameters, [266](#)
- CODE\_L9A
  - core\_system\_parameters, [266](#)
- CODE\_L9B
  - core\_system\_parameters, [266](#)
- CODE\_L9C
  - core\_system\_parameters, [266](#)
- CODE\_L9X
  - core\_system\_parameters, [266](#)
- code\_length
  - Acq\_Conf\_Fpga, [330](#)
- code\_length\_chips
  - Acq\_Conf\_Fpga, [330](#)
  - Dll\_Pll\_Conf\_Fpga, [500](#)
- CODE\_NONE
  - core\_system\_parameters, [266](#)
- code\_on\_L2
  - Gps\_Ephemeris, [942](#)
- Code\_Phase
  - Gps\_Acq\_Assist, [917](#)
- Code\_Phase\_int
  - Gps\_Acq\_Assist, [917](#)
- Code\_phase\_samples
  - Gnss\_Synchro, [903](#)
- code\_phase\_sd\_chips
  - Kf\_Conf, [1136](#)
- Code\_Phase\_window
  - Gps\_Acq\_Assist, [917](#)
- code\_rate\_cps
  - Acq\_Conf\_Fpga, [330](#)
- code\_samples\_per\_chip
  - Dll\_Pll\_Conf\_Fpga, [500](#)
- CODES\_BDS
  - rtklib\_rtc3.h, [1836](#)
- CODES\_GAL
  - rtklib\_rtc3.h, [1836](#)
- CODES\_GLO
  - rtklib\_rtc3.h, [1836](#)
- CODES\_GPS
  - rtklib\_rtc3.h, [1836](#)
- CODES\_QZS
  - rtklib\_rtc3.h, [1837](#)
- CODES\_SBS
  - rtklib\_rtc3.h, [1837](#)
- codesmooth
  - prcopt\_t, [1298](#)
- coef
  - lexion\_t, [1150](#)
- cog
  - Monitor\_Pvt, [1172](#)
- Command\_Event, [452](#)
  - command\_id, [453](#)
  - event\_type, [453](#)
- command\_event.h, [2330](#), [2331](#)
- command\_event\_sptr
  - core\_libs, [209](#)
- command\_id
  - Command\_Event, [453](#)
- comment
  - opt\_t, [1222](#)
- COMMENTH
  - algorithms\_libs\_rtklib, [130](#)
- complex\_byte\_to\_float\_x2, [453](#)
- complex\_byte\_to\_float\_x2.h, [1724](#)
- complex\_byte\_to\_float\_x2\_sptr
  - algorithms\_libs, [95](#)
- complex\_exp\_gen
  - algorithms\_libs, [100](#)
- complex\_exp\_gen\_conj
  - algorithms\_libs, [100](#)
- complex\_float\_to\_complex\_byte, [454](#)
- complex\_float\_to\_complex\_byte.h, [1725](#), [1726](#)
- complex\_float\_to\_complex\_byte\_sptr
  - algorithms\_libs, [95](#)
- compute\_CMAC\_AES
  - Gnss\_Crypto, [868](#)
- compute\_eph
  - Galileo\_Reduced\_CED, [652](#)
- compute\_GLONASS\_time
  - Glonass\_Gnav\_Ephemeris, [798](#)
- compute\_HMAC\_SHA\_256
  - Gnss\_Crypto, [868](#)
- compute\_SHA3\_256
  - Gnss\_Crypto, [869](#)
- compute\_SHA\_256
  - Gnss\_Crypto, [869](#)
- computed\_tag
  - Tag, [1473](#)
- computeSamplesInFile
  - FileSourceBase, [551](#)
- Concurrent\_Map< Data >, [456](#)
  - get\_map\_copy, [456](#)
  - read, [456](#)
  - size, [457](#)
  - write, [457](#)
- concurrent\_map.h, [2379](#)
- Concurrent\_Queue< Data >, [457](#)
  - clear, [458](#)

- empty, [458](#)
- push, [458](#), [459](#)
- size, [459](#)
- timed\_wait\_and\_pop, [459](#)
- try\_pop, [459](#)
- wait\_and\_pop, [459](#)
- concurrent\_queue.h, [2380](#), [2381](#)
- conditioner\_adapters, [83](#)
- configuration\_interface.h, [2315](#), [2316](#)
- ConfigurationInterface, [459](#)
- configure\_acquisition
  - Fpga\_Acquisition, [577](#)
- conjugate\_cc, [461](#)
- conjugate\_cc.h, [1727](#), [1728](#)
- conjugate\_cc\_sptr
  - algorithms\_libs, [96](#)
- conjugate\_ic, [463](#)
- conjugate\_ic.h, [1729](#), [1730](#)
- conjugate\_ic\_sptr
  - algorithms\_libs, [96](#)
- conjugate\_sc, [464](#)
- conjugate\_sc.h, [1730](#), [1731](#)
- conjugate\_sc\_sptr
  - algorithms\_libs, [96](#)
- connect
  - Ad936xCustomSignalSource, [344](#)
  - Adrv9361z7035SignalSourceFPGA, [348](#)
  - ArraySignalConditioner, [362](#)
  - BeamformerFilter, [368](#)
  - BeidouB1iDIIPIITracking, [396](#)
  - BeidouB1iPcpsAcquisition, [401](#)
  - BeidouB1iTelemetryDecoder, [406](#)
  - BeidouB3iDIIPIITracking, [410](#)
  - BeidouB3iPcpsAcquisition, [414](#)
  - BeidouB3iTelemetryDecoder, [420](#)
  - ByteToShort, [425](#)
  - Channel, [429](#)
  - CshortToGrComplex, [475](#)
  - CustomUDPSignalSource, [480](#)
  - DirectResamplerConditioner, [491](#)
  - DMASignalSourceFPGA, [511](#)
  - FifoSignalSource, [535](#)
  - FileSourceBase, [551](#)
  - FirFilter, [560](#)
  - FlexibandSignalSource, [564](#)
  - Fmcomms2SignalSource, [568](#)
  - Fmcomms5SignalSourceFPGA, [571](#)
  - FreqXlatingFirFilter, [593](#)
  - GalileoE1BTelemetryDecoder, [663](#)
  - GalileoE1DIIPIIVemlTracking, [667](#)
  - GalileoE1DIIPIIVemlTrackingFpga, [672](#)
  - GalileoE1Pcps8msAmbiguousAcquisition, [676](#)
  - GalileoE1PcpsAmbiguousAcquisition, [683](#)
  - GalileoE1PcpsAmbiguousAcquisitionFpga, [689](#)
  - GalileoE1PcpsCccwsrAmbiguousAcquisition, [694](#)
  - GalileoE1PcpsQuickSyncAmbiguousAcquisition, [700](#)
  - GalileoE1PcpsTongAmbiguousAcquisition, [706](#)
  - GalileoE1TcpConnectorTracking, [712](#)
  - GalileoE5aDIIPIITracking, [716](#)
  - GalileoE5aDIIPIITrackingFpga, [721](#)
  - GalileoE5aNoncoherentIQAcquisitionCaf, [725](#)
  - GalileoE5aPcpsAcquisition, [731](#)
  - GalileoE5aPcpsAcquisitionFpga, [738](#)
  - GalileoE5aTelemetryDecoder, [744](#)
  - GalileoE5bDIIPIITracking, [748](#)
  - GalileoE5bPcpsAcquisition, [753](#)
  - GalileoE5bPcpsAcquisitionFpga, [760](#)
  - GalileoE5bTelemetryDecoder, [766](#)
  - GalileoE6DIIPIITracking, [770](#)
  - GalileoE6PcpsAcquisition, [775](#)
  - GalileoE6TelemetryDecoder, [780](#)
  - GenSignalSource, [784](#)
  - GlionassL1CaDIIPIICAidTracking, [828](#)
  - GlionassL1CaDIIPIITracking, [832](#)
  - GlionassL1CaPcpsAcquisition, [836](#)
  - GlionassL1CaTelemetryDecoder, [841](#)
  - GlionassL2CaDIIPIICAidTracking, [845](#)
  - GlionassL2CaDIIPIITracking, [849](#)
  - GlionassL2CaPcpsAcquisition, [853](#)
  - GlionassL2CaTelemetryDecoder, [858](#)
  - GNSSBlockInterface, [909](#)
  - GNSSFlowgraph, [913](#)
  - GpsL1CaDIIPIITracking, [967](#)
  - GpsL1CaDIIPIITrackingFpga, [972](#)
  - GpsL1CaDIIPIITrackingGPU, [976](#)
  - GpsL1CaGaussianTracking, [980](#)
  - GpsL1CaKfTracking, [984](#)
  - GpsL1CaPcpsAcquisition, [989](#)
  - GpsL1CaPcpsAcquisitionFpga, [1000](#)
  - GpsL1CaPcpsAssistedAcquisition, [1006](#)
  - GpsL1CaPcpsOpenCIAcquisition, [1011](#)
  - GpsL1CaPcpsQuickSyncAcquisition, [1017](#)
  - GpsL1CaPcpsTongAcquisition, [1022](#)
  - GpsL1CaTcpConnectorTracking, [1027](#)
  - GpsL1CaTelemetryDecoder, [1031](#)
  - GpsL2CTelemetryDecoder, [1035](#)
  - GpsL2MDIIPIITracking, [1039](#)
  - GpsL2MDIIPIITrackingFpga, [1043](#)
  - GpsL2MPcpsAcquisition, [1048](#)
  - GpsL2MPcpsAcquisitionFpga, [1053](#)
  - GpsL5DIIPIITracking, [1058](#)
  - GpsL5DIIPIITrackingFpga, [1063](#)
  - GpsL5iPcpsAcquisition, [1068](#)
  - GpsL5iPcpsAcquisitionFpga, [1074](#)
  - GpsL5TelemetryDecoder, [1079](#)
  - HybridObservables, [1096](#)
  - IbyteToCbyte, [1100](#)
  - IbyteToComplex, [1103](#)
  - IbyteToCshort, [1106](#)
  - IONGSMSSignalSource, [1124](#)
  - IshortToComplex, [1128](#)
  - IshortToCshort, [1131](#)
  - LabsatSignalSource, [1145](#)
  - LimesdrSignalSource, [1155](#)
  - MAX2771EVKITSignalSourceFPGA, [1165](#)

- MmseResamplerConditioner, [1168](#)
- MultichannelFileSignalSource, [1183](#)
- NotchFilter, [1201](#)
- NotchFilterLite, [1204](#)
- OsmosdrSignalSource, [1225](#)
- Pass\_Through, [1240](#)
- PlutosdrSignalSource, [1291](#)
- PulseBlankingFilter, [1306](#)
- RawArraySignalSource, [1330](#)
- Rtklib\_Pvt, [1363](#)
- RtlTcpSignalSource, [1385](#)
- SbasL1TelemetryDecoder, [1393](#)
- SignalConditioner, [1427](#)
- SignalGenerator, [1431](#)
- SpirGSS6450FileSignalSource, [1456](#)
- UhdSignalSource, [1514](#)
- ZmqSignalSource, [1536](#)
- connected
  - GNSSFlowgraph, [914](#)
- CONST\_AMB
  - rtklib\_ppp.h, [1824](#)
- context
  - cl\_fft\_plan, [442](#)
- control\_thread.h, [2383](#)
- ControlThread, [465](#)
  - ~ControlThread, [466](#)
  - applied\_actions, [466](#)
  - ControlThread, [466](#)
  - flowgraph, [466](#)
  - me, [467](#)
  - processed\_control\_messages, [466](#)
  - run, [466](#)
  - set\_control\_queue, [467](#)
- cop
  - MACK\_header, [1156](#)
  - MACK\_tag\_info, [1162](#)
  - Tag, [1473](#)
- Core GNSS Receiver, [205](#)
- core\_libs, [207](#)
  - b\_io\_context, [209](#)
  - channel\_event\_sptr, [209](#)
  - channel\_status\_msg\_receiver\_sptr, [209](#)
  - command\_event\_sptr, [209](#)
  - find\_uio\_dev\_file\_name, [210](#)
  - galileo\_e6\_has\_msg\_receiver\_sptr, [209](#)
  - galileo\_tow\_map\_sptr, [209](#)
  - gnss\_sdr\_fpga\_sample\_counter\_sptr, [209](#)
  - gnss\_sdr\_sample\_counter\_sptr, [209](#)
  - gnss\_sdr\_time\_counter\_sptr, [209](#)
  - INI\_ALLOW\_MULTILINE, [209](#)
  - ini\_parse, [210](#)
  - nav\_message\_monitor\_sptr, [209](#)
  - osnma\_msg\_receiver\_sptr, [209](#)
- core\_monitor, [210](#)
  - b\_io\_context, [211](#)
  - gnss\_synchro\_monitor\_sptr, [211](#)
- core\_receiver, [211](#)
- core\_system\_parameters, [212](#)
- A0\_6\_LSB, [247](#)
- A1\_6\_LSB, [247](#)
- A\_0\_LSB, [247](#)
- A\_0G\_10\_LSB, [247](#)
- A\_1\_LSB, [247](#)
- A\_1\_LSB\_GAL, [247](#)
- A\_1G\_10\_LSB, [247](#)
- A\_F0\_LSB, [247](#)
- A\_F1\_LSB, [247](#)
- A\_F2\_LSB, [247](#)
- AF0\_10\_LSB, [247](#)
- AF0\_4\_LSB, [248](#)
- AF0\_8\_LSB, [248](#)
- AF0\_9\_LSB, [248](#)
- AF1\_10\_LSB, [248](#)
- AF1\_4\_LSB, [248](#)
- AF1\_8\_LSB, [248](#)
- AF1\_9\_LSB, [248](#)
- AF2\_4\_LSB, [248](#)
- AI0\_5\_LSB, [248](#)
- AI1\_5\_LSB, [248](#)
- AI2\_5\_LSB, [248](#)
- ALM\_AF0\_LSB, [248](#)
- ALM\_AF1\_LSB, [249](#)
- ALM\_DELTAI\_LSB, [249](#)
- ALM\_ECC\_LSB, [249](#)
- ALM\_MZERO\_LSB, [249](#)
- ALM\_OMEGA\_LSB, [249](#)
- ALM\_OMEGADOT\_LSB, [249](#)
- ALM\_OMEGAZERO\_LSB, [249](#)
- ALM\_SQUAREA\_LSB, [249](#)
- ALM\_TOA\_LSB, [249](#)
- ALPHA\_0, [246](#)
- ALPHA\_0\_LSB, [249](#)
- ALPHA\_1\_LSB, [249](#)
- ALPHA\_2\_LSB, [249](#)
- ALPHA\_3\_LSB, [250](#)
- AODO\_LSB, [250](#)
- AS2R, [250](#)
- AU, [250](#)
- BEIDOU\_B1I\_CODE\_LENGTH\_CHIPS, [250](#)
- BEIDOU\_B1I\_CODE\_PERIOD\_MS, [250](#)
- BEIDOU\_B1I\_CODE\_PERIOD\_S, [250](#)
- BEIDOU\_B1I\_CODE\_RATE\_CPS, [250](#)
- BEIDOU\_B1I\_D2\_SECONDARY\_CODE\_STR, [250](#)
- BEIDOU\_B1I\_FREQ\_HZ, [250](#)
- BEIDOU\_B1I\_GEO\_PREAMBLE\_LENGTH\_SYMBOLS, [250](#)
- BEIDOU\_B1I\_GEO\_PREAMBLE\_SYMBOLS\_STR, [251](#)
- BEIDOU\_B1I\_GEO\_TELEMETRY\_SYMBOLS\_PER\_BIT, [251](#)
- BEIDOU\_B1I\_PREAMBLE\_DURATION\_MS, [251](#)
- BEIDOU\_B1I\_PREAMBLE\_DURATION\_S, [251](#)
- BEIDOU\_B1I\_PREAMBLE\_LENGTH\_BITS, [251](#)
- BEIDOU\_B1I\_PREAMBLE\_LENGTH\_SYMBOLS, [251](#)

- BEIDOU\_B1I\_SECONDARY\_CODE\_LENGTH, [251](#)
- BEIDOU\_B1I\_SECONDARY\_CODE\_STR, [251](#)
- BEIDOU\_B1I\_TELEMETRY\_RATE\_BITS\_SECOND, [251](#)
- BEIDOU\_B1I\_TELEMETRY\_RATE\_SYMBOLS\_SECOND, [251](#)
- BEIDOU\_B1I\_TELEMETRY\_SYMBOL\_PERIOD\_MS, [251](#)
- BEIDOU\_B1I\_TELEMETRY\_SYMBOLS\_PER\_BIT, [251](#)
- BEIDOU\_B3I\_CODE\_LENGTH\_CHIPS, [252](#)
- BEIDOU\_B3I\_CODE\_PERIOD\_MS, [252](#)
- BEIDOU\_B3I\_CODE\_PERIOD\_S, [252](#)
- BEIDOU\_B3I\_CODE\_RATE\_CPS, [252](#)
- BEIDOU\_B3I\_D2\_SECONDARY\_CODE\_STR, [252](#)
- BEIDOU\_B3I\_FREQ\_HZ, [252](#)
- BEIDOU\_B3I\_GEO\_PREAMBLE\_LENGTH\_SYMBOLS, [252](#)
- BEIDOU\_B3I\_GEO\_PREAMBLE\_SYMBOLS\_STR, [252](#)
- BEIDOU\_B3I\_GEO\_TELEMETRY\_SYMBOLS\_PER\_BIT, [252](#)
- BEIDOU\_B3I\_PREAMBLE\_DURATION\_MS, [252](#)
- BEIDOU\_B3I\_PREAMBLE\_DURATION\_S, [252](#)
- BEIDOU\_B3I\_PREAMBLE\_LENGTH\_BITS, [253](#)
- BEIDOU\_B3I\_PREAMBLE\_LENGTH\_SYMBOLS, [253](#)
- BEIDOU\_B3I\_SECONDARY\_CODE\_LENGTH, [253](#)
- BEIDOU\_B3I\_SECONDARY\_CODE\_STR, [253](#)
- BEIDOU\_B3I\_TELEMETRY\_RATE\_BITS\_SECOND, [253](#)
- BEIDOU\_B3I\_TELEMETRY\_RATE\_SYMBOLS\_SECOND, [253](#)
- BEIDOU\_B3I\_TELEMETRY\_SYMBOL\_PERIOD\_MS, [253](#)
- BEIDOU\_B3I\_TELEMETRY\_SYMBOLS\_PER\_BIT, [253](#)
- BEIDOU\_DNAV\_BDT2GPST\_LEAP\_SEC\_OFFSET, [253](#)
- BEIDOU\_DNAV\_BDT2GPST\_WEEK\_NUM\_OFFSET, [253](#)
- BEIDOU\_DNAV\_PREAMBLE, [253](#)
- BEIDOU\_DNAV\_PREAMBLE\_LENGTH\_BITS, [254](#)
- BEIDOU\_DNAV\_PREAMBLE\_LENGTH\_SYMBOLS, [254](#)
- BEIDOU\_DNAV\_PREAMBLE\_PERIOD\_SYMBOLS, [254](#)
- BEIDOU\_DNAV\_SUBFRAME\_DATA\_BITS, [254](#)
- BEIDOU\_DNAV\_SUBFRAME\_SYMBOLS, [254](#)
- BEIDOU\_DNAV\_WORD\_LENGTH\_BITS, [254](#)
- BEIDOU\_DNAV\_WORDS\_SUBFRAME, [254](#)
- BEIDOU\_F, [254](#)
- BEIDOU\_GM, [254](#)
- BEIDOU\_OMEGA\_EARTH\_DOT, [254](#)
- BETA\_0\_LSB, [254](#)
- BETA\_1\_LSB, [255](#)
- BETA\_2\_LSB, [255](#)
- BETA\_3\_LSB, [255](#)
- BGD\_E1\_E5A\_5\_LSB, [255](#)
- BGD\_E1\_E5B\_5\_LSB, [255](#)
- BITS\_IN\_OCTET, [255](#)
- C\_IC\_4\_LSB, [255](#)
- C\_IC\_LSB, [255](#)
- C\_IS\_4\_LSB, [255](#)
- C\_IS\_LSB, [255](#)
- C\_RC\_3\_LSB, [255](#)
- C\_RC\_LSB, [255](#)
- C\_RS\_3\_LSB, [256](#)
- C\_RS\_LSB, [256](#)
- C\_UC\_3\_LSB, [256](#)
- C\_UC\_LSB, [256](#)
- C\_US\_3\_LSB, [256](#)
- C\_US\_LSB, [256](#)
- CED\_af0red\_LSB, [256](#)
- CED\_af1red\_LSB, [256](#)
- CED\_DeltaAred\_LSB, [256](#)
- CED\_DeltaI0red\_LSB, [256](#)
- CED\_exred\_LSB, [256](#)
- CED\_eyred\_LSB, [256](#)
- CED\_lambda0red\_LSB, [257](#)
- CED\_Omega0red\_LSB, [257](#)
- CNAV\_A0\_LSB, [257](#)
- CNAV\_A1\_LSB, [257](#)
- CNAV\_A2\_LSB, [257](#)
- CNAV\_A\_DOT\_LSB, [257](#)
- CNAV\_A\_REF, [257](#)
- CNAV\_AF0\_LSB, [257](#)
- CNAV\_AF1\_LSB, [257](#)
- CNAV\_AF2\_LSB, [257](#)
- CNAV\_ALPHA0\_LSB, [257](#)
- CNAV\_ALPHA1\_LSB, [257](#)
- CNAV\_ALPHA2\_LSB, [258](#)
- CNAV\_ALPHA3\_LSB, [258](#)
- CNAV\_BETA0\_LSB, [258](#)
- CNAV\_BETA1\_LSB, [258](#)
- CNAV\_BETA2\_LSB, [258](#)
- CNAV\_BETA3\_LSB, [258](#)
- CNAV\_CIC\_LSB, [258](#)
- CNAV\_CIS\_LSB, [258](#)
- CNAV\_CRC\_LSB, [258](#)
- CNAV\_CRS\_LSB, [258](#)
- CNAV\_CUC\_LSB, [258](#)
- CNAV\_CUS\_LSB, [258](#)
- CNAV\_DELTA\_A\_LSB, [259](#)
- CNAV\_DELTA\_N0\_DOT\_LSB, [259](#)
- CNAV\_DELTA\_N0\_LSB, [259](#)
- CNAV\_DELTA\_OMEGA\_DOT\_LSB, [259](#)
- CNAV\_DELTA\_TLS\_LSB, [259](#)
- CNAV\_DELTA\_TLSF\_LSB, [259](#)
- CNAV\_DN\_LSB, [259](#)
- CNAV\_E\_ECCENTRICITY\_LSB, [259](#)
- CNAV\_I0\_DOT\_LSB, [259](#)

CNAV\_I0\_LSB, 259  
CNAV\_ISCL1\_LSB, 259  
CNAV\_ISCL2\_LSB, 259  
CNAV\_ISCL5I\_LSB, 260  
CNAV\_ISCL5Q\_LSB, 260  
CNAV\_M0\_LSB, 260  
CNAV\_OMEGA0\_LSB, 260  
CNAV\_OMEGA\_DOT\_REF, 260  
CNAV\_OMEGA\_LSB, 260  
CNAV\_TGD\_LSB, 260  
CNAV\_TOC\_LSB, 260  
CNAV\_TOE1\_LSB, 260  
CNAV\_TOE2\_LSB, 260  
CNAV\_TOP1\_LSB, 260  
CNAV\_TOP2\_LSB, 260  
CNAV\_TOT\_LSB, 261  
CNAV\_TOW\_LSB, 261  
CNAV\_WN\_LSF\_LSB, 261  
CNAV\_WN\_OT\_LSB, 261  
CODE\_L1A, 261  
CODE\_L1B, 261  
CODE\_L1C, 261  
CODE\_L1E, 261  
CODE\_L1I, 261  
CODE\_L1L, 261  
CODE\_L1M, 261  
CODE\_L1N, 262  
CODE\_L1P, 262  
CODE\_L1Q, 262  
CODE\_L1S, 262  
CODE\_L1W, 262  
CODE\_L1X, 262  
CODE\_L1Y, 262  
CODE\_L1Z, 262  
CODE\_L2C, 262  
CODE\_L2D, 262  
CODE\_L2I, 263  
CODE\_L2L, 263  
CODE\_L2M, 263  
CODE\_L2N, 263  
CODE\_L2P, 263  
CODE\_L2Q, 263  
CODE\_L2S, 263  
CODE\_L2W, 263  
CODE\_L2X, 263  
CODE\_L2Y, 263  
CODE\_L3I, 264  
CODE\_L3Q, 264  
CODE\_L3X, 264  
CODE\_L5A, 264  
CODE\_L5B, 264  
CODE\_L5C, 264  
CODE\_L5I, 264  
CODE\_L5Q, 264  
CODE\_L5X, 264  
CODE\_L6A, 264  
CODE\_L6B, 265  
CODE\_L6C, 265  
CODE\_L6I, 265  
CODE\_L6L, 265  
CODE\_L6Q, 265  
CODE\_L6S, 265  
CODE\_L6X, 265  
CODE\_L6Z, 265  
CODE\_L7I, 265  
CODE\_L7Q, 265  
CODE\_L7X, 266  
CODE\_L8I, 266  
CODE\_L8Q, 266  
CODE\_L8X, 266  
CODE\_L9A, 266  
CODE\_L9B, 266  
CODE\_L9C, 266  
CODE\_L9X, 266  
CODE\_NONE, 266  
D1\_A0\_ALMANAC\_LSB, 266  
D1\_A0\_LSB, 267  
D1\_A0GAL\_LSB, 267  
D1\_A0GLO\_LSB, 267  
D1\_A0GPS\_LSB, 267  
D1\_A0UTC\_LSB, 267  
D1\_A1\_ALMANAC\_LSB, 267  
D1\_A1\_LSB, 267  
D1\_A1GAL\_LSB, 267  
D1\_A1GLO\_LSB, 267  
D1\_A1GPS\_LSB, 267  
D1\_A1UTC\_LSB, 267  
D1\_A2\_LSB, 267  
D1\_ALPHA0\_LSB, 268  
D1\_ALPHA1\_LSB, 268  
D1\_ALPHA2\_LSB, 268  
D1\_ALPHA3\_LSB, 268  
D1\_BETA0\_LSB, 268  
D1\_BETA1\_LSB, 268  
D1\_BETA2\_LSB, 268  
D1\_BETA3\_LSB, 268  
D1\_CIC\_LSB, 268  
D1\_CIS\_LSB, 268  
D1\_CRC\_LSB, 268  
D1\_CRS\_LSB, 268  
D1\_CUC\_LSB, 269  
D1\_CUS\_LSB, 269  
D1\_DELTA\_I\_LSB, 269  
D1\_DELTA\_N\_LSB, 269  
D1\_E\_ALMANAC\_LSB, 269  
D1\_E\_LSB, 269  
D1\_I0\_LSB, 269  
D1\_IDOT\_LSB, 269  
D1\_M0\_ALMANAC\_LSB, 269  
D1\_M0\_LSB, 269  
D1\_OMEGA0\_ALMANAC\_LSB, 269  
D1\_OMEGA0\_LSB, 269  
D1\_OMEGA\_ALMANAC\_LSB, 270  
D1\_OMEGA\_DOT\_ALMANAC\_LSB, 270  
D1\_OMEGA\_DOT\_LSB, 270  
D1\_OMEGA\_LSB, 270



- D1\_SQRT\_A\_ALMANAC\_LSB, [270](#)
- D1\_SQRT\_A\_LSB, [270](#)
- D1\_TGD1\_LSB, [270](#)
- D1\_TGD2\_LSB, [270](#)
- D1\_TOA\_LSB, [270](#)
- D1\_TOC\_LSB, [270](#)
- D1\_TOE\_LSB, [270](#)
- D2R, [270](#)
- DELTA\_A\_7\_LSB, [271](#)
- DELTA\_A\_8\_LSB, [271](#)
- DELTA\_A\_9\_LSB, [271](#)
- DELTA\_I\_7\_LSB, [271](#)
- DELTA\_I\_8\_LSB, [271](#)
- DELTA\_I\_9\_LSB, [271](#)
- DELTA\_N\_3\_LSB, [271](#)
- DELTA\_N\_LSB, [271](#)
- DELTAT\_LS\_LSB, [271](#)
- DELTAT\_LSF\_LSB, [271](#)
- DFRQ1\_GLO, [271](#)
- DFRQ2\_GLO, [271](#)
- DISPLAY\_COLORS, [246](#)
- DN\_LSB, [272](#)
- E\_1\_LSB, [272](#)
- E\_7\_LSB, [272](#)
- E\_8\_LSB, [272](#)
- E\_9\_LSB, [272](#)
- ECCENTRICITY\_LSB, [272](#)
- FIRST\_RS\_BIT, [272](#)
- FIRST\_RS\_BIT\_AFTER\_IODNAV, [272](#)
- FNAV\_A0\_4\_LSB, [272](#)
- FNAV\_A0G\_4\_LSB, [272](#)
- FNAV\_A12\_2\_LSB, [272](#)
- FNAV\_A1\_4\_LSB, [272](#)
- FNAV\_A1G\_4\_LSB, [273](#)
- FNAV\_AF0\_1\_LSB, [273](#)
- FNAV\_AF0\_5\_LSB, [273](#)
- FNAV\_AF1\_1\_LSB, [273](#)
- FNAV\_AF1\_5\_LSB, [273](#)
- FNAV\_AF2\_1\_LSB, [273](#)
- FNAV\_AI0\_1\_LSB, [273](#)
- FNAV\_AI1\_1\_LSB, [273](#)
- FNAV\_AI2\_1\_LSB, [273](#)
- FNAV\_BGD\_1\_LSB, [273](#)
- FNAV\_CIC\_4\_LSB, [273](#)
- FNAV\_CIS\_4\_LSB, [273](#)
- FNAV\_CRC\_3\_LSB, [274](#)
- FNAV\_CRS\_3\_LSB, [274](#)
- FNAV\_CUC\_3\_LSB, [274](#)
- FNAV\_CUS\_3\_LSB, [274](#)
- FNAV\_DELTAA12\_5\_LSB, [274](#)
- FNAV\_DELTAI\_5\_LSB, [274](#)
- FNAV\_DELTAN\_3\_LSB, [274](#)
- FNAV\_E\_2\_LSB, [274](#)
- FNAV\_E\_5\_LSB, [274](#)
- FNAV\_I0\_3\_LSB, [274](#)
- FNAV\_IDOT\_2\_LSB, [274](#)
- FNAV\_M0\_2\_LSB, [274](#)
- FNAV\_M0\_5\_LSB, [275](#)
- FNAV\_OMEGA0\_2\_LSB, [275](#)
- FNAV\_OMEGA0\_5\_LSB, [275](#)
- FNAV\_OMEGADOT\_2\_LSB, [275](#)
- FNAV\_OMEGADOT\_5\_LSB, [275](#)
- FNAV\_T0A\_5\_LSB, [275](#)
- FNAV\_T0C\_1\_LSB, [275](#)
- FNAV\_T0E\_3\_LSB, [275](#)
- FNAV\_T0G\_4\_LSB, [275](#)
- FNAV\_T0T\_4\_LSB, [275](#)
- FNAV\_W\_3\_LSB, [275](#)
- FNAV\_W\_5\_LSB, [275](#)
- FREQ1, [276](#)
- FREQ1\_BDS, [276](#)
- FREQ1\_GLO, [276](#)
- FREQ2, [276](#)
- FREQ2\_BDS, [276](#)
- FREQ2\_GLO, [276](#)
- FREQ3\_BDS, [276](#)
- FREQ3\_GLO, [276](#)
- FREQ5, [276](#)
- FREQ6, [276](#)
- FREQ7, [277](#)
- FREQ8, [277](#)
- FREQ9, [277](#)
- GALILEO\_CNAV\_BITS\_FOR\_CRC, [277](#)
- GALILEO\_CNAV\_BYTES\_FOR\_CRC, [277](#)
- GALILEO\_CNAV\_CRC\_LENGTH, [277](#)
- GALILEO\_CNAV\_HAS\_PAGE\_DATA\_BITS, [277](#)
- GALILEO\_CNAV\_INFORMATION\_VECTOR\_LENGTH, [277](#)
- GALILEO\_CNAV\_INTERLEAVER\_COLS, [277](#)
- GALILEO\_CNAV\_INTERLEAVER\_ROWS, [277](#)
- GALILEO\_CNAV\_MAX\_NUMBER\_SYMBOLS\_ENCODED\_BLOCK, [277](#)
- GALILEO\_CNAV\_MESSAGE\_BITS\_PER\_PAGE, [278](#)
- GALILEO\_CNAV\_MT1\_HEADER\_BITS, [278](#)
- GALILEO\_CNAV\_OCTETS\_IN\_SUBPAGE, [278](#)
- GALILEO\_CNAV\_PAGE\_HEADER\_BITS, [278](#)
- GALILEO\_CNAV\_PAGE\_MS, [278](#)
- GALILEO\_CNAV\_PAGE\_RESERVED\_BITS, [278](#)
- GALILEO\_CNAV\_PREAMBLE, [278](#)
- GALILEO\_CNAV\_PREAMBLE\_LENGTH\_BITS, [278](#)
- GALILEO\_CNAV\_PREAMBLE\_PERIOD\_SYMBOLS, [278](#)
- GALILEO\_CNAV\_SYMBOLS\_PER\_PAGE, [278](#)
- GALILEO\_CNAV\_TELEMETRY\_RATE\_BITS\_SECOND, [278](#)
- GALILEO\_DATA\_FRAME\_BITS, [278](#)
- GALILEO\_DATA\_FRAME\_BYTES, [279](#)
- GALILEO\_DATA\_JK\_BITS, [279](#)
- GALILEO\_E1\_B\_CODE\_LENGTH\_CHIPS, [279](#)
- GALILEO\_E1\_B\_PRIMARY\_CODE, [279](#)
- GALILEO\_E1\_B\_PRIMARY\_CODE\_STR\_LENGTH, [279](#)
- GALILEO\_E1\_B\_SAMPLES\_PER\_SYMBOL, [279](#)
- GALILEO\_E1\_B\_SYMBOL\_RATE\_BPS, [279](#)



- GALILEO\_E1\_C\_PRIMARY\_CODE, [279](#)  
GALILEO\_E1\_C\_PRIMARY\_CODE\_STR\_LENGTH, [279](#)  
GALILEO\_E1\_C\_SECONDARY\_CODE, [279](#)  
GALILEO\_E1\_C\_SECONDARY\_CODE\_LENGTH, [279](#)  
GALILEO\_E1\_CODE\_CHIP\_RATE\_CPS, [280](#)  
GALILEO\_E1\_CODE\_PERIOD\_MS, [280](#)  
GALILEO\_E1\_CODE\_PERIOD\_S, [280](#)  
GALILEO\_E1\_FREQ\_HZ, [280](#)  
GALILEO\_E1\_HISTORY\_DEEP, [280](#)  
GALILEO\_E1\_NUMBER\_OF\_CODES, [280](#)  
GALILEO\_E1\_OPT\_ACQ\_FS\_SPS, [280](#)  
GALILEO\_E1\_SUB\_CARRIER\_A\_RATE\_HZ, [280](#)  
GALILEO\_E1\_SUB\_CARRIER\_B\_RATE\_HZ, [280](#)  
GALILEO\_E5A\_CODE\_CHIP\_RATE\_CPS, [280](#)  
GALILEO\_E5A\_CODE\_LENGTH\_CHIPS, [281](#)  
GALILEO\_E5A\_CODE\_PERIOD\_MS, [281](#)  
GALILEO\_E5A\_CODE\_PERIOD\_S, [281](#)  
GALILEO\_E5A\_CRC\_ERROR\_LIMIT, [281](#)  
GALILEO\_E5A\_FREQ\_HZ, [281](#)  
GALILEO\_E5A\_HISTORY\_DEEP, [281](#)  
GALILEO\_E5A\_I\_PRIMARY\_CODE, [281](#)  
GALILEO\_E5A\_I\_PRIMARY\_CODE\_STR\_LENGTH, [281](#)  
GALILEO\_E5A\_I\_SECONDARY\_CODE, [281](#)  
GALILEO\_E5A\_I\_SECONDARY\_CODE\_LENGTH, [281](#)  
GALILEO\_E5A\_I\_TIERED\_CODE\_PERIOD\_S, [281](#)  
GALILEO\_E5A\_NUMBER\_OF\_CODES, [282](#)  
GALILEO\_E5A\_OPT\_ACQ\_FS\_SPS, [282](#)  
GALILEO\_E5A\_Q\_PRIMARY\_CODE, [282](#)  
GALILEO\_E5A\_Q\_PRIMARY\_CODE\_STR\_LENGTH, [282](#)  
GALILEO\_E5A\_Q\_SECONDARY\_CODE, [282](#)  
GALILEO\_E5A\_Q\_SECONDARY\_CODE\_LENGTH, [282](#)  
GALILEO\_E5A\_Q\_SECONDARY\_CODE\_STR\_LENGTH, [282](#)  
GALILEO\_E5A\_Q\_TIERED\_CODE\_PERIOD\_S, [282](#)  
GALILEO\_E5A\_SYMBOL\_RATE\_BPS, [282](#)  
GALILEO\_E5B\_CODE\_CHIP\_RATE\_CPS, [282](#)  
GALILEO\_E5B\_CODE\_LENGTH\_CHIPS, [282](#)  
GALILEO\_E5B\_CODE\_PERIOD\_MS, [283](#)  
GALILEO\_E5B\_CODE\_PERIOD\_S, [283](#)  
GALILEO\_E5B\_FREQ\_HZ, [283](#)  
GALILEO\_E5B\_HISTORY\_DEEP, [283](#)  
GALILEO\_E5B\_I\_PRIMARY\_CODE, [283](#)  
GALILEO\_E5B\_I\_PRIMARY\_CODE\_STR\_LENGTH, [283](#)  
GALILEO\_E5B\_I\_SECONDARY\_CODE, [283](#)  
GALILEO\_E5B\_I\_SECONDARY\_CODE\_LENGTH, [283](#)  
GALILEO\_E5B\_I\_TIERED\_CODE\_PERIOD\_S, [283](#)  
GALILEO\_E5B\_NUMBER\_OF\_CODES, [283](#)  
GALILEO\_E5B\_OPT\_ACQ\_FS\_SPS, [283](#)  
GALILEO\_E5B\_Q\_PRIMARY\_CODE, [284](#)  
GALILEO\_E5B\_Q\_PRIMARY\_CODE\_STR\_LENGTH, [284](#)  
GALILEO\_E5B\_Q\_SECONDARY\_CODE, [284](#)  
GALILEO\_E5B\_Q\_SECONDARY\_CODE\_LENGTH, [284](#)  
GALILEO\_E5B\_Q\_SECONDARY\_CODE\_STR\_LENGTH, [284](#)  
GALILEO\_E5B\_Q\_TIERED\_CODE\_PERIOD\_S, [284](#)  
GALILEO\_E5B\_SYMBOL\_RATE\_BPS, [284](#)  
GALILEO\_E6\_B\_CODE\_CHIP\_RATE\_CPS, [284](#)  
GALILEO\_E6\_B\_CODE\_LENGTH\_CHIPS, [284](#)  
GALILEO\_E6\_B\_PRIMARY\_CODE, [284](#)  
GALILEO\_E6\_B\_PRIMARY\_CODE\_STR\_LENGTH, [284](#)  
GALILEO\_E6\_C\_CODE\_CHIP\_RATE\_CPS, [285](#)  
GALILEO\_E6\_C\_CODE\_LENGTH\_CHIPS, [285](#)  
GALILEO\_E6\_C\_PRIMARY\_CODE, [285](#)  
GALILEO\_E6\_C\_PRIMARY\_CODE\_STR\_LENGTH, [285](#)  
GALILEO\_E6\_C\_SECONDARY\_CODE, [285](#)  
GALILEO\_E6\_C\_SECONDARY\_CODE\_LENGTH\_CHIPS, [285](#)  
GALILEO\_E6\_C\_SECONDARY\_CODE\_STR\_LENGTH, [285](#)  
GALILEO\_E6\_CODE\_PERIOD\_MS, [285](#)  
GALILEO\_E6\_CODE\_PERIOD\_S, [285](#)  
GALILEO\_E6\_FREQ\_HZ, [285](#)  
GALILEO\_E6\_NUMBER\_OF\_CODES, [285](#)  
GALILEO\_E6\_OPT\_ACQ\_FS\_SPS, [286](#)  
GALILEO\_F, [286](#)  
GALILEO\_FNAV\_CODES\_PER\_PAGE, [286](#)  
GALILEO\_FNAV\_CODES\_PER\_PREAMBLE, [286](#)  
GALILEO\_FNAV\_CODES\_PER\_SYMBOL, [286](#)  
GALILEO\_FNAV\_DATA\_FRAME\_BITS, [286](#)  
GALILEO\_FNAV\_DATA\_FRAME\_BYTES, [286](#)  
GALILEO\_FNAV\_INTERLEAVER\_COLS, [286](#)  
GALILEO\_FNAV\_INTERLEAVER\_ROWS, [286](#)  
GALILEO\_FNAV\_PAGE\_TYPE\_BITS, [286](#)  
GALILEO\_FNAV\_PREAMBLE, [286](#)  
GALILEO\_FNAV\_PREAMBLE\_LENGTH\_BITS, [286](#)  
GALILEO\_FNAV\_SECONDS\_PER\_PAGE, [287](#)  
GALILEO\_FNAV\_SYMBOLS\_PER\_PAGE, [287](#)  
GALILEO\_GM, [287](#)  
GALILEO\_INAV\_ENCODED\_SSP1, [287](#)  
GALILEO\_INAV\_ENCODED\_SSP2, [287](#)  
GALILEO\_INAV\_ENCODED\_SSP3, [287](#)  
GALILEO\_INAV\_INTERLEAVER\_COLS, [287](#)  
GALILEO\_INAV\_INTERLEAVER\_ROWS, [287](#)  
GALILEO\_INAV\_PAGE\_PART\_MS, [287](#)  
GALILEO\_INAV\_PAGE\_PART\_SECONDS, [287](#)  
GALILEO\_INAV\_PAGE\_PART\_SYMBOLS, [287](#)  
GALILEO\_INAV\_PAGE\_PART\_WITH\_PREAMBLE\_SECONDS, [288](#)  
GALILEO\_INAV\_PAGE\_SECONDS, [288](#)

- GALILEO\_INAV\_PAGE\_SYMBOLS, [288](#)
- GALILEO\_INAV\_PLAIN\_SSP1, [288](#)
- GALILEO\_INAV\_PLAIN\_SSP2, [288](#)
- GALILEO\_INAV\_PLAIN\_SSP3, [288](#)
- GALILEO\_INAV\_PREAMBLE, [288](#)
- GALILEO\_INAV\_PREAMBLE\_LENGTH\_BITS, [288](#)
- GALILEO\_INAV\_PREAMBLE\_PERIOD\_SYMBOLS, [288](#)
- GALILEO\_ISM\_CRC\_DATA\_BITS, [288](#)
- GALILEO\_ISM\_CRC\_DATA\_BYTES, [288](#)
- GALILEO\_PAGE\_TYPE\_BITS, [289](#)
- GALILEO\_TELEMETRY\_RATE\_BITS\_SECOND, [289](#)
- GLONASS\_C20, [289](#)
- GLONASS\_CA\_NBR\_SATS, [289](#)
- GLONASS\_EARTH\_INCLINATION, [289](#)
- GLONASS\_EARTH\_RADIUS, [289](#)
- GLONASS\_F\_M\_A, [289](#)
- GLONASS\_FLATTENING, [289](#)
- GLONASS\_GM, [289](#)
- GLONASS\_GNAV\_CRC\_I\_INDEX, [289](#)
- GLONASS\_GNAV\_CRC\_J\_INDEX, [290](#)
- GLONASS\_GNAV\_CRC\_K\_INDEX, [290](#)
- GLONASS\_GNAV\_CRC\_L\_INDEX, [290](#)
- GLONASS\_GNAV\_CRC\_M\_INDEX, [290](#)
- GLONASS\_GNAV\_CRC\_N\_INDEX, [290](#)
- GLONASS\_GNAV\_CRC\_P\_INDEX, [290](#)
- GLONASS\_GNAV\_CRC\_Q\_INDEX, [290](#)
- GLONASS\_GNAV\_DATA\_SYMBOLS, [290](#)
- GLONASS\_GNAV\_ECC\_LOCATOR, [290](#)
- GLONASS\_GNAV\_HAMMING\_CODE\_BITS, [291](#)
- GLONASS\_GNAV\_PREAMBLE, [246](#)
- GLONASS\_GNAV\_PREAMBLE\_DURATION\_S, [291](#)
- GLONASS\_GNAV\_PREAMBLE\_LENGTH\_BITS, [291](#)
- GLONASS\_GNAV\_PREAMBLE\_LENGTH\_SYMBOLS, [291](#)
- GLONASS\_GNAV\_PREAMBLE\_PERIOD\_SYMBOLS, [291](#)
- GLONASS\_GNAV\_STRING\_BITS, [291](#)
- GLONASS\_GNAV\_STRING\_SYMBOLS, [291](#)
- GLONASS\_GNAV\_TELEMETRY\_RATE\_BITS\_SECOND, [291](#)
- GLONASS\_GNAV\_TELEMETRY\_RATE\_SYMBOLS\_SECOND, [291](#)
- GLONASS\_GNAV\_TELEMETRY\_SYMBOLS\_PER\_BIT, [291](#)
- GLONASS\_GNAV\_TELEMETRY\_SYMBOLS\_PER\_PREAMBLE, [292](#)
- GLONASS\_GRAVITY, [292](#)
- GLONASS\_GRAVITY\_CORRECTION, [292](#)
- GLONASS\_J2, [292](#)
- GLONASS\_J4, [292](#)
- GLONASS\_J6, [292](#)
- GLONASS\_J8, [292](#)
- GLONASS\_L1\_CA\_CHIP\_PERIOD\_S, [292](#)
- GLONASS\_L1\_CA\_CODE\_LENGTH\_CHIPS, [292](#)
- GLONASS\_L1\_CA\_CODE\_PERIOD\_S, [292](#)
- GLONASS\_L1\_CA\_CODE\_RATE\_CPS, [293](#)
- GLONASS\_L1\_CA\_DFREQ\_HZ, [293](#)
- GLONASS\_L1\_CA\_FREQ\_HZ, [293](#)
- GLONASS\_L1\_CA\_HISTORY\_DEEP, [293](#)
- GLONASS\_L1\_CA\_SYMBOL\_RATE\_BPS, [293](#)
- GLONASS\_L2\_CA\_CHIP\_PERIOD\_S, [293](#)
- GLONASS\_L2\_CA\_CODE\_LENGTH\_CHIPS, [293](#)
- GLONASS\_L2\_CA\_CODE\_PERIOD\_S, [293](#)
- GLONASS\_L2\_CA\_CODE\_RATE\_CPS, [293](#)
- GLONASS\_L2\_CA\_DFREQ\_HZ, [293](#)
- GLONASS\_L2\_CA\_FREQ\_HZ, [294](#)
- GLONASS\_L2\_CA\_SYMBOL\_RATE\_BPS, [294](#)
- GLONASS\_LEAP\_SECONDS, [294](#)
- GLONASS\_MOON\_ECCENTRICITY, [294](#)
- GLONASS\_MOON\_GM, [294](#)
- GLONASS\_MOON\_INCLINATION, [294](#)
- GLONASS\_MOON\_OMEGA\_0, [294](#)
- GLONASS\_MOON\_OMEGA\_1, [295](#)
- GLONASS\_MOON\_Q0, [295](#)
- GLONASS\_MOON\_Q1, [295](#)
- GLONASS\_MOON\_SEMI\_MAJOR\_AXIS, [295](#)
- GLONASS\_OMEGA\_EARTH\_DOT, [295](#)
- GLONASS\_PRN, [295](#)
- GLONASS\_SEMI\_MAJOR\_AXIS, [295](#)
- GLONASS\_SUN\_ECCENTRICITY, [295](#)
- GLONASS\_SUN\_GM, [295](#)
- GLONASS\_SUN\_OMEGA, [295](#)
- GLONASS\_SUN\_Q0, [296](#)
- GLONASS\_SUN\_Q1, [296](#)
- GLONASS\_SUN\_SEMI\_MAJOR\_AXIS, [296](#)
- GLONASS\_TAU\_0, [296](#)
- GLONASS\_TAU\_1, [296](#)
- GLONASS\_U0, [296](#)
- GNSS\_OMEGA\_EARTH\_DOT, [296](#)
- GNSS\_PI, [296](#)
- GPS\_CA\_PREAMBLE, [296](#)
- GPS\_CA\_PREAMBLE\_DURATION\_MS, [296](#)
- GPS\_CA\_PREAMBLE\_DURATION\_S, [297](#)
- GPS\_CA\_PREAMBLE\_LENGTH\_BITS, [297](#)
- GPS\_CA\_PREAMBLE\_LENGTH\_SYMBOLS, [297](#)
- GPS\_CA\_PREAMBLE\_SYMBOLS\_STR, [297](#)
- GPS\_CA\_TELEMETRY\_RATE\_BITS\_SECOND, [297](#)
- GPS\_CA\_TELEMETRY\_RATE\_SYMBOLS\_SECOND, [297](#)
- GPS\_CA\_TELEMETRY\_SYMBOLS\_PER\_BIT, [297](#)
- GPS\_CA\_TELEMETRY\_SYMBOLS\_PER\_PREAMBLE, [297](#)
- GPS\_CA\_TELEMETRY\_SYMBOLS\_PER\_SECOND, [297](#)
- GPS\_F, [297](#)
- GPS\_GM, [297](#)
- GPS\_L1\_CA\_BIT\_PERIOD\_MS, [297](#)
- GPS\_L1\_CA\_CHIP\_PERIOD\_S, [298](#)
- GPS\_L1\_CA\_CODE\_LENGTH\_CHIPS, [298](#)
- GPS\_L1\_CA\_CODE\_PERIOD\_MS, [298](#)
- GPS\_L1\_CA\_CODE\_PERIOD\_S, [298](#)
- GPS\_L1\_CA\_CODE\_RATE\_CPS, [298](#)

- GPS\_L1\_CA\_HISTORY\_DEEP, 298
- GPS\_L1\_CA\_OPT\_ACQ\_FS\_SPS, 298
- GPS\_L1\_FREQ\_HZ, 298
- GPS\_L2\_CNAV\_DATA\_PAGE\_BITS, 298
- GPS\_L2\_CNAV\_DATA\_PAGE\_DURATION\_S, 298
- GPS\_L2\_CNAV\_DATA\_PAGE\_SYMBOLS, 299
- GPS\_L2\_FREQ\_HZ, 299
- GPS\_L2\_L\_CODE\_LENGTH\_CHIPS, 299
- GPS\_L2\_L\_CODE\_RATE\_CPS, 299
- GPS\_L2\_L\_PERIOD\_S, 299
- GPS\_L2\_M\_CODE\_LENGTH\_CHIPS, 299
- GPS\_L2\_M\_CODE\_RATE\_CPS, 299
- GPS\_L2\_M\_PERIOD\_S, 299
- GPS\_L2\_SAMPLES\_PER\_SYMBOL, 299
- GPS\_L2\_SYMBOLS\_PER\_BIT, 299
- GPS\_L2C\_HISTORY\_DEEP, 299
- GPS\_L2C\_M\_INIT\_REG, 300
- GPS\_L2C\_OPT\_ACQ\_FS\_SPS, 300
- GPS\_L5\_CNAV\_DATA\_PAGE\_BITS, 300
- GPS\_L5\_CNAV\_DATA\_PAGE\_DURATION\_S, 300
- GPS\_L5\_CNAV\_DATA\_PAGE\_SYMBOLS, 300
- GPS\_L5\_FREQ\_HZ, 300
- GPS\_L5\_HISTORY\_DEEP, 300
- GPS\_L5\_OPT\_ACQ\_FS\_SPS, 301
- GPS\_L5\_SAMPLES\_PER\_SYMBOL, 301
- GPS\_L5\_SYMBOLS\_PER\_BIT, 301
- GPS\_L5I\_CODE\_LENGTH\_CHIPS, 301
- GPS\_L5I\_CODE\_RATE\_CPS, 301
- GPS\_L5I\_INIT\_REG, 301
- GPS\_L5I\_NH\_CODE, 301
- GPS\_L5I\_NH\_CODE\_LENGTH, 301
- GPS\_L5I\_NH\_CODE\_STR, 301
- GPS\_L5I\_PERIOD\_MS, 301
- GPS\_L5I\_PERIOD\_S, 301
- GPS\_L5I\_SYMBOL\_PERIOD\_MS, 302
- GPS\_L5I\_SYMBOL\_PERIOD\_S, 302
- GPS\_L5Q\_CODE\_LENGTH\_CHIPS, 302
- GPS\_L5Q\_CODE\_RATE\_CPS, 302
- GPS\_L5Q\_INIT\_REG, 302
- GPS\_L5Q\_NH\_CODE, 302
- GPS\_L5Q\_NH\_CODE\_LENGTH, 302
- GPS\_L5Q\_NH\_CODE\_STR, 302
- GPS\_L5Q\_PERIOD\_S, 302
- GPS\_SUBFRAME\_BITS, 302
- GPS\_SUBFRAME\_LENGTH, 303
- GPS\_SUBFRAME\_MS, 303
- GPS\_SUBFRAME\_SECONDS, 303
- GPS\_WORD\_BITS, 303
- GPS\_WORD\_LENGTH, 303
- HALF\_PI, 303
- HAS\_MSG\_CODE\_BIAS\_LENGTH, 303
- HAS\_MSG\_CODE\_BIAS\_SCALE\_FACTOR, 303
- HAS\_MSG\_DELTA\_CLOCK\_CORRECTION\_LENGTH, 303
- HAS\_MSG\_DELTA\_CLOCK\_CORRECTION\_SUBSET\_LENGTH, 303
- HAS\_MSG\_DELTA\_CLOCK\_MULTIPLIER\_LENGTH, 303
- HAS\_MSG\_DELTA\_CLOCK\_MULTIPLIER\_SUBSET\_LENGTH, 304
- HAS\_MSG\_DELTA\_CLOCK\_SCALE\_FACTOR, 304
- HAS\_MSG\_DELTA\_CROSS\_TRACK\_LENGTH, 304
- HAS\_MSG\_DELTA\_CROSS\_TRACK\_SCALE\_FACTOR, 304
- HAS\_MSG\_DELTA\_IN\_TRACK\_LENGTH, 304
- HAS\_MSG\_DELTA\_IN\_TRACK\_SCALE\_FACTOR, 304
- HAS\_MSG\_DELTA\_RADIAL\_LENGTH, 304
- HAS\_MSG\_DELTA\_RADIAL\_SCALE\_FACTOR, 304
- HAS\_MSG\_GALILEO\_SYSTEM, 304
- HAS\_MSG\_GPS\_SYSTEM, 304
- HAS\_MSG\_ID\_CLOCK\_SUBSET\_LENGTH, 304
- HAS\_MSG\_ID\_MASK\_LENGTH, 304
- HAS\_MSG\_IOD\_GAL\_LENGTH, 305
- HAS\_MSG\_IOD\_GPS\_LENGTH, 305
- HAS\_MSG\_MASK\_RESERVED\_LENGTH, 305
- HAS\_MSG\_NAV\_MESSAGE\_LENGTH, 305
- HAS\_MSG\_NSYS\_LENGTH, 305
- HAS\_MSG\_NSYS\_SUB\_LENGTH, 305
- HAS\_MSG\_NUMBER\_GNSS\_IDS, 305
- HAS\_MSG\_NUMBER\_MASK\_IDS, 305
- HAS\_MSG\_NUMBER\_MAX\_TOH, 305
- HAS\_MSG\_NUMBER\_MESSAGE\_IDS, 305
- HAS\_MSG\_NUMBER\_SATELLITE\_IDS, 305
- HAS\_MSG\_NUMBER\_SIGNAL\_MASKS, 305
- HAS\_MSG\_PHASE\_BIAS\_LENGTH, 306
- HAS\_MSG\_PHASE\_BIAS\_SCALE\_FACTOR, 306
- HAS\_MSG\_PHASE\_DISCONTINUITY\_INDICATOR\_LENGTH, 306
- HAS\_MSG\_SATELLITE\_MASK\_LENGTH, 306
- HAS\_MSG\_SIGNAL\_MASK\_LENGTH, 306
- HAS\_MSG\_VALIDITY\_INDEX\_LENGTH, 306
- HAS\_MSG\_WRONG\_SYSTEM, 306
- HAS\_SIGNAL\_INDEX\_TABLE, 306
- HAS\_VALIDITY\_INTERVALS, 306
- I\_0\_2\_LSB, 307
- I\_0\_LSB, 307
- I\_DOT\_2\_LSB, 307
- I\_DOT\_LSB, 307
- INAV\_RS\_BUFFER\_LENGTH, 307
- INAV\_RS\_INFO\_VECTOR\_LENGTH, 307
- INAV\_RS\_PARITY\_VECTOR\_LENGTH, 307
- INAV\_RS\_SUBVECTOR\_LENGTH, 307
- M0\_10\_LSB, 307
- M0\_1\_LSB, 307
- M0\_7\_LSB, 307
- M0\_9\_LSB, 307
- M\_0\_LSB, 308
- MAX\_SECONDS\_REMEMBERING\_MID, 308
- MSG\_HTOA\_DELAY\_MS, 308
- MAXCODE, 308
- OMEGA0\_10\_LSB, 308
- OMEGA0\_7\_LSB, 308

OMEGA0\_8\_LSB, 308  
OMEGA\_0\_2\_LSB, 308  
OMEGA\_0\_LSB, 308  
OMEGA\_2\_LSB, 308  
OMEGA\_7\_LSB, 308  
OMEGA\_8\_LSB, 309  
OMEGA\_9\_LSB, 309  
OMEGA\_DOT\_10\_LSB, 309  
OMEGA\_DOT\_3\_LSB, 309  
OMEGA\_DOT\_7\_LSB, 309  
OMEGA\_DOT\_8\_LSB, 309  
OMEGA\_DOT\_LSB, 309  
OMEGA\_LSB, 309  
OSNMA\_TABLE\_1, 309  
OSNMA\_TABLE\_10, 309  
OSNMA\_TABLE\_11, 310  
OSNMA\_TABLE\_15, 310  
OSNMA\_TABLE\_16, 310  
OSNMA\_TABLE\_2, 310  
OSNMA\_TABLE\_3, 311  
OSNMA\_TABLE\_5, 311  
OSNMA\_TABLE\_6, 311  
OSNMA\_TABLE\_7, 311  
OSNMA\_TABLE\_8, 312  
PI\_TWO\_N19, 312  
PI\_TWO\_N23, 312  
PI\_TWO\_N31, 312  
PI\_TWO\_N38, 312  
PI\_TWO\_N43, 312  
R2D, 312  
SC2RAD, 312  
SIGNAL\_FREQ\_MAP, 313  
SIZE\_DSM\_BLOCKS\_BYTES, 313  
SPEED\_OF\_LIGHT\_M\_MS, 313  
SPEED\_OF\_LIGHT\_M\_S, 313  
SQRT\_A\_LSB, 313  
T0A\_7\_LSB, 313  
T0A\_9\_LSB, 313  
T0C\_4\_LSB, 313  
T0E\_1\_LSB, 313  
T0T\_6\_LSB, 314  
T\_0\_G\_10\_LSB, 314  
T\_GD\_LSB, 314  
T\_OA, 246  
T\_OA\_LSB, 314  
T\_OC\_LSB, 314  
T\_OE\_LSB, 314  
T\_OT\_LSB, 314  
TEXT\_BLACK, 314  
TEXT\_BLUE, 314  
TEXT\_BOLD\_BLACK, 314  
TEXT\_BOLD\_BLUE, 314  
TEXT\_BOLD\_CYAN, 314  
TEXT\_BOLD\_GREEN, 315  
TEXT\_BOLD\_MAGENTA, 315  
TEXT\_BOLD\_RED, 315  
TEXT\_BOLD\_WHITE, 315  
TEXT\_BOLD\_YELLOW, 315  
TEXT\_CYAN, 315  
TEXT\_GREEN, 315  
TEXT\_MAGENTA, 315  
TEXT\_RED, 315  
TEXT\_RESET, 315  
TEXT\_WHITE, 315  
TEXT\_YELLOW, 315  
TWO\_N10, 316  
TWO\_N11, 316  
TWO\_N14, 316  
TWO\_N15, 316  
TWO\_N16, 316  
TWO\_N17, 316  
TWO\_N18, 316  
TWO\_N19, 316  
TWO\_N2, 316  
TWO\_N20, 316  
TWO\_N21, 317  
TWO\_N22, 317  
TWO\_N23, 317  
TWO\_N24, 317  
TWO\_N25, 317  
TWO\_N26, 317  
TWO\_N27, 317  
TWO\_N29, 317  
TWO\_N30, 317  
TWO\_N31, 317  
TWO\_N32, 318  
TWO\_N33, 318  
TWO\_N34, 318  
TWO\_N35, 318  
TWO\_N38, 318  
TWO\_N39, 318  
TWO\_N40, 318  
TWO\_N43, 318  
TWO\_N44, 318  
TWO\_N46, 318  
TWO\_N48, 319  
TWO\_N5, 319  
TWO\_N50, 319  
TWO\_N51, 319  
TWO\_N55, 319  
TWO\_N57, 319  
TWO\_N59, 319  
TWO\_N6, 319  
TWO\_N60, 319  
TWO\_N66, 319  
TWO\_N68, 320  
TWO\_N8, 320  
TWO\_N9, 320  
TWO\_P11, 320  
TWO\_P12, 320  
TWO\_P14, 320  
TWO\_P16, 320  
TWO\_P19, 320  
TWO\_P3, 320  
TWO\_P31, 320  
TWO\_P32, 321

- TWO\_P4, [321](#)
- TWO\_P56, [321](#)
- TWO\_P57, [321](#)
- TWO\_P8, [321](#)
- TWO\_PI, [321](#)
- WN\_LSF\_LSB, [321](#)
- WN\_T\_LSB, [321](#)
- corr
  - half\_cyc\_tag, [1087](#)
- correlation\_length\_ms
  - Gnss\_Synchro, [903](#)
- CosBocCorrelationFunction
  - tracking\_libs, [202](#)
- counter
  - MACK\_tag\_and\_info, [1161](#)
- countwords\_
  - Allocator, [353](#)
- cov
  - peph\_t, [1288](#)
- cov\_xx
  - Monitor\_Pvt, [1173](#)
- cov\_xy
  - Monitor\_Pvt, [1173](#)
- cov\_yy
  - Monitor\_Pvt, [1173](#)
- cov\_yz
  - Monitor\_Pvt, [1173](#)
- cov\_zx
  - Monitor\_Pvt, [1173](#)
- cov\_zz
  - Monitor\_Pvt, [1173](#)
- cp
  - rtcm\_t, [1354](#)
- cpks
  - DSM\_nma\_header, [516](#)
- Cpu\_Multicorrelator, [467](#)
- cpu\_multicorrelator.h, [2274](#), [2275](#)
- Cpu\_Multicorrelator\_16sc, [468](#)
- cpu\_multicorrelator\_16sc.h, [2276](#), [2277](#)
- Cpu\_Multicorrelator\_Real\_Codes, [469](#)
- cpu\_multicorrelator\_real\_codes.h, [2277](#), [2278](#)
- cputime
  - rtksvr\_t, [1376](#)
- Crc
  - Gnss\_Ephemeris, [874](#)
- crc
  - eph\_t, [521](#)
- crc\_ok
  - cnav\_v27\_part\_t, [451](#)
- CRC\_test
  - Glonass\_Gnav\_Navigation\_Message, [806](#)
- create\_file\_source\_hook
  - FileTimestampSignalSource, [557](#)
  - FourBitCpxFileSignalSource, [575](#)
  - NsrFileSignalSource, [1210](#)
  - SpirFileSignalSource, [1452](#)
  - TwoBitCpxFileSignalSource, [1505](#)
  - TwoBitPackedFileSignalSource, [1510](#)
- createProtobuffer
  - Serdes\_Galileo\_Eph, [1412](#)
  - Serdes\_Gnss\_Synchro, [1414](#)
  - Serdes\_Gps\_Eph, [1416](#)
  - Serdes\_Monitor\_Pvt, [1418](#)
  - Serdes\_Nav\_Message, [1420](#)
- cross\_track\_m
  - HAS\_orbit\_corrections, [1090](#)
- Crs
  - Gnss\_Ephemeris, [874](#)
- crs
  - eph\_t, [521](#)
- cshort\_to\_float\_x2, [470](#)
- cshort\_to\_float\_x2.h, [1732](#), [1733](#)
- cshort\_to\_float\_x2\_sptr
  - algorithms\_libs, [96](#)
- cshort\_to\_gr\_complex, [472](#)
- cshort\_to\_gr\_complex.h, [1692](#), [1693](#)
- cshort\_to\_gr\_complex\_sptr
  - data\_type\_gr\_blocks, [86](#)
- cshort\_to\_grcomplex.h, [1682](#)
- CshortToGrComplex, [473](#)
  - connect, [475](#)
  - disconnect, [475](#)
  - get\_left\_block, [475](#)
  - get\_right\_block, [475](#)
  - implementation, [475](#)
  - item\_size, [475](#)
  - role, [475](#)
- CTR
  - Tag, [1474](#)
- CubatureFilter, [476](#)
- Cuc
  - Gnss\_Ephemeris, [875](#)
- cuc
  - eph\_t, [521](#)
- CUDA\_CALLABLE\_MEMBER\_DEVICE
  - cuda\_multicorrelator.h, [2281](#)
- CUDA\_CALLABLE\_MEMBER\_GLOBAL
  - cuda\_multicorrelator.h, [2281](#)
- cuda\_multicorrelator, [476](#)
- cuda\_multicorrelator.h, [2279](#), [2281](#)
  - CUDA\_CALLABLE\_MEMBER\_DEVICE, [2281](#)
  - CUDA\_CALLABLE\_MEMBER\_GLOBAL, [2281](#)
- current\_word\_
  - IONGSMSChunkUnpackingCtx< WT >, [1119](#)
- Cus
  - Gnss\_Ephemeris, [875](#)
- cus
  - eph\_t, [521](#)
- custom\_udp\_signal\_source.h, [1988](#), [1989](#)
- CustomUDPSignalSource, [477](#)
  - connect, [480](#)
  - disconnect, [480](#)
  - get\_left\_block, [480](#)
  - get\_right\_block, [480](#)
  - item\_size, [481](#)
- cycle

- rtksvr\_t, [1376](#)
- cyclic
  - solbuf\_t, [1441](#)
- D
  - obsd\_t, [1218](#)
- D1\_A0\_ALMANAC\_LSB
  - core\_system\_parameters, [266](#)
- D1\_A0\_LSB
  - core\_system\_parameters, [267](#)
- D1\_A0GAL\_LSB
  - core\_system\_parameters, [267](#)
- D1\_A0GLO\_LSB
  - core\_system\_parameters, [267](#)
- D1\_A0GPS\_LSB
  - core\_system\_parameters, [267](#)
- D1\_A0UTC\_LSB
  - core\_system\_parameters, [267](#)
- D1\_A1\_ALMANAC\_LSB
  - core\_system\_parameters, [267](#)
- D1\_A1\_LSB
  - core\_system\_parameters, [267](#)
- D1\_A1GAL\_LSB
  - core\_system\_parameters, [267](#)
- D1\_A1GLO\_LSB
  - core\_system\_parameters, [267](#)
- D1\_A1GPS\_LSB
  - core\_system\_parameters, [267](#)
- D1\_A1UTC\_LSB
  - core\_system\_parameters, [267](#)
- D1\_A2\_LSB
  - core\_system\_parameters, [267](#)
- D1\_ALPHA0\_LSB
  - core\_system\_parameters, [268](#)
- D1\_ALPHA1\_LSB
  - core\_system\_parameters, [268](#)
- D1\_ALPHA2\_LSB
  - core\_system\_parameters, [268](#)
- D1\_ALPHA3\_LSB
  - core\_system\_parameters, [268](#)
- D1\_BETA0\_LSB
  - core\_system\_parameters, [268](#)
- D1\_BETA1\_LSB
  - core\_system\_parameters, [268](#)
- D1\_BETA2\_LSB
  - core\_system\_parameters, [268](#)
- D1\_BETA3\_LSB
  - core\_system\_parameters, [268](#)
- D1\_CIC\_LSB
  - core\_system\_parameters, [268](#)
- D1\_CIS\_LSB
  - core\_system\_parameters, [268](#)
- D1\_CRC\_LSB
  - core\_system\_parameters, [268](#)
- D1\_CRS\_LSB
  - core\_system\_parameters, [268](#)
- D1\_CUC\_LSB
  - core\_system\_parameters, [269](#)
- D1\_CUS\_LSB
  - core\_system\_parameters, [269](#)
- D1\_DELTA\_I\_LSB
  - core\_system\_parameters, [269](#)
- D1\_DELTA\_N\_LSB
  - core\_system\_parameters, [269](#)
- D1\_E\_ALMANAC\_LSB
  - core\_system\_parameters, [269](#)
- D1\_E\_LSB
  - core\_system\_parameters, [269](#)
- D1\_I0\_LSB
  - core\_system\_parameters, [269](#)
- D1\_IDOT\_LSB
  - core\_system\_parameters, [269](#)
- D1\_M0\_ALMANAC\_LSB
  - core\_system\_parameters, [269](#)
- D1\_M0\_LSB
  - core\_system\_parameters, [269](#)
- D1\_OMEGA0\_ALMANAC\_LSB
  - core\_system\_parameters, [269](#)
- D1\_OMEGA0\_LSB
  - core\_system\_parameters, [269](#)
- D1\_OMEGA\_ALMANAC\_LSB
  - core\_system\_parameters, [270](#)
- D1\_OMEGA\_DOT\_ALMANAC\_LSB
  - core\_system\_parameters, [270](#)
- D1\_OMEGA\_DOT\_LSB
  - core\_system\_parameters, [270](#)
- D1\_OMEGA\_LSB
  - core\_system\_parameters, [270](#)
- D1\_SQRT\_A\_ALMANAC\_LSB
  - core\_system\_parameters, [270](#)
- D1\_SQRT\_A\_LSB
  - core\_system\_parameters, [270](#)
- d1\_subframe\_decoder
  - Beidou\_Dnav\_Navigation\_Message, [389](#)
- D1\_TGD1\_LSB
  - core\_system\_parameters, [270](#)
- D1\_TGD2\_LSB
  - core\_system\_parameters, [270](#)
- D1\_TOA\_LSB
  - core\_system\_parameters, [270](#)
- D1\_TOC\_LSB
  - core\_system\_parameters, [270](#)
- D1\_TOE\_LSB
  - core\_system\_parameters, [270](#)
- d2\_subframe\_decoder
  - Beidou\_Dnav\_Navigation\_Message, [389](#)
- D2R
  - core\_system\_parameters, [270](#)
- d\_acc
  - Sbas\_Ephemeris, [1387](#)
- d\_af0
  - Sbas\_Ephemeris, [1387](#)
- d\_af1
  - Sbas\_Ephemeris, [1387](#)
- d\_AXn
  - Glonass\_Gnav\_Ephemeris, [799](#)
- d\_AYn



- Glionass\_Gnav\_Ephemeris, 799
- d\_AZn
  - Glionass\_Gnav\_Ephemeris, 800
- d\_B1
  - Glionass\_Gnav\_Utc\_Model, 810
- d\_B2
  - Glionass\_Gnav\_Utc\_Model, 810
- d\_B\_n
  - Glionass\_Gnav\_Ephemeris, 800
- d\_C\_n
  - Glionass\_Gnav\_Almanac, 792
- d\_Delta\_i\_n\_A
  - Glionass\_Gnav\_Almanac, 792
- d\_Delta\_T\_n\_A
  - Glionass\_Gnav\_Almanac, 792
- d\_Delta\_T\_n\_A\_dot
  - Glionass\_Gnav\_Almanac, 792
- d\_Delta\_tau\_n
  - Glionass\_Gnav\_Ephemeris, 800
- d\_dsm\_header
  - OSNMA\_data, 1227
- d\_dsm\_kroot\_message
  - OSNMA\_data, 1227
- d\_dsm\_kroot\_new\_message
  - OSNMA\_data, 1227
- d\_dsm\_pkr\_message
  - OSNMA\_data, 1227
- d\_dtr
  - Glionass\_Gnav\_Ephemeris, 800
- d\_E\_n
  - Glionass\_Gnav\_Ephemeris, 800
- d\_epsilon\_n\_A
  - Glionass\_Gnav\_Almanac, 792
- d\_F\_T
  - Glionass\_Gnav\_Ephemeris, 800
- d\_gamma\_n
  - Glionass\_Gnav\_Ephemeris, 800
- d\_H\_n\_A
  - Glionass\_Gnav\_Almanac, 792
- d\_iodc
  - Glionass\_Gnav\_Ephemeris, 800
- d\_KP
  - Glionass\_Gnav\_Almanac, 793
- d\_l3rd\_n
  - Glionass\_Gnav\_Ephemeris, 800
- d\_l5th\_n
  - Glionass\_Gnav\_Ephemeris, 801
- d\_I\_n
  - Glionass\_Gnav\_Almanac, 793
- d\_lambda\_n\_A
  - Glionass\_Gnav\_Almanac, 793
- d\_M
  - Glionass\_Gnav\_Ephemeris, 801
- d\_m
  - Glionass\_Gnav\_Ephemeris, 801
- d\_M\_n\_A
  - Glionass\_Gnav\_Almanac, 793
- d\_mack\_message
  - OSNMA\_data, 1227
- OSNMA\_data, 1227
- d\_n
  - Glionass\_Gnav\_Ephemeris, 801
- d\_N\_4
  - Glionass\_Gnav\_Utc\_Model, 810
- d\_N\_A
  - Glionass\_Gnav\_Utc\_Model, 810
- d\_n\_A
  - Glionass\_Gnav\_Almanac, 793
- d\_N\_T
  - Glionass\_Gnav\_Ephemeris, 801
- d\_nav\_data
  - OSNMA\_data, 1227
- d\_nma\_header
  - OSNMA\_data, 1227
- d\_omega\_n\_A
  - Glionass\_Gnav\_Almanac, 793
- d\_P
  - Glionass\_Gnav\_Ephemeris, 801
- d\_P\_1
  - Glionass\_Gnav\_Ephemeris, 801
- d\_P\_2
  - Glionass\_Gnav\_Ephemeris, 801
- d\_P\_3
  - Glionass\_Gnav\_Ephemeris, 802
- d\_P\_4
  - Glionass\_Gnav\_Ephemeris, 802
- d\_pos
  - Sbas\_Ephemeris, 1387
- d\_satClkDrift
  - Glionass\_Gnav\_Ephemeris, 802
- d\_t\_b
  - Glionass\_Gnav\_Ephemeris, 802
- d\_t\_k
  - Glionass\_Gnav\_Ephemeris, 802
- d\_t\_lambda\_n\_A
  - Glionass\_Gnav\_Almanac, 793
- d\_tau\_c
  - Glionass\_Gnav\_Ephemeris, 802
  - Glionass\_Gnav\_Utc\_Model, 811
- d\_tau\_gps
  - Glionass\_Gnav\_Utc\_Model, 811
- d\_tau\_n
  - Glionass\_Gnav\_Ephemeris, 802
- d\_tau\_n\_A
  - Glionass\_Gnav\_Almanac, 793
- d\_tod
  - Glionass\_Gnav\_Ephemeris, 802
- d\_tof
  - Sbas\_Ephemeris, 1387
- d\_TOW
  - Glionass\_Gnav\_Ephemeris, 803
- d\_vel
  - Sbas\_Ephemeris, 1387
- d\_VXn
  - Glionass\_Gnav\_Ephemeris, 803
- d\_VYn
  - Glionass\_Gnav\_Ephemeris, 803

- d\_VZn
  - Glonass\_Gnav\_Ephemeris, [803](#)
- d\_WN
  - Glonass\_Gnav\_Ephemeris, [803](#)
- d\_Xn
  - Glonass\_Gnav\_Ephemeris, [803](#)
- d\_Yn
  - Glonass\_Gnav\_Ephemeris, [803](#)
- d\_yr
  - Glonass\_Gnav\_Ephemeris, [803](#)
- d\_Zn
  - Glonass\_Gnav\_Ephemeris, [803](#)
- daf0
  - sbslcorr\_t, [1401](#)
- daf1
  - sbslcorr\_t, [1402](#)
- data
  - an\_packet\_t, [359](#)
  - erp\_t, [524](#)
  - obs\_t, [1217](#)
  - solbuf\_t, [1441](#)
  - solstatbuf\_t, [1448](#)
  - tec\_t, [1482](#)
  - tle\_t, [1486](#)
- Data Type Adapters, [84](#)
- data\_codes
  - Dll\_Pll\_Conf\_Fpga, [500](#)
- data\_type\_adapter
  - ArraySignalConditioner, [362](#)
  - SignalConditioner, [1427](#)
- data\_type\_adapters, [84](#)
- data\_type\_gr\_blocks, [85](#)
  - cshort\_to\_gr\_complex\_sptr, [86](#)
  - interleaved\_byte\_to\_complex\_byte\_sptr, [86](#)
  - interleaved\_byte\_to\_complex\_short\_sptr, [86](#)
  - interleaved\_short\_to\_complex\_short\_sptr, [86](#)
- datum
  - solo\_t, [1443](#)
- dclk
  - ssr\_t, [1462](#)
- ddeph
  - ssr\_t, [1462](#)
- dec
  - cnav\_v27\_part\_t, [451](#)
- decisions
  - cnav\_v27\_part\_t, [451](#)
  - v27\_t, [1530](#)
- decisions\_count
  - v27\_t, [1530](#)
- decisions\_index
  - v27\_t, [1530](#)
- decode
  - Geohash, [786](#)
  - ReedSolomon, [1332](#)
  - Viterbi\_Decoder, [1531](#)
- decode\_block
  - Viterbi\_Decoder\_Sbas, [1532](#)
- decode\_item\_type
  - SignalSourceBase, [1434](#)
- decoded
  - cnav\_v27\_part\_t, [451](#)
- DEFAULT\_CARRIER\_SMOOTHING\_FACTOR
  - gnss\_sdr\_flags, [118](#)
- degf
  - solo\_t, [1443](#)
- del
  - sta\_t, [1465](#)
- delay
  - sbsigp\_t, [1398](#)
- Deleter, [481](#)
  - buffer\_, [482](#)
  - Deleter, [481](#)
  - operator(), [482](#)
- deln
  - eph\_t, [521](#)
- delta\_A
  - Gps\_CNAV\_Ephemeris, [928](#)
- DELTA\_A\_7
  - Galileo\_Almanac\_Helper, [602](#)
- DELTA\_A\_7\_LSB
  - core\_system\_parameters, [271](#)
- DELTA\_A\_8
  - Galileo\_Almanac\_Helper, [602](#)
- DELTA\_A\_8\_LSB
  - core\_system\_parameters, [271](#)
- DELTA\_A\_9
  - Galileo\_Almanac\_Helper, [602](#)
- DELTA\_A\_9\_LSB
  - core\_system\_parameters, [271](#)
- delta\_clock\_correction
  - Galileo\_HAS\_data, [633](#)
- delta\_clock\_correction\_clock\_subset
  - Galileo\_HAS\_data, [633](#)
- delta\_clock\_multiplier
  - Galileo\_HAS\_data, [633](#)
- delta\_clock\_multiplier\_clock\_subset
  - Galileo\_HAS\_data, [633](#)
- delta\_cross\_track
  - Galileo\_HAS\_data, [633](#)
- delta\_i
  - Gnss\_Almanac, [863](#)
- delta\_i\_7
  - Galileo\_Almanac\_Helper, [602](#)
- DELTA\_I\_7\_LSB
  - core\_system\_parameters, [271](#)
- delta\_i\_8
  - Galileo\_Almanac\_Helper, [602](#)
- DELTA\_I\_8\_LSB
  - core\_system\_parameters, [271](#)
- delta\_i\_9
  - Galileo\_Almanac\_Helper, [602](#)
- DELTA\_I\_9\_LSB
  - core\_system\_parameters, [271](#)
- delta\_in\_track
  - Galileo\_HAS\_data, [633](#)
- delta\_n



- Gnss\_Ephemeris, [875](#)
- DELTA\_N\_3\_LSB
  - core\_system\_parameters, [271](#)
- DELTA\_N\_LSB
  - core\_system\_parameters, [271](#)
- delta\_ndot
  - Gps\_CNAV\_Ephemeris, [928](#)
- delta\_OMEGAdot
  - Gps\_CNAV\_Ephemeris, [928](#)
- delta\_radial
  - Galileo\_HAS\_data, [633](#)
- Delta\_tLS
  - Galileo\_Utc\_Model, [659](#)
- Delta\_tLSF
  - Galileo\_Utc\_Model, [659](#)
- DeltaAred
  - Galileo\_Reduced\_CED, [652](#)
- Deltai0red
  - Galileo\_Reduced\_CED, [652](#)
- DeltaT\_LS
  - Beidou\_Dnav\_Utc\_Model, [393](#)
  - Gps\_Utc\_Model, [963](#)
- DELTA\_T\_LS\_LSB
  - core\_system\_parameters, [271](#)
- DeltaT\_LSF
  - Beidou\_Dnav\_Utc\_Model, [393](#)
  - Gps\_Utc\_Model, [963](#)
- DELTA\_T\_LSF\_LSB
  - core\_system\_parameters, [271](#)
- delttype
  - sta\_t, [1465](#)
- deph
  - ssr\_t, [1462](#)
- desig
  - tled\_t, [1488](#)
- dev
  - serial\_t, [1421](#)
- dev\_t
  - algorithms\_libs\_rtklib, [130](#)
- device\_name
  - Acq\_Conf\_Fpga, [330](#)
  - Dll\_Pll\_Conf\_Fpga, [500](#)
- DFRQ1\_GLO
  - core\_system\_parameters, [271](#)
- DFRQ2\_GLO
  - core\_system\_parameters, [271](#)
- dgps
  - nav\_t, [1192](#)
  - rtcm\_t, [1354](#)
- dgps\_t, [482](#)
  - iod, [483](#)
  - prc, [483](#)
  - rrc, [483](#)
  - t0, [483](#)
  - udre, [483](#)
- dim
  - cl\_fft\_plan, [442](#)
- dir
  - kernel\_info\_t, [1132](#)
  - url\_t, [1527](#)
- direct\_resampler\_conditioner.h, [1969](#)
- direct\_resampler\_conditioner\_cb, [483](#)
  - sample\_freq\_in, [484](#)
  - sample\_freq\_out, [484](#)
- direct\_resampler\_conditioner\_cb.h, [1972](#), [1973](#)
- direct\_resampler\_conditioner\_cb\_sptr
  - resampler\_gr\_blocks, [177](#)
- direct\_resampler\_conditioner\_cc, [485](#)
  - sample\_freq\_in, [486](#)
  - sample\_freq\_out, [486](#)
- direct\_resampler\_conditioner\_cc.h, [1974](#), [1975](#)
- direct\_resampler\_conditioner\_cc\_sptr
  - resampler\_gr\_blocks, [177](#)
- direct\_resampler\_conditioner\_cs, [486](#)
  - sample\_freq\_in, [488](#)
  - sample\_freq\_out, [488](#)
- direct\_resampler\_conditioner\_cs.h, [1976](#), [1977](#)
- direct\_resampler\_conditioner\_cs\_sptr
  - resampler\_gr\_blocks, [177](#)
- DirectResamplerConditioner, [488](#)
  - connect, [491](#)
  - disconnect, [491](#)
  - get\_left\_block, [491](#)
  - get\_right\_block, [491](#)
  - implementation, [491](#)
  - item\_size, [491](#)
  - role, [491](#)
- disable\_secondary\_codes
  - Fpga\_Multicorrelator\_8sc, [586](#)
- disconnect
  - Ad936xCustomSignalSource, [344](#)
  - Adrv9361z7035SignalSourceFPGA, [348](#)
  - ArraySignalConditioner, [362](#)
  - BeamformerFilter, [368](#)
  - BeidouB1iDIIPITracking, [396](#)
  - BeidouB1iPcpsAcquisition, [401](#)
  - BeidouB1iTelemetryDecoder, [406](#)
  - BeidouB3iDIIPITracking, [410](#)
  - BeidouB3iPcpsAcquisition, [414](#)
  - BeidouB3iTelemetryDecoder, [420](#)
  - ByteToShort, [425](#)
  - Channel, [429](#)
  - CshortToGrComplex, [475](#)
  - CustomUDPSignalSource, [480](#)
  - DirectResamplerConditioner, [491](#)
  - DMASignalSourceFPGA, [511](#)
  - FifoSignalSource, [535](#)
  - FileSourceBase, [552](#)
  - FirFilter, [560](#)
  - FlexibandSignalSource, [564](#)
  - Fmcomms2SignalSource, [568](#)
  - Fmcomms5SignalSourceFPGA, [571](#)
  - FreqXlatingFirFilter, [593](#)
  - GalileoE1BTelemetryDecoder, [663](#)
  - GalileoE1DIIPIVemITracking, [667](#)
  - GalileoE1DIIPIVemITrackingFpga, [672](#)

- GalileoE1Pcps8msAmbiguousAcquisition, [676](#)
- GalileoE1PcpsAmbiguousAcquisition, [683](#)
- GalileoE1PcpsAmbiguousAcquisitionFpga, [689](#)
- GalileoE1PcpsCccwsrAmbiguousAcquisition, [694](#)
- GalileoE1PcpsQuickSyncAmbiguousAcquisition, [700](#)
- GalileoE1PcpsTongAmbiguousAcquisition, [706](#)
- GalileoE1TcpConnectorTracking, [712](#)
- GalileoE5aDIIPIITracking, [716](#)
- GalileoE5aDIIPIITrackingFpga, [721](#)
- GalileoE5aNoncoherentIQAcquisitionCaf, [725](#)
- GalileoE5aPcpsAcquisition, [731](#)
- GalileoE5aPcpsAcquisitionFpga, [738](#)
- GalileoE5aTelemetryDecoder, [744](#)
- GalileoE5bDIIPIITracking, [748](#)
- GalileoE5bPcpsAcquisition, [753](#)
- GalileoE5bPcpsAcquisitionFpga, [760](#)
- GalileoE5bTelemetryDecoder, [766](#)
- GalileoE6DIIPIITracking, [770](#)
- GalileoE6PcpsAcquisition, [775](#)
- GalileoE6TelemetryDecoder, [780](#)
- GenSignalSource, [784](#)
- GlonassL1CaDIIPIICAidTracking, [828](#)
- GlonassL1CaDIIPIITracking, [832](#)
- GlonassL1CaPcpsAcquisition, [836](#)
- GlonassL1CaTelemetryDecoder, [841](#)
- GlonassL2CaDIIPIICAidTracking, [845](#)
- GlonassL2CaDIIPIITracking, [849](#)
- GlonassL2CaPcpsAcquisition, [853](#)
- GlonassL2CaTelemetryDecoder, [858](#)
- GNSSBlockInterface, [909](#)
- GNSSFlowgraph, [914](#)
- GpsL1CaDIIPIITracking, [967](#)
- GpsL1CaDIIPIITrackingFpga, [972](#)
- GpsL1CaDIIPIITrackingGPU, [976](#)
- GpsL1CaGaussianTracking, [980](#)
- GpsL1CaKfTracking, [984](#)
- GpsL1CaPcpsAcquisition, [989](#)
- GpsL1CaPcpsAcquisitionFpga, [1000](#)
- GpsL1CaPcpsAssistedAcquisition, [1006](#)
- GpsL1CaPcpsOpenCIAcquisition, [1011](#)
- GpsL1CaPcpsQuickSyncAcquisition, [1017](#)
- GpsL1CaPcpsTongAcquisition, [1022](#)
- GpsL1CaTcpConnectorTracking, [1027](#)
- GpsL1CaTelemetryDecoder, [1031](#)
- GpsL2CTelemetryDecoder, [1035](#)
- GpsL2MDIPIITracking, [1039](#)
- GpsL2MDIPIITrackingFpga, [1043](#)
- GpsL2MPcpsAcquisition, [1048](#)
- GpsL2MPcpsAcquisitionFpga, [1053](#)
- GpsL5DIIPIITracking, [1058](#)
- GpsL5DIIPIITrackingFpga, [1063](#)
- GpsL5iPcpsAcquisition, [1068](#)
- GpsL5iPcpsAcquisitionFpga, [1074](#)
- GpsL5TelemetryDecoder, [1079](#)
- HybridObservables, [1096](#)
- lbyteToCbyte, [1100](#)
- lbyteToComplex, [1103](#)
- lbyteToCshort, [1106](#)
- LONGSMSSignalSource, [1124](#)
- lshortToComplex, [1128](#)
- lshortToCshort, [1131](#)
- LabsatSignalSource, [1145](#)
- LimesdrSignalSource, [1155](#)
- MAX2771EVKITSignalSourceFPGA, [1165](#)
- MmseResamplerConditioner, [1168](#)
- MultichannelFileSignalSource, [1183](#)
- NotchFilter, [1201](#)
- NotchFilterLite, [1204](#)
- OsmosdrSignalSource, [1225](#)
- Pass\_Through, [1240](#)
- PlutosdrSignalSource, [1291](#)
- PulseBlankingFilter, [1306](#)
- RawArraySignalSource, [1330](#)
- Rtklib\_Pvt, [1363](#)
- RtlTcpSignalSource, [1385](#)
- SbasL1TelemetryDecoder, [1393](#)
- SignalConditioner, [1427](#)
- SignalGenerator, [1431](#)
- SpirGSS6450FileSignalSource, [1456](#)
- UhdSignalSource, [1514](#)
- ZmqSignalSource, [1536](#)
- display.h, [2431](#), [2432](#)
- DISPLAY\_COLORS
  - core\_system\_parameters, [246](#)
- display\_rate\_ms
  - Pvt\_Conf, [1309](#)
- dll\_bw\_hz
  - Dll\_Pll\_Conf, [494](#)
  - Dll\_Pll\_Conf\_Fpga, [501](#)
- dll\_bw\_narrow\_hz
  - Dll\_Pll\_Conf, [494](#)
  - Dll\_Pll\_Conf\_Fpga, [501](#)
- dll\_filter\_order
  - Dll\_Pll\_Conf, [494](#)
  - Dll\_Pll\_Conf\_Fpga, [501](#)
- dll\_nc\_e\_minus\_l\_normalized
  - tracking\_libs, [202](#)
- dll\_nc\_vemlp\_normalized
  - tracking\_libs, [202](#)
- Dll\_Pll\_Conf, [492](#)
  - bit\_synchronization\_time\_limit\_s, [493](#)
  - carrier\_aiding, [493](#)
  - carrier\_lock\_test\_smoother\_alpha, [493](#)
  - carrier\_lock\_test\_smoother\_samples, [494](#)
  - carrier\_lock\_th, [494](#)
  - cn0\_min, [494](#)
  - cn0\_samples, [494](#)
  - cn0\_smoother\_alpha, [494](#)
  - cn0\_smoother\_samples, [494](#)
  - dll\_bw\_hz, [494](#)
  - dll\_bw\_narrow\_hz, [494](#)
  - dll\_filter\_order, [494](#)
  - dll\_pull\_in\_bw\_hz, [494](#)
  - dump, [494](#)
  - dump\_filename, [494](#)

- dump\_mat, 495
- early\_late\_space\_chips, 495
- early\_late\_space\_narrow\_chips, 495
- enable\_doppler\_correction, 495
- enable\_fill\_pull\_in, 495
- enable\_fill\_steady\_state, 495
- extend\_correlation\_symbols, 495
- fill\_bw\_hz, 495
- fill\_filter\_order, 495
- fs\_in, 495
- high\_dyn, 495
- item\_type, 495
- max\_carrier\_lock\_fail, 496
- max\_code\_lock\_fail, 496
- pll\_bw\_hz, 496
- pll\_bw\_narrow\_hz, 496
- pll\_filter\_order, 496
- pll\_pull\_in\_bw\_hz, 496
- pull\_in\_time\_s, 496
- signal, 496
- slope, 496
- smoother\_length, 496
- spc, 496
- system, 496
- track\_pilot, 497
- vector\_length, 497
- very\_early\_late\_space\_chips, 497
- very\_early\_late\_space\_narrow\_chips, 497
- y\_intercept, 497
- dll\_pll\_conf.h, 2283, 2284
- Dll\_Pll\_Conf\_Fpga, 498
  - bit\_synchronization\_time\_limit\_s, 499
  - ca\_codes, 499
  - carrier\_aiding, 500
  - carrier\_lock\_test\_smoother\_alpha, 500
  - carrier\_lock\_test\_smoother\_samples, 500
  - carrier\_lock\_th, 500
  - cn0\_min, 500
  - cn0\_samples, 500
  - cn0\_smoother\_alpha, 500
  - cn0\_smoother\_samples, 500
  - code\_length\_chips, 500
  - code\_samples\_per\_chip, 500
  - data\_codes, 500
  - device\_name, 500
  - dll\_bw\_hz, 501
  - dll\_bw\_narrow\_hz, 501
  - dll\_filter\_order, 501
  - dll\_pull\_in\_bw\_hz, 501
  - dump, 501
  - dump\_filename, 501
  - dump\_mat, 501
  - early\_late\_space\_chips, 501
  - early\_late\_space\_narrow\_chips, 501
  - enable\_doppler\_correction, 501
  - enable\_fill\_pull\_in, 501
  - enable\_fill\_steady\_state, 501
  - extend\_correlation\_symbols, 502
  - extend\_fpga\_integration\_periods, 502
  - extended\_correlation\_in\_fpga, 502
  - fill\_bw\_hz, 502
  - fill\_filter\_order, 502
  - fpga\_integration\_period, 502
  - fs\_in, 502
  - high\_dyn, 502
  - max\_carrier\_lock\_fail, 502
  - max\_code\_lock\_fail, 502
  - pll\_bw\_hz, 502
  - pll\_bw\_narrow\_hz, 502
  - pll\_filter\_order, 503
  - pll\_pull\_in\_bw\_hz, 503
  - pull\_in\_time\_s, 503
  - signal, 503
  - slope, 503
  - smoother\_length, 503
  - spc, 503
  - system, 503
  - track\_pilot, 503
  - vector\_length, 503
  - very\_early\_late\_space\_chips, 503
  - very\_early\_late\_space\_narrow\_chips, 503
  - y\_intercept, 504
- dll\_pll\_conf\_fpga.h, 2285, 2286
- dll\_pll\_veml\_tracking, 504
- dll\_pll\_veml\_tracking.h, 2226, 2227
- dll\_pll\_veml\_tracking\_fpga, 505
  - ~dll\_pll\_veml\_tracking\_fpga, 508
  - general\_work, 508
  - reset, 508
  - set\_channel, 508
  - set\_gnss\_synchro, 508
  - start\_tracking, 508
  - stop\_tracking, 508
- dll\_pll\_veml\_tracking\_fpga.h, 2229, 2230
- dll\_pll\_veml\_tracking\_fpga\_sptr
  - tracking\_gr\_blocks, 196
- dll\_pll\_veml\_tracking\_sptr
  - tracking\_gr\_blocks, 196
- dll\_pull\_in\_bw\_hz
  - Dll\_Pll\_Conf, 494
  - Dll\_Pll\_Conf\_Fpga, 501
- DMA\_close
  - Fpga\_DMA, 582
- DMA\_open
  - Fpga\_DMA, 582
- dma\_signal\_source\_fpga.h, 1990, 1991
- DMA\_write
  - Fpga\_DMA, 582
- DMASignalSourceFPGA, 509
  - connect, 511
  - disconnect, 511
  - get\_left\_block, 511
  - get\_right\_block, 511
  - item\_size, 511
  - start, 511
- DN

- Beidou\_Dnav\_Utc\_Model, 393
- Galileo\_Utc\_Model, 659
- Gps\_Utc\_Model, 963
- DN\_LSB
  - core\_system\_parameters, 272
- doppler
  - sdr\_gnss\_packet\_t, 1408
- Doppler0
  - Gps\_Acq\_Assist, 917
- Doppler1
  - Gps\_Acq\_Assist, 917
- doppler\_max
  - Acq\_Conf, 326
  - Acq\_Conf\_Fpga, 330
- doppler\_min
  - Acq\_Conf, 326
- doppler\_step
  - Acq\_Conf, 326
  - Acq\_Conf\_Fpga, 330
- doppler\_step2
  - Acq\_Conf, 326
  - Acq\_Conf\_Fpga, 331
- dopplerUncertainty
  - Gps\_Acq\_Assist, 918
- downsampling\_factor
  - Acq\_Conf\_Fpga, 331
- downsampling\_filter\_delay
  - Acq\_Conf\_Fpga, 331
- downsampling\_filter\_num
  - Acq\_Conf\_Fpga, 331
- dpCA
  - raw\_t, 1324
- dpos
  - sbslcorr\_t, 1402
- ds
  - DSM\_KROOT\_message, 514
- dsm\_block\_id
  - DSM\_dsm\_header, 512
- DSM\_dsm\_header, 512
  - dsm\_block\_id, 512
  - dsm\_id, 512
- dsm\_id
  - DSM\_dsm\_header, 512
- DSM\_KROOT\_message, 513
  - alpha, 514
  - cidkr, 514
  - ds, 514
  - hf, 514
  - kroot, 514
  - ks, 514
  - maclt, 514
  - mf, 514
  - nb\_dk, 514
  - p\_dk, 515
  - pkid, 515
  - reserved, 515
  - reserved1, 515
  - towh\_k, 515
  - ts, 515
  - verified, 515
  - wn\_k, 515
- DSM\_nma\_header, 516
  - cid, 516
  - cpks, 516
  - nmas, 516
  - reserved, 516
- DSM\_PKR\_message, 517
  - itn, 518
  - mid, 518
  - nb\_dp, 518
  - npk, 518
  - npkt, 518
  - npktid, 518
  - p\_dp, 518
- dt
  - sbsfcorr\_t, 1396
- dtaun
  - geph\_t, 789
- dtr
  - Gnss\_Ephemeris, 875
  - sol\_t, 1438
- DTTOL
  - algorithms\_libs\_rtklib, 141
- dump
  - Acq\_Conf, 326
  - Dll\_Pll\_Conf, 494
  - Dll\_Pll\_Conf\_Fpga, 501
  - Kf\_Conf, 1136
  - Obs\_Conf, 1215
  - Pvt\_Conf, 1309
  - Tlm\_Conf, 1491
- dump\_channel
  - Acq\_Conf, 326
- dump\_crc\_stats
  - Tlm\_Conf, 1491
- dump\_crc\_stats\_filename
  - Tlm\_Conf, 1491
- dump\_filename
  - Acq\_Conf, 326
  - Dll\_Pll\_Conf, 494
  - Dll\_Pll\_Conf\_Fpga, 501
  - Kf\_Conf, 1137
  - Obs\_Conf, 1215
  - Pvt\_Conf, 1309
  - Tlm\_Conf, 1491
- dump\_mat
  - Dll\_Pll\_Conf, 495
  - Dll\_Pll\_Conf\_Fpga, 501
  - Kf\_Conf, 1137
  - Obs\_Conf, 1215
  - Pvt\_Conf, 1309
  - Tlm\_Conf, 1491
- dvel
  - sbslcorr\_t, 1402
- dynamics
  - prcopt\_t, 1298

- e
  - alm\_t, [354](#)
  - eph\_t, [521](#)
- E1B\_DVS
  - Galileo\_Ephemeris, [622](#)
- E1B\_HS
  - Galileo\_Almanac, [599](#)
  - Galileo\_Ephemeris, [622](#)
- E1B\_HS\_10
  - Galileo\_Almanac\_Helper, [602](#)
- E1B\_HS\_8
  - Galileo\_Almanac\_Helper, [602](#)
- E1B\_HS\_9
  - Galileo\_Almanac\_Helper, [602](#)
- E5a\_DVS
  - Galileo\_Ephemeris, [622](#)
- E5a\_HS
  - Galileo\_Almanac, [599](#)
  - Galileo\_Ephemeris, [622](#)
- E5a\_HS\_10
  - Galileo\_Almanac\_Helper, [602](#)
- E5a\_HS\_8
  - Galileo\_Almanac\_Helper, [603](#)
- E5a\_HS\_9
  - Galileo\_Almanac\_Helper, [603](#)
- E5b\_DVS
  - Galileo\_Ephemeris, [622](#)
- E5b\_HS
  - Galileo\_Almanac, [599](#)
  - Galileo\_Ephemeris, [622](#)
- E5b\_HS\_10
  - Galileo\_Almanac\_Helper, [603](#)
- E5b\_HS\_8
  - Galileo\_Almanac\_Helper, [603](#)
- E5b\_HS\_9
  - Galileo\_Almanac\_Helper, [603](#)
- E\_1\_LSB
  - core\_system\_parameters, [272](#)
- e\_7
  - Galileo\_Almanac\_Helper, [603](#)
- E\_7\_LSB
  - core\_system\_parameters, [272](#)
- e\_8
  - Galileo\_Almanac\_Helper, [603](#)
- E\_8\_LSB
  - core\_system\_parameters, [272](#)
- e\_9
  - Galileo\_Almanac\_Helper, [603](#)
- E\_9\_LSB
  - core\_system\_parameters, [272](#)
- e\_verification\_status
  - Tag, [1473](#)
- early\_late\_space\_chips
  - Dll\_Pll\_Conf, [495](#)
  - Dll\_Pll\_Conf\_Fpga, [501](#)
  - Kf\_Conf, [1137](#)
- early\_late\_space\_narrow\_chips
  - Dll\_Pll\_Conf, [495](#)
- Dll\_Pll\_Conf\_Fpga, [501](#)
- Kf\_Conf, [1137](#)
- ecc
  - Gnss\_Almanac, [863](#)
  - Gnss\_Ephemeris, [875](#)
  - tled\_t, [1488](#)
- ECCENTRICITY\_LSB
  - core\_system\_parameters, [272](#)
- edc.h, [2165](#), [2166](#)
- EFACT\_BDS
  - algorithms\_libs\_rtklib, [141](#)
- EFACT\_GAL
  - algorithms\_libs\_rtklib, [141](#)
- EFACT\_GLO
  - algorithms\_libs\_rtklib, [141](#)
- EFACT\_GPS
  - algorithms\_libs\_rtklib, [141](#)
- EFACT\_IRN
  - algorithms\_libs\_rtklib, [142](#)
- EFACT\_QZS
  - algorithms\_libs\_rtklib, [142](#)
- EFACT\_SBS
  - algorithms\_libs\_rtklib, [142](#)
- el
  - solstat\_t, [1446](#)
- eleno
  - tled\_t, [1488](#)
- Elevation
  - Gps\_Acq\_Assist, [918](#)
- elmaskar
  - prcopt\_t, [1298](#)
- elmaskhold
  - prcopt\_t, [1298](#)
- elmin
  - prcopt\_t, [1298](#)
- empty
  - Concurrent\_Queue< Data >, [458](#)
- ena
  - exterr\_t, [529](#)
  - snrmask\_t, [1437](#)
- ENABDS
  - algorithms\_libs\_rtklib, [130](#)
- enable\_carrier\_nco\_cmd
  - TrackingCmd, [1498](#)
- enable\_carrier\_smoothing
  - Obs\_Conf, [1215](#)
- enable\_code\_nco\_cmd
  - TrackingCmd, [1498](#)
- enable\_doppler\_correction
  - Dll\_Pll\_Conf, [495](#)
  - Dll\_Pll\_Conf\_Fpga, [501](#)
  - Kf\_Conf, [1137](#)
- enable\_E6
  - Obs\_Conf, [1215](#)
- enable\_fll\_pull\_in
  - Dll\_Pll\_Conf, [495](#)
  - Dll\_Pll\_Conf\_Fpga, [501](#)
- enable\_fll\_steady\_state

- Dll\_Pll\_Conf, [495](#)
- Dll\_Pll\_Conf\_Fpga, [501](#)
- enable\_monitor\_output
  - Acq\_Conf, [326](#)
- enable\_navdata\_monitor
  - Tlm\_Conf, [1491](#)
- enable\_pvt\_kf
  - Pvt\_Conf, [1309](#)
- enable\_reed\_solomon
  - Galileo\_Inav\_Message, [639](#)
  - Tlm\_Conf, [1491](#)
- enable\_rx\_clock\_correction
  - Pvt\_Conf, [1309](#)
- enable\_secondary\_codes
  - Fpga\_Multicorrelator\_8sc, [586](#)
- enable\_zero\_padding
  - Acq\_Conf\_Fpga, [331](#)
- ENAGLO
  - algorithms\_libs\_rtklib, [130](#)
- encode
  - Geohash, [786](#)
- encode\_with\_generator\_matrix
  - ReedSolomon, [1332](#)
- encode\_with\_generator\_poly
  - ReedSolomon, [1332](#)
- encoding\_from\_string
  - ion\_gsms\_stream\_encodings.h, [2101](#)
- end
  - solbuf\_t, [1441](#)
- eph
  - nav\_t, [1192](#)
- eph\_t, [519](#)
  - A, [520](#)
  - Adot, [520](#)
  - apply\_has\_corrections, [520](#)
  - cic, [520](#)
  - cis, [520](#)
  - code, [520](#)
  - crc, [521](#)
  - crs, [521](#)
  - cuc, [521](#)
  - cus, [521](#)
  - deln, [521](#)
  - e, [521](#)
  - f0, [521](#)
  - f1, [521](#)
  - f2, [521](#)
  - fit, [521](#)
  - flag, [521](#)
  - has\_clock\_correction\_m, [521](#)
  - has\_orbit\_cross\_track\_correction\_m, [522](#)
  - has\_orbit\_in\_track\_correction\_m, [522](#)
  - has\_orbit\_radial\_correction\_m, [522](#)
  - i0, [522](#)
  - idot, [522](#)
  - iodc, [522](#)
  - iode, [522](#)
  - isc, [522](#)
  - M0, [522](#)
  - ndot, [522](#)
  - omg, [522](#)
  - OMG0, [522](#)
  - OMGd, [523](#)
  - sat, [523](#)
  - sva, [523](#)
  - svh, [523](#)
  - tgd, [523](#)
  - toc, [523](#)
  - toe, [523](#)
  - toes, [523](#)
  - ttr, [523](#)
  - week, [523](#)
- eph\_to\_rtklib
  - algorithms\_libs\_rtklib, [139](#)
- EPHOPT\_BRDC
  - algorithms\_libs\_rtklib, [142](#)
- EPHOPT\_LEX
  - algorithms\_libs\_rtklib, [142](#)
- EPHOPT\_PREC
  - algorithms\_libs\_rtklib, [142](#)
- EPHOPT\_SBAS
  - algorithms\_libs\_rtklib, [143](#)
- EPHOPT\_SSRAPC
  - algorithms\_libs\_rtklib, [143](#)
- EPHOPT\_SSRCOM
  - algorithms\_libs\_rtklib, [143](#)
- ephsat
  - raw\_t, [1324](#)
  - rtcm\_t, [1354](#)
- epoch
  - ambc\_t, [356](#)
  - tled\_t, [1488](#)
- eratio
  - prcopt\_t, [1298](#)
- erp
  - nav\_t, [1192](#)
- erp\_t, [524](#)
  - data, [524](#)
  - n, [524](#)
  - nmax, [524](#)
- erpd\_t, [525](#)
  - lod, [525](#)
  - mjd, [525](#)
  - ut1\_utc, [525](#)
  - xp, [525](#)
  - xpr, [526](#)
  - yp, [526](#)
  - ypr, [526](#)
- err
  - prcopt\_t, [1298](#)
- ERR\_BRDCI
  - algorithms\_libs\_rtklib, [143](#)
- ERR\_CBIAS
  - algorithms\_libs\_rtklib, [143](#)
- ERR\_ION
  - rtklib\_pntpos.h, [1819](#)

ERR\_SAAS  
     algorithms\_libs\_rtklib, 143  
 ERR\_TROP  
     rtklib\_pntpos.h, 1819  
 errbuf  
     rtk\_t, 1358  
 error  
     ftp\_t, 595  
     serial\_t, 1421  
 etype  
     tled\_t, 1488  
 event\_type  
     Channel\_Event, 431  
     Command\_Event, 453  
 excludelimit  
     Acq\_Conf\_Fpga, 331  
 Exponential\_Smoother, 526  
     ~Exponential\_Smoother, 527  
     Exponential\_Smoother, 527  
     operator=, 528  
     set\_alpha, 528  
     set\_samples\_for\_initialization, 528  
 exponential\_smoother.h, 2287, 2288  
 exred  
     Galileo\_Reduced\_CED, 652  
 exsats  
     prcopt\_t, 1299  
 extend\_correlation\_symbols  
     Dll\_Pll\_Conf, 495  
     Dll\_Pll\_Conf\_Fpga, 502  
     Kf\_Conf, 1137  
 extend\_fpga\_integration\_periods  
     Dll\_Pll\_Conf\_Fpga, 502  
 extended\_correlation\_in\_fpga  
     Dll\_Pll\_Conf\_Fpga, 502  
 exterr  
     prcopt\_t, 1299  
 EXTERR\_CLK  
     rtklib\_preceph.h, 1828  
 EXTERR\_EPH  
     rtklib\_preceph.h, 1828  
 exterr\_t, 528  
     cerr, 529  
     ena, 529  
     gloicb, 529  
     gpsglob, 529  
     perr, 529  
 eyred  
     Galileo\_Reduced\_CED, 652  
 f0  
     alm\_t, 354  
     eph\_t, 521  
 f1  
     alm\_t, 354  
     eph\_t, 521  
 f2  
     eph\_t, 521  
 fatalfunc\_t  
     algorithms\_libs\_rtklib, 139  
 fcb  
     nav\_t, 1192  
 fcbd\_t, 530  
     bias, 530  
     std, 530  
     te, 530  
     ts, 530  
 fcorr  
     sbssatp\_t, 1406  
 FE\_WGS84  
     algorithms\_libs\_rtklib, 144  
 fec.h, 2166, 2168  
     V27POLYA, 2168  
     V27POLYB, 2168  
 fft\_base\_kernels.h, 1774, 1775  
 fft\_internal.h, 1778, 1779  
     kernel\_dir\_t, 1779  
 fft\_size  
     Acq\_Conf\_Fpga, 331  
 fifo\_reader.h, 2044, 2045  
 fifo\_signal\_source.h, 1992, 1993  
 FifoReader, 531  
     sptr, 532  
     start, 532  
 FifoSignalSource, 533  
     connect, 535  
     disconnect, 535  
     get\_left\_block, 535  
     get\_right\_block, 536  
     item\_size, 536  
 file  
     ftp\_t, 595  
 file\_configuration.h, 2386, 2387  
 file\_signal\_source.h, 1994, 1995  
 file\_source\_base.h, 1995, 1996  
 file\_t, 537  
     fp, 538  
     fp\_tag, 538  
     fp\_tag\_tmp, 538  
     fp\_tmp, 538  
     fpos, 538  
     lock, 538  
     mode, 538  
     offset, 538  
     openpath, 538  
     path, 539  
     repmode, 539  
     speed, 539  
     start, 539  
     swapintv, 539  
     tick, 539  
     tick\_f, 539  
     time, 539  
     timetag, 539  
     wtime, 539  
 file\_timestamp\_signal\_source.h, 1999  
 FileConfiguration, 540

- property, [542](#), [543](#)
- set\_property, [543](#)
- filename
  - FileSourceBase, [552](#)
  - MultichannelFileSignalSource, [1183](#)
  - SpirGSS6450FileSignalSource, [1456](#)
- FILEPATHSEP
  - algorithms\_libs\_rtklib, [144](#)
- files
  - rtksvr\_t, [1376](#)
- FileSignalSource, [543](#)
- FileSourceBase, [547](#)
  - computeSamplesInFile, [551](#)
  - connect, [551](#)
  - disconnect, [552](#)
  - filename, [552](#)
  - FileSourceBase, [551](#)
  - get\_left\_block, [552](#)
  - get\_right\_block, [552](#)
  - init, [552](#)
  - item\_size, [552](#)
  - item\_type, [552](#)
  - itemTypeToSize, [552](#)
  - packetsPerSample, [552](#)
  - repeat, [552](#)
  - samples, [552](#)
  - samplesToSkip, [553](#)
  - sampling\_frequency, [553](#)
  - source, [553](#)
  - source\_item\_size, [553](#)
- FileTimestampSignalSource, [553](#)
  - create\_file\_source\_hook, [557](#)
  - pre\_connect\_hook, [557](#)
  - pre\_disconnect\_hook, [557](#)
  - source, [557](#)
- find\_uio\_dev\_file\_name
  - core\_libs, [210](#)
- findUtmZone
  - algorithms\_libs, [100](#)
- FIR\_BUF\_SIZE
  - signal\_source\_libs, [184](#)
- fir\_filter.h, [1702](#)
- FirFilter, [558](#)
  - ~FirFilter, [560](#)
  - connect, [560](#)
  - disconnect, [560](#)
  - FirFilter, [560](#)
  - get\_left\_block, [560](#)
  - get\_right\_block, [560](#)
  - implementation, [560](#)
  - item\_size, [561](#)
  - role, [561](#)
- FIRST\_RS\_BIT
  - core\_system\_parameters, [272](#)
- FIRST\_RS\_BIT\_AFTER\_IODNAV
  - core\_system\_parameters, [272](#)
- fit
  - eph\_t, [521](#)
- fit\_interval\_flag
  - Beidou\_Dnav\_Ephemeris, [383](#)
  - Gps\_Ephemeris, [942](#)
- five\_bit\_look\_up
  - ion\_gsms\_stream\_encodings.h, [2101](#)
- fix
  - ssat\_t, [1459](#)
- fixcnt
  - ambc\_t, [356](#)
- flag
  - eph\_t, [521](#)
  - raw\_t, [1324](#)
  - solstat\_t, [1446](#)
  - stec\_t, [1466](#)
- flag\_all\_ephemeris
  - Galileo\_Ephemeris, [622](#)
- flag\_nmea\_tty\_port
  - Pvt\_Conf, [1309](#)
- Flag\_PLL\_180\_deg\_phase\_locked
  - Gnss\_Synchro, [903](#)
- flag\_rtcn\_server
  - Pvt\_Conf, [1310](#)
- flag\_rtcn\_tty\_port
  - Pvt\_Conf, [1310](#)
- flag\_utc\_model
  - Galileo\_Utc\_Model, [659](#)
- Flag\_valid\_acquisition
  - Gnss\_Synchro, [903](#)
- Flag\_valid\_pseudorange
  - Gnss\_Synchro, [903](#)
- Flag\_valid\_symbol\_output
  - Gnss\_Synchro, [903](#)
- Flag\_valid\_word
  - Gnss\_Synchro, [903](#)
- flags
  - ambc\_t, [356](#)
- flexiband\_signal\_source.h, [2000](#), [2001](#)
- FlexibandSignalSource, [561](#)
  - connect, [564](#)
  - disconnect, [564](#)
  - get\_left\_block, [564](#)
  - get\_right\_block, [564](#)
  - item\_size, [565](#)
- fil\_bw\_hz
  - Dll\_Pll\_Conf, [495](#)
  - Dll\_Pll\_Conf\_Fpga, [502](#)
- fil\_filter\_order
  - Dll\_Pll\_Conf, [495](#)
  - Dll\_Pll\_Conf\_Fpga, [502](#)
- fil\_four\_quadrant\_atan
  - tracking\_libs, [203](#)
- flowgraph
  - ControlThread, [466](#)
- fmcomms2\_signal\_source.h, [2002](#), [2003](#)
- Fmcomms2SignalSource, [566](#)
  - connect, [568](#)
  - disconnect, [568](#)
  - get\_left\_block, [568](#)



- get\_right\_block, [568](#)
- item\_size, [568](#)
- fmcomms5\_signal\_source\_fpga.h, [2004](#), [2005](#)
- Fmcomms5SignalSourceFPGA, [569](#)
  - connect, [571](#)
  - disconnect, [571](#)
  - get\_left\_block, [571](#)
  - get\_right\_block, [571](#)
  - item\_size, [571](#)
- FNAV\_A0\_4\_LSB
  - core\_system\_parameters, [272](#)
- FNAV\_A0G\_4\_LSB
  - core\_system\_parameters, [272](#)
- FNAV\_A12\_2\_LSB
  - core\_system\_parameters, [272](#)
- FNAV\_A1\_4\_LSB
  - core\_system\_parameters, [272](#)
- FNAV\_A1G\_4\_LSB
  - core\_system\_parameters, [273](#)
- FNAV\_AF0\_1\_LSB
  - core\_system\_parameters, [273](#)
- FNAV\_AF0\_5\_LSB
  - core\_system\_parameters, [273](#)
- FNAV\_AF1\_1\_LSB
  - core\_system\_parameters, [273](#)
- FNAV\_AF1\_5\_LSB
  - core\_system\_parameters, [273](#)
- FNAV\_AF2\_1\_LSB
  - core\_system\_parameters, [273](#)
- FNAV\_AI0\_1\_LSB
  - core\_system\_parameters, [273](#)
- FNAV\_AI1\_1\_LSB
  - core\_system\_parameters, [273](#)
- FNAV\_AI2\_1\_LSB
  - core\_system\_parameters, [273](#)
- FNAV\_BGD\_1\_LSB
  - core\_system\_parameters, [273](#)
- FNAV\_CIC\_4\_LSB
  - core\_system\_parameters, [273](#)
- FNAV\_CIS\_4\_LSB
  - core\_system\_parameters, [273](#)
- FNAV\_CRC\_3\_LSB
  - core\_system\_parameters, [274](#)
- FNAV\_CRS\_3\_LSB
  - core\_system\_parameters, [274](#)
- FNAV\_CUC\_3\_LSB
  - core\_system\_parameters, [274](#)
- FNAV\_CUS\_3\_LSB
  - core\_system\_parameters, [274](#)
- FNAV\_DELTA12\_5\_LSB
  - core\_system\_parameters, [274](#)
- FNAV\_DELTAI\_5\_LSB
  - core\_system\_parameters, [274](#)
- FNAV\_DELTAN\_3\_LSB
  - core\_system\_parameters, [274](#)
- FNAV\_E\_2\_LSB
  - core\_system\_parameters, [274](#)
- FNAV\_E\_5\_LSB
  - core\_system\_parameters, [274](#)
- FNAV\_I0\_3\_LSB
  - core\_system\_parameters, [274](#)
- FNAV\_IDOT\_2\_LSB
  - core\_system\_parameters, [274](#)
- FNAV\_M0\_2\_LSB
  - core\_system\_parameters, [274](#)
- FNAV\_M0\_5\_LSB
  - core\_system\_parameters, [275](#)
- FNAV\_OMEGA0\_2\_LSB
  - core\_system\_parameters, [275](#)
- FNAV\_OMEGA0\_5\_LSB
  - core\_system\_parameters, [275](#)
- FNAV\_OMEGADOT\_2\_LSB
  - core\_system\_parameters, [275](#)
- FNAV\_OMEGADOT\_5\_LSB
  - core\_system\_parameters, [275](#)
- FNAV\_T0A\_5\_LSB
  - core\_system\_parameters, [275](#)
- FNAV\_T0C\_1\_LSB
  - core\_system\_parameters, [275](#)
- FNAV\_T0E\_3\_LSB
  - core\_system\_parameters, [275](#)
- FNAV\_T0G\_4\_LSB
  - core\_system\_parameters, [275](#)
- FNAV\_T0T\_4\_LSB
  - core\_system\_parameters, [275](#)
- FNAV\_W\_3\_LSB
  - core\_system\_parameters, [275](#)
- FNAV\_W\_5\_LSB
  - core\_system\_parameters, [275](#)
- format
  - cl\_fft\_plan, [442](#)
  - opt\_t, [1222](#)
  - rtksvr\_t, [1376](#)
- four\_bit\_cpx\_file\_signal\_source.h, [2007](#)
- four\_bit\_look\_up
  - ion\_gsms\_stream\_encodings.h, [2102](#)
- FourBitCpxFileSignalSource, [572](#)
  - create\_file\_source\_hook, [575](#)
  - itemTypeToSize, [575](#)
  - packetsPerSample, [575](#)
  - pre\_connect\_hook, [575](#)
  - pre\_disconnect\_hook, [575](#)
  - source, [576](#)
- FP
  - ion\_gsms\_stream\_encodings.h, [2102](#)
- fp
  - file\_t, [538](#)
- fp\_tag
  - file\_t, [538](#)
- fp\_tag\_tmp
  - file\_t, [538](#)
- fp\_tmp
  - file\_t, [538](#)
- Fpga\_Acquisition, [576](#)
  - ~Fpga\_Acquisition, [577](#)
  - close\_device, [577](#)

- configure\_acquisition, 577
- Fpga\_Acquisition, 577
- init, 578
- open\_device, 578
- read\_acquisition\_results, 578
- reset\_acquisition, 578
- run\_acquisition, 578
- set\_block\_exp, 578
- set\_doppler\_max, 578
- set\_doppler\_step, 579
- set\_doppler\_sweep, 579
- set\_local\_code, 579
- stop\_acquisition, 579
- write\_local\_code, 579
- fpga\_acquisition.h, 1667, 1668
- Fpga\_buffer\_monitor, 579
  - ~Fpga\_buffer\_monitor, 580
  - check\_buffer\_overflow\_and\_monitor\_buffer\_status, 580
  - Fpga\_buffer\_monitor, 580
- fpga\_buffer\_monitor.h, 2077, 2078
- Fpga\_DMA, 581
  - ~Fpga\_DMA, 581
  - DMA\_close, 582
  - DMA\_open, 582
  - DMA\_write, 582
  - Fpga\_DMA, 581
  - get\_buffer\_address, 582
- fpga\_dma-proxy.h, 2079, 2081
  - buffer, 2081
  - length, 2081
  - proxy\_status, 2081
- Fpga\_dynamic\_bit\_selection, 582
  - ~Fpga\_dynamic\_bit\_selection, 583
  - bit\_selection, 583
  - Fpga\_dynamic\_bit\_selection, 583
- fpga\_dynamic\_bit\_selection.h, 2082, 2083
- fpga\_integration\_period
  - Dll\_Pll\_Conf\_Fpga, 502
- fpga\_multicorrelator.h, 2289, 2290
- Fpga\_Multicorrelator\_8sc, 583
  - ~Fpga\_Multicorrelator\_8sc, 585
  - Carrier\_wipeoff\_multicorrelator\_resampler, 585
  - disable\_secondary\_codes, 586
  - enable\_secondary\_codes, 586
  - Fpga\_Multicorrelator\_8sc, 585
  - free, 586
  - initialize\_secondary\_code, 586
  - lock\_channel, 586
  - open\_channel, 586
  - read\_sample\_counter, 586
  - set\_initial\_sample, 586
  - set\_local\_code\_and\_taps, 586
  - set\_output\_vectors, 586
  - set\_secondary\_code\_lengths, 587
  - unlock\_channel, 587
  - update\_local\_code, 587
  - update\_prn\_code\_length, 587
- Fpga\_spidev, 588
  - ~Fpga\_spidev, 588
  - Fpga\_spidev, 588
  - read\_reg32, 589
  - SPI\_close, 589
  - SPI\_open, 589
  - write\_reg32, 589
- fpga\_spidev.h, 2084, 2085
- Fpga\_Switch, 589
  - ~Fpga\_Switch, 590
  - Fpga\_Switch, 590
  - set\_switch\_position, 590
- fpga\_switch.h, 2086, 2087
- fpos
  - file\_t, 538
- free
  - Fpga\_Multicorrelator\_8sc, 586
- freq
  - half\_cyc\_tag, 1087
- FREQ1
  - core\_system\_parameters, 276
- FREQ1\_BDS
  - core\_system\_parameters, 276
- FREQ1\_GLO
  - core\_system\_parameters, 276
- FREQ2
  - core\_system\_parameters, 276
- FREQ2\_BDS
  - core\_system\_parameters, 276
- FREQ2\_GLO
  - core\_system\_parameters, 276
- FREQ3\_BDS
  - core\_system\_parameters, 276
- FREQ3\_GLO
  - core\_system\_parameters, 276
- FREQ5
  - core\_system\_parameters, 276
- FREQ6
  - core\_system\_parameters, 276
- FREQ7
  - core\_system\_parameters, 277
- FREQ8
  - core\_system\_parameters, 277
- FREQ9
  - core\_system\_parameters, 277
- freq\_xlating\_fir\_filter.h, 1704
- freqn
  - raw\_t, 1324
- freqopt
  - prcopt\_t, 1299
- FreqXlatingFirFilter, 590
  - connect, 593
  - disconnect, 593
  - get\_left\_block, 593
  - get\_right\_block, 593
  - implementation, 593
  - item\_size, 593
  - role, 593

- FRIEND\_TEST
  - osnma\_msg\_receiver.h, [2361](#)
- front
  - algorithms\_libs, [100](#)
- frq
  - geph\_t, [789](#)
  - solstat\_t, [1446](#)
- fs
  - Gnss\_Synchro, [903](#)
- fs\_hz
  - stream\_cfg, [1467](#)
- fs\_in
  - Acq\_Conf, [326](#)
  - Acq\_Conf\_Fpga, [331](#)
  - Dll\_Pll\_Conf, [495](#)
  - Dll\_Pll\_Conf\_Fpga, [502](#)
  - Kf\_Conf, [1137](#)
- ftime
  - rtksvr\_t, [1376](#)
- FTP\_CMD
  - algorithms\_libs\_rtklib, [131](#)
  - rtklib\_stream.h, [1866](#)
- ftp\_t, [594](#)
  - addr, [595](#)
  - error, [595](#)
  - file, [595](#)
  - local, [595](#)
  - passwd, [595](#)
  - proto, [595](#)
  - state, [595](#)
  - thread, [595](#)
  - tnext, [595](#)
  - topts, [595](#)
  - user, [595](#)
- FTP\_TIMEOUT
  - algorithms\_libs\_rtklib, [144](#)
  - rtklib\_stream.h, [1866](#)
- gal\_almanac\_map
  - Gnss\_Sdr\_Supl\_Client, [890](#)
- gal\_ephemeris\_map
  - Gnss\_Sdr\_Supl\_Client, [890](#)
- gal\_iono
  - Gnss\_Sdr\_Supl\_Client, [890](#)
- gal\_utc
  - Gnss\_Sdr\_Supl\_Client, [890](#)
- galhas\_status
  - Monitor\_Pvt, [1173](#)
- Galileo\_Almanac, [596](#)
  - E1B\_HS, [599](#)
  - E5a\_HS, [599](#)
  - E5b\_HS, [599](#)
  - Galileo\_Almanac, [598](#)
  - IODa, [599](#)
  - serialize, [599](#)
- galileo\_almanac.h, [2433](#), [2434](#)
- Galileo\_Almanac\_Helper, [599](#)
  - af0\_10, [601](#)
  - af0\_8, [601](#)
  - af0\_9, [601](#)
  - af1\_10, [601](#)
  - af1\_8, [602](#)
  - af1\_9, [602](#)
  - DELTA\_A\_7, [602](#)
  - DELTA\_A\_8, [602](#)
  - DELTA\_A\_9, [602](#)
  - delta\_i\_7, [602](#)
  - delta\_i\_8, [602](#)
  - delta\_i\_9, [602](#)
  - E1B\_HS\_10, [602](#)
  - E1B\_HS\_8, [602](#)
  - E1B\_HS\_9, [602](#)
  - E5a\_HS\_10, [602](#)
  - E5a\_HS\_8, [603](#)
  - E5a\_HS\_9, [603](#)
  - E5b\_HS\_10, [603](#)
  - E5b\_HS\_8, [603](#)
  - E5b\_HS\_9, [603](#)
  - e\_7, [603](#)
  - e\_8, [603](#)
  - e\_9, [603](#)
  - Galileo\_Almanac\_Helper, [601](#)
  - IOD\_a\_10, [603](#)
  - IOD\_a\_7, [603](#)
  - IOD\_a\_8, [603](#)
  - IOD\_a\_9, [603](#)
  - M0\_10, [604](#)
  - M0\_7, [604](#)
  - M0\_9, [604](#)
  - Omega0\_10, [604](#)
  - Omega0\_7, [604](#)
  - Omega0\_8, [604](#)
  - omega\_7, [604](#)
  - omega\_8, [604](#)
  - omega\_9, [604](#)
  - Omega\_dot\_10, [604](#)
  - Omega\_dot\_7, [604](#)
  - Omega\_dot\_8, [604](#)
  - SVID1\_7, [605](#)
  - SVID2\_8, [605](#)
  - SVID3\_9, [605](#)
  - t0a\_7, [605](#)
  - t0a\_9, [605](#)
  - WN\_a\_7, [605](#)
  - WN\_a\_9, [605](#)
- galileo\_almanac\_helper.h, [2435](#), [2436](#)
- galileo\_almanac\_map
  - Rtklib\_Solver, [1372](#)
- Galileo\_CNAV.h, [2437](#), [2440](#)
- GALILEO\_CNAV\_BITS\_FOR\_CRC
  - core\_system\_parameters, [277](#)
- GALILEO\_CNAV\_BYTES\_FOR\_CRC
  - core\_system\_parameters, [277](#)
- GALILEO\_CNAV\_CRC\_LENGTH
  - core\_system\_parameters, [277](#)
- GALILEO\_CNAV\_HAS\_PAGE\_DATA\_BITS
  - core\_system\_parameters, [277](#)

- GALILEO\_CNAV\_INFORMATION\_VECTOR\_LENGTH
  - core\_system\_parameters, [277](#)
- GALILEO\_CNAV\_INTERLEAVER\_COLS
  - core\_system\_parameters, [277](#)
- GALILEO\_CNAV\_INTERLEAVER\_ROWS
  - core\_system\_parameters, [277](#)
- GALILEO\_CNAV\_MAX\_NUMBER\_SYMBOLS\_ENCODED
  - core\_system\_parameters, [277](#)
- Galileo\_Cnav\_Message, [605](#)
  - get\_flag\_CRC\_test, [606](#)
  - get\_HAS\_encoded\_page, [606](#)
  - have\_new\_HAS\_page, [606](#)
  - is\_HAS\_in\_test\_mode, [606](#)
  - is\_HAS\_page\_dummy, [607](#)
  - set\_time\_stamp, [607](#)
  - set\_tow, [607](#)
- galileo\_cnav\_message.h, [2442](#), [2443](#)
- GALILEO\_CNAV\_MESSAGE\_BITS\_PER\_PAGE
  - core\_system\_parameters, [278](#)
- GALILEO\_CNAV\_MT1\_HEADER\_BITS
  - core\_system\_parameters, [278](#)
- GALILEO\_CNAV\_OCTETS\_IN\_SUBPAGE
  - core\_system\_parameters, [278](#)
- GALILEO\_CNAV\_PAGE\_HEADER\_BITS
  - core\_system\_parameters, [278](#)
- GALILEO\_CNAV\_PAGE\_MS
  - core\_system\_parameters, [278](#)
- GALILEO\_CNAV\_PAGE\_RESERVED\_BITS
  - core\_system\_parameters, [278](#)
- GALILEO\_CNAV\_PREAMBLE
  - core\_system\_parameters, [278](#)
- GALILEO\_CNAV\_PREAMBLE\_LENGTH\_BITS
  - core\_system\_parameters, [278](#)
- GALILEO\_CNAV\_PREAMBLE\_PERIOD\_SYMBOLS
  - core\_system\_parameters, [278](#)
- GALILEO\_CNAV\_SYMBOLS\_PER\_PAGE
  - core\_system\_parameters, [278](#)
- GALILEO\_CNAV\_TELEMETRY\_RATE\_BITS\_SECOND
  - core\_system\_parameters, [278](#)
- GALILEO\_DATA\_FRAME\_BITS
  - core\_system\_parameters, [278](#)
- GALILEO\_DATA\_FRAME\_BYTES
  - core\_system\_parameters, [279](#)
- GALILEO\_DATA\_JK\_BITS
  - core\_system\_parameters, [279](#)
- Galileo\_E1.h, [2445](#), [2446](#)
- GALILEO\_E1\_B\_CODE\_LENGTH\_CHIPS
  - core\_system\_parameters, [279](#)
- GALILEO\_E1\_B\_PRIMARY\_CODE
  - core\_system\_parameters, [279](#)
- GALILEO\_E1\_B\_PRIMARY\_CODE\_STR\_LENGTH
  - core\_system\_parameters, [279](#)
- GALILEO\_E1\_B\_SAMPLES\_PER\_SYMBOL
  - core\_system\_parameters, [279](#)
- GALILEO\_E1\_B\_SYMBOL\_RATE\_BPS
  - core\_system\_parameters, [279](#)
- GALILEO\_E1\_C\_PRIMARY\_CODE
  - core\_system\_parameters, [279](#)
- GALILEO\_E1\_C\_PRIMARY\_CODE\_STR\_LENGTH
  - core\_system\_parameters, [279](#)
- GALILEO\_E1\_C\_SECONDARY\_CODE
  - core\_system\_parameters, [279](#)
- GALILEO\_E1\_C\_SECONDARY\_CODE\_LENGTH
  - core\_system\_parameters, [279](#)
- GALILEO\_E1\_CODE\_CHIP\_RATE\_CPS
  - core\_system\_parameters, [280](#)
- galileo\_e1\_code\_gen\_complex\_sampled
  - algorithms\_libs, [100](#), [101](#)
- galileo\_e1\_code\_gen\_float\_sampled
  - algorithms\_libs, [101](#)
- galileo\_e1\_code\_gen\_sinoc11\_float
  - algorithms\_libs, [101](#)
- GALILEO\_E1\_CODE\_PERIOD\_MS
  - core\_system\_parameters, [280](#)
- GALILEO\_E1\_CODE\_PERIOD\_S
  - core\_system\_parameters, [280](#)
- galileo\_e1\_dll\_pll\_veml\_tracking.h, [2183](#)
- galileo\_e1\_dll\_pll\_veml\_tracking\_fpga.h, [2185](#)
- GALILEO\_E1\_FREQ\_HZ
  - core\_system\_parameters, [280](#)
- GALILEO\_E1\_HISTORY\_DEEP
  - core\_system\_parameters, [280](#)
- GALILEO\_E1\_NUMBER\_OF\_CODES
  - core\_system\_parameters, [280](#)
- GALILEO\_E1\_OPT\_ACQ\_FS\_SPS
  - core\_system\_parameters, [280](#)
- galileo\_e1\_pcps\_8ms\_ambiguous\_acquisition.h, [1544](#), [1545](#)
- galileo\_e1\_pcps\_ambiguous\_acquisition.h, [1547](#), [1548](#)
- galileo\_e1\_pcps\_ambiguous\_acquisition\_fpga.h, [1550](#), [1551](#)
- galileo\_e1\_pcps\_cccwsr\_ambiguous\_acquisition.h, [1554](#)
- galileo\_e1\_pcps\_quicksync\_ambiguous\_acquisition.h, [1556](#), [1557](#)
- galileo\_e1\_pcps\_tong\_ambiguous\_acquisition.h, [1559](#), [1560](#)
- galileo\_e1\_signal\_replica.h, [1734](#), [1735](#)
- GALILEO\_E1\_SUB\_CARRIER\_A\_RATE\_HZ
  - core\_system\_parameters, [280](#)
- GALILEO\_E1\_SUB\_CARRIER\_B\_RATE\_HZ
  - core\_system\_parameters, [280](#)
- galileo\_e1\_tcp\_connector\_tracking.h, [2187](#), [2188](#)
- Galileo\_E1\_Tcp\_Connector\_Tracking\_cc, [607](#)
- galileo\_e1\_tcp\_connector\_tracking\_cc.h, [2233](#), [2235](#)
- galileo\_e1\_tcp\_connector\_tracking\_cc\_sptr
  - tracking\_gr\_blocks, [197](#)
- galileo\_e1b\_telemetry\_decoder.h, [2116](#), [2117](#)
- galileo\_e5\_a\_code\_gen\_complex\_primary
  - algorithms\_libs, [102](#)
- galileo\_e5\_a\_code\_gen\_complex\_sampled
  - algorithms\_libs, [102](#)
- galileo\_e5\_b\_code\_gen\_complex\_primary
  - algorithms\_libs, [102](#)
- galileo\_e5\_b\_code\_gen\_complex\_sampled
  - algorithms\_libs, [102](#)

- galileo\_e5\_signal\_replica.h, 1736, 1737
- Galileo\_E5a.h, 2463, 2465
- GALILEO\_E5A\_CODE\_CHIP\_RATE\_CPS
  - core\_system\_parameters, 280
- GALILEO\_E5A\_CODE\_LENGTH\_CHIPS
  - core\_system\_parameters, 281
- GALILEO\_E5A\_CODE\_PERIOD\_MS
  - core\_system\_parameters, 281
- GALILEO\_E5A\_CODE\_PERIOD\_S
  - core\_system\_parameters, 281
- GALILEO\_E5A\_CRC\_ERROR\_LIMIT
  - core\_system\_parameters, 281
- galileo\_e5a\_dll\_pll\_tracking.h, 2189, 2190
- galileo\_e5a\_dll\_pll\_tracking\_fpga.h, 2191
- GALILEO\_E5A\_FREQ\_HZ
  - core\_system\_parameters, 281
- GALILEO\_E5A\_HISTORY\_DEEP
  - core\_system\_parameters, 281
- GALILEO\_E5A\_I\_PRIMARY\_CODE
  - core\_system\_parameters, 281
- GALILEO\_E5A\_I\_PRIMARY\_CODE\_STR\_LENGTH
  - core\_system\_parameters, 281
- GALILEO\_E5A\_I\_SECONDARY\_CODE
  - core\_system\_parameters, 281
- GALILEO\_E5A\_I\_SECONDARY\_CODE\_LENGTH
  - core\_system\_parameters, 281
- GALILEO\_E5A\_I\_TIERED\_CODE\_PERIOD\_S
  - core\_system\_parameters, 281
- galileo\_e5a\_noncoherent\_iq\_acquisition\_caf.h, 1562, 1563
- galileo\_e5a\_noncoherent\_iq\_acquisition\_caf\_cc.h, 1619, 1620
- galileo\_e5a\_noncoherentIQ\_acquisition\_caf\_cc, 608
  - ~galileo\_e5a\_noncoherentIQ\_acquisition\_caf\_cc, 611
  - general\_work, 611
  - init, 611
  - mag, 611
  - set\_active, 611
  - set\_channel, 612
  - set\_channel\_fsm, 612
  - set\_doppler\_max, 612
  - set\_doppler\_step, 612
  - set\_gnss\_synchro, 612
  - set\_local\_code, 614
  - set\_state, 614
  - set\_threshold, 614
- galileo\_e5a\_noncoherentIQ\_acquisition\_caf\_cc\_sptr
  - acquisition\_gr\_blocks, 79
- GALILEO\_E5A\_NUMBER\_OF\_CODES
  - core\_system\_parameters, 282
- GALILEO\_E5A\_OPT\_ACQ\_FS\_SPS
  - core\_system\_parameters, 282
- galileo\_e5a\_pcps\_acquisition.h, 1565, 1566
- galileo\_e5a\_pcps\_acquisition\_fpga.h, 1568
- GALILEO\_E5A\_Q\_PRIMARY\_CODE
  - core\_system\_parameters, 282
- GALILEO\_E5A\_Q\_PRIMARY\_CODE\_STR\_LENGTH
  - core\_system\_parameters, 282
- GALILEO\_E5A\_Q\_SECONDARY\_CODE
  - core\_system\_parameters, 282
- GALILEO\_E5A\_Q\_SECONDARY\_CODE\_LENGTH
  - core\_system\_parameters, 282
- GALILEO\_E5A\_Q\_SECONDARY\_CODE\_STR\_LENGTH
  - core\_system\_parameters, 282
- GALILEO\_E5A\_Q\_TIERED\_CODE\_PERIOD\_S
  - core\_system\_parameters, 282
- GALILEO\_E5A\_SYMBOL\_RATE\_BPS
  - core\_system\_parameters, 282
- galileo\_e5a\_telemetry\_decoder.h, 2118, 2119
- Galileo\_E5b.h, 2507, 2509
- GALILEO\_E5B\_CODE\_CHIP\_RATE\_CPS
  - core\_system\_parameters, 282
- GALILEO\_E5B\_CODE\_LENGTH\_CHIPS
  - core\_system\_parameters, 282
- GALILEO\_E5B\_CODE\_PERIOD\_MS
  - core\_system\_parameters, 283
- GALILEO\_E5B\_CODE\_PERIOD\_S
  - core\_system\_parameters, 283
- galileo\_e5b\_dll\_pll\_tracking.h, 2193, 2194
- GALILEO\_E5B\_FREQ\_HZ
  - core\_system\_parameters, 283
- GALILEO\_E5B\_HISTORY\_DEEP
  - core\_system\_parameters, 283
- GALILEO\_E5B\_I\_PRIMARY\_CODE
  - core\_system\_parameters, 283
- GALILEO\_E5B\_I\_PRIMARY\_CODE\_STR\_LENGTH
  - core\_system\_parameters, 283
- GALILEO\_E5B\_I\_SECONDARY\_CODE
  - core\_system\_parameters, 283
- GALILEO\_E5B\_I\_SECONDARY\_CODE\_LENGTH
  - core\_system\_parameters, 283
- GALILEO\_E5B\_I\_TIERED\_CODE\_PERIOD\_S
  - core\_system\_parameters, 283
- GALILEO\_E5B\_NUMBER\_OF\_CODES
  - core\_system\_parameters, 283
- GALILEO\_E5B\_OPT\_ACQ\_FS\_SPS
  - core\_system\_parameters, 283
- galileo\_e5b\_pcps\_acquisition.h, 1571, 1572
- galileo\_e5b\_pcps\_acquisition\_fpga.h, 1574, 1575
- GALILEO\_E5B\_Q\_PRIMARY\_CODE
  - core\_system\_parameters, 284
- GALILEO\_E5B\_Q\_PRIMARY\_CODE\_STR\_LENGTH
  - core\_system\_parameters, 284
- GALILEO\_E5B\_Q\_SECONDARY\_CODE
  - core\_system\_parameters, 284
- GALILEO\_E5B\_Q\_SECONDARY\_CODE\_LENGTH
  - core\_system\_parameters, 284
- GALILEO\_E5B\_Q\_SECONDARY\_CODE\_STR\_LENGTH
  - core\_system\_parameters, 284
- GALILEO\_E5B\_Q\_TIERED\_CODE\_PERIOD\_S
  - core\_system\_parameters, 284
- GALILEO\_E5B\_SYMBOL\_RATE\_BPS
  - core\_system\_parameters, 284
- galileo\_e5b\_telemetry\_decoder.h, 2120
- Galileo\_E6.h, 2551, 2553

- GALILEO\_E6\_B\_CODE\_CHIP\_RATE\_CPS
  - core\_system\_parameters, [284](#)
- galileo\_e6\_b\_code\_gen\_complex\_primary
  - algorithms\_libs, [102](#)
- galileo\_e6\_b\_code\_gen\_complex\_sampled
  - algorithms\_libs, [103](#)
- galileo\_e6\_b\_code\_gen\_float\_primary
  - algorithms\_libs, [103](#)
- GALILEO\_E6\_B\_CODE\_LENGTH\_CHIPS
  - core\_system\_parameters, [284](#)
- GALILEO\_E6\_B\_PRIMARY\_CODE
  - core\_system\_parameters, [284](#)
- GALILEO\_E6\_B\_PRIMARY\_CODE\_STR\_LENGTH
  - core\_system\_parameters, [284](#)
- GALILEO\_E6\_C\_CODE\_CHIP\_RATE\_CPS
  - core\_system\_parameters, [285](#)
- galileo\_e6\_c\_code\_gen\_complex\_primary
  - algorithms\_libs, [103](#)
- galileo\_e6\_c\_code\_gen\_complex\_sampled
  - algorithms\_libs, [103](#)
- galileo\_e6\_c\_code\_gen\_float\_primary
  - algorithms\_libs, [103](#)
- GALILEO\_E6\_C\_CODE\_LENGTH\_CHIPS
  - core\_system\_parameters, [285](#)
- GALILEO\_E6\_C\_PRIMARY\_CODE
  - core\_system\_parameters, [285](#)
- GALILEO\_E6\_C\_PRIMARY\_CODE\_STR\_LENGTH
  - core\_system\_parameters, [285](#)
- GALILEO\_E6\_C\_SECONDARY\_CODE
  - core\_system\_parameters, [285](#)
- galileo\_e6\_c\_secondary\_code
  - algorithms\_libs, [103](#)
- galileo\_e6\_c\_secondary\_code\_gen\_complex
  - algorithms\_libs, [104](#)
- galileo\_e6\_c\_secondary\_code\_gen\_float
  - algorithms\_libs, [104](#)
- GALILEO\_E6\_C\_SECONDARY\_CODE\_LENGTH\_CHIPSGALILEO\_FNAV\_INTERLEAVER\_ROWS
  - core\_system\_parameters, [285](#)
- GALILEO\_E6\_C\_SECONDARY\_CODE\_STR\_LENGTH
  - core\_system\_parameters, [285](#)
- GALILEO\_E6\_CODE\_PERIOD\_MS
  - core\_system\_parameters, [285](#)
- GALILEO\_E6\_CODE\_PERIOD\_S
  - core\_system\_parameters, [285](#)
- galileo\_e6\_dll\_pll\_tracking.h, [2195](#), [2196](#)
- GALILEO\_E6\_FREQ\_HZ
  - core\_system\_parameters, [285](#)
- galileo\_e6\_has\_msg\_receiver, [614](#)
  - ~galileo\_e6\_has\_msg\_receiver, [616](#)
  - process\_test\_page, [616](#)
- galileo\_e6\_has\_msg\_receiver.h, [2332](#), [2333](#)
- galileo\_e6\_has\_msg\_receiver\_sptr
  - core\_libs, [209](#)
- GALILEO\_E6\_NUMBER\_OF\_CODES
  - core\_system\_parameters, [285](#)
- GALILEO\_E6\_OPT\_ACQ\_FS\_SPS
  - core\_system\_parameters, [286](#)
- galileo\_e6\_pcps\_acquisition.h, [1578](#)
- galileo\_e6\_signal\_replica.h, [1738](#), [1739](#)
- galileo\_e6\_telemetry\_decoder.h, [2122](#)
- Galileo\_Ephemeris, [616](#)
  - BGD\_E1E5a, [621](#)
  - BGD\_E1E5b, [621](#)
  - E1B\_DVS, [622](#)
  - E1B\_HS, [622](#)
  - E5a\_DVS, [622](#)
  - E5a\_HS, [622](#)
  - E5b\_DVS, [622](#)
  - E5b\_HS, [622](#)
  - flag\_all\_ephemeris, [622](#)
  - Galileo\_Ephemeris, [621](#)
  - Galileo\_System\_Time, [621](#)
  - IOD\_ephemeris, [622](#)
  - IOD\_nav, [622](#)
  - serialize, [621](#)
  - SISA, [622](#)
- galileo\_ephemeris.h, [2575](#), [2576](#)
- galileo\_ephemeris\_map
  - Rtklib\_Solver, [1372](#)
- GALILEO\_F
  - core\_system\_parameters, [286](#)
- Galileo\_FNAV.h, [2577](#), [2581](#)
- GALILEO\_FNAV\_CODES\_PER\_PAGE
  - core\_system\_parameters, [286](#)
- GALILEO\_FNAV\_CODES\_PER\_PREAMBLE
  - core\_system\_parameters, [286](#)
- GALILEO\_FNAV\_CODES\_PER\_SYMBOL
  - core\_system\_parameters, [286](#)
- GALILEO\_FNAV\_DATA\_FRAME\_BITS
  - core\_system\_parameters, [286](#)
- GALILEO\_FNAV\_DATA\_FRAME\_BYTES
  - core\_system\_parameters, [286](#)
- GALILEO\_FNAV\_INTERLEAVER\_COLS
  - core\_system\_parameters, [286](#)
- GALILEO\_FNAV\_INTERLEAVER\_ROWS
  - core\_system\_parameters, [286](#)
- Galileo\_Fnav\_Message, [623](#)
  - get\_flag\_CRC\_test, [624](#)
  - get\_flag\_TOW\_set, [624](#)
  - get\_TOW1, [624](#)
  - get\_TOW2, [624](#)
  - get\_TOW3, [624](#)
  - get\_TOW4, [624](#)
  - is\_TOW1\_set, [624](#)
  - is\_TOW2\_set, [624](#)
  - is\_TOW3\_set, [625](#)
  - is\_TOW4\_set, [625](#)
  - set\_flag\_TOW\_set, [625](#)
  - set\_TOW1\_flag, [625](#)
  - set\_TOW2\_flag, [625](#)
  - set\_TOW3\_flag, [625](#)
  - set\_TOW4\_flag, [625](#)
- galileo\_fnav\_message.h, [2583](#), [2584](#)
- GALILEO\_FNAV\_PAGE\_TYPE\_BITS
  - core\_system\_parameters, [286](#)
- GALILEO\_FNAV\_PREAMBLE



- core\_system\_parameters, 286
- GALILEO\_FNAV\_PREAMBLE\_LENGTH\_BITS
  - core\_system\_parameters, 286
- GALILEO\_FNAV\_SECONDS\_PER\_PAGE
  - core\_system\_parameters, 287
- GALILEO\_FNAV\_SYMBOLS\_PER\_PAGE
  - core\_system\_parameters, 287
- GALILEO\_GM
  - core\_system\_parameters, 287
- Galileo\_HAS\_data, 625
  - cell\_mask, 632
  - cell\_mask\_availability\_flag, 632
  - code\_bias, 632
  - delta\_clock\_correction, 633
  - delta\_clock\_correction\_clock\_subset, 633
  - delta\_clock\_multiplier, 633
  - delta\_clock\_multiplier\_clock\_subset, 633
  - delta\_cross\_track, 633
  - delta\_in\_track, 633
  - delta\_radial, 633
  - get\_clock\_correction\_mult\_m, 629
  - get\_clock\_subset\_correction\_mult\_m, 629
  - get\_code\_bias\_m, 629
  - get\_delta\_clock\_correction\_m, 629
  - get\_delta\_clock\_subset\_correction\_m, 629, 630
  - get\_delta\_cross\_track\_m, 630
  - get\_delta\_in\_track\_m, 630
  - get\_delta\_radial\_m, 630
  - get\_gnss\_id, 631
  - get\_gnss\_iod, 631
  - get\_nsat, 631
  - get\_nsat\_sub, 631
  - get\_num\_satellites, 631
  - get\_num\_subset\_satellites, 631
  - get\_phase\_bias\_cycle, 631
  - get\_PRNs\_in\_mask, 631, 632
  - get\_PRNs\_in\_submask, 632
  - get\_signals\_in\_mask, 632
  - get\_systems\_string, 632
  - get\_systems\_subset\_string, 632
  - get\_validity\_interval\_s, 632
  - gnss\_id\_clock\_subset, 633
  - gnss\_id\_mask, 633
  - gnss\_iod, 633
  - has\_status, 634
  - header, 634
  - message\_id, 634
  - nav\_message, 634
  - Nsys, 634
  - Nsys\_sub, 634
  - phase\_bias, 634
  - phase\_discontinuity\_indicator, 634
  - satellite\_mask, 634
  - satellite\_submask, 634
  - signal\_mask, 635
  - tow, 635
  - validity\_interval\_index\_clock\_fullset\_corrections, 635
  - validity\_interval\_index\_clock\_subset\_corrections, 635
  - validity\_interval\_index\_code\_bias\_corrections, 635
  - validity\_interval\_index\_orbit\_corrections, 635
  - validity\_interval\_index\_phase\_bias\_corrections, 635
- galileo\_has\_data.h, 2588, 2589
- Galileo\_HAS\_page, 635
  - has\_message\_string, 637
  - has\_status, 637
  - message\_id, 637
  - message\_page\_id, 637
  - message\_size, 637
  - message\_type, 637
  - reserved, 637
  - time\_stamp, 637
  - tow, 637
- galileo\_has\_page.h, 2591, 2592
- Galileo\_INAV.h, 2593, 2598
- GALILEO\_INAV\_ENCODED\_SSP1
  - core\_system\_parameters, 287
- GALILEO\_INAV\_ENCODED\_SSP2
  - core\_system\_parameters, 287
- GALILEO\_INAV\_ENCODED\_SSP3
  - core\_system\_parameters, 287
- GALILEO\_INAV\_INTERLEAVER\_COLS
  - core\_system\_parameters, 287
- GALILEO\_INAV\_INTERLEAVER\_ROWS
  - core\_system\_parameters, 287
- Galileo\_Inav\_Message, 638
  - enable\_reed\_solomon, 639
  - get\_A0G, 639
  - get\_A1G, 639
  - get\_flag\_CRC\_test, 639
  - get\_flag\_GGTO, 639
  - get\_flag\_TOW\_set, 640
  - get\_Galileo\_week, 640
  - get\_t0G, 640
  - get\_TOW0, 640
  - get\_TOW5, 640
  - get\_TOW6, 640
  - get\_WN0G, 640
  - init\_PRN, 640
  - is\_TOW0\_set, 640
  - is\_TOW5\_set, 640
  - is\_TOW6\_set, 640
  - set\_flag\_TOW\_set, 640
  - set\_TOW0\_flag, 641
  - set\_TOW5\_flag, 641
  - set\_TOW6\_flag, 641
- galileo\_inav\_message.h, 2602, 2603
- GALILEO\_INAV\_PAGE\_PART\_MS
  - core\_system\_parameters, 287
- GALILEO\_INAV\_PAGE\_PART\_SECONDS
  - core\_system\_parameters, 287
- GALILEO\_INAV\_PAGE\_PART\_SYMBOLS
  - core\_system\_parameters, 287
- GALILEO\_INAV\_PAGE\_PART\_WITH\_PREAMBLE\_SECONDS

- core\_system\_parameters, [288](#)
- GALILEO\_INAV\_PAGE\_SECONDS
  - core\_system\_parameters, [288](#)
- GALILEO\_INAV\_PAGE\_SYMBOLS
  - core\_system\_parameters, [288](#)
- GALILEO\_INAV\_PLAIN\_SSP1
  - core\_system\_parameters, [288](#)
- GALILEO\_INAV\_PLAIN\_SSP2
  - core\_system\_parameters, [288](#)
- GALILEO\_INAV\_PLAIN\_SSP3
  - core\_system\_parameters, [288](#)
- GALILEO\_INAV\_PREAMBLE
  - core\_system\_parameters, [288](#)
- GALILEO\_INAV\_PREAMBLE\_LENGTH\_BITS
  - core\_system\_parameters, [288](#)
- GALILEO\_INAV\_PREAMBLE\_PERIOD\_SYMBOLS
  - core\_system\_parameters, [288](#)
- Galileo\_Iono, [641](#)
  - ai0, [643](#)
  - ai1, [643](#)
  - ai2, [643](#)
  - Galileo\_Iono, [642](#)
  - Region1\_flag, [643](#)
  - Region2\_flag, [643](#)
  - Region3\_flag, [643](#)
  - Region4\_flag, [643](#)
  - Region5\_flag, [643](#)
  - serialize, [642](#)
  - tow, [643](#)
  - WN, [644](#)
- galileo\_iono
  - Rtklib\_Solver, [1373](#)
- galileo\_iono.h, [2609](#), [2610](#)
- Galileo\_ISM, [644](#)
  - Galileo\_ISM, [645](#)
- galileo\_ism.h, [2611](#), [2612](#)
- GALILEO\_ISM\_CRC\_DATA\_BITS
  - core\_system\_parameters, [288](#)
- GALILEO\_ISM\_CRC\_DATA\_BYTES
  - core\_system\_parameters, [288](#)
- Galileo\_OSNMA.h, [2615](#), [2616](#)
- GALILEO\_PAGE\_TYPE\_BITS
  - core\_system\_parameters, [289](#)
- galileo\_pcps\_8ms\_acquisition\_cc, [645](#)
  - ~galileo\_pcps\_8ms\_acquisition\_cc, [648](#)
  - general\_work, [648](#)
  - init, [648](#)
  - mag, [648](#)
  - set\_active, [648](#)
  - set\_channel, [649](#)
  - set\_channel\_fsm, [649](#)
  - set\_doppler\_max, [649](#)
  - set\_doppler\_step, [649](#)
  - set\_gnss\_synchro, [649](#)
  - set\_local\_code, [649](#)
  - set\_state, [650](#)
  - set\_threshold, [650](#)
- galileo\_pcps\_8ms\_acquisition\_cc.h, [1623](#), [1625](#)
- galileo\_pcps\_8ms\_acquisition\_cc\_sptr
  - acquisition\_gr\_blocks, [79](#)
- Galileo\_Reduced\_CED, [650](#)
  - af0red, [652](#)
  - af1red, [652](#)
  - compute\_eph, [652](#)
  - DeltaAred, [652](#)
  - Deltai0red, [652](#)
  - exred, [652](#)
  - eyred, [652](#)
  - Galileo\_Reduced\_CED, [652](#)
  - IODnav, [652](#)
  - lambda0red, [653](#)
  - Omega0red, [653](#)
  - PRN, [653](#)
  - TOTRedCED, [653](#)
- galileo\_reduced\_ced.h, [2618](#), [2619](#)
- galileo\_satellites
  - sdr\_gnss\_packet\_t, [1408](#)
- Galileo\_System\_Time
  - Galileo\_Ephemeris, [621](#)
- galileo\_telemetry\_decoder\_gs, [653](#)
  - general\_work, [655](#)
  - set\_channel, [655](#)
  - set\_satellite, [655](#)
- galileo\_telemetry\_decoder\_gs.h, [2140](#), [2141](#)
- galileo\_telemetry\_decoder\_gs\_sptr
  - telemetry\_decoder\_gr\_blocks, [188](#)
- GALILEO\_TELEMETRY\_RATE\_BITS\_SECOND
  - core\_system\_parameters, [289](#)
- galileo\_tow\_map, [656](#)
  - ~galileo\_tow\_map, [657](#)
- galileo\_tow\_map.h, [2335](#), [2336](#)
- galileo\_tow\_map\_sptr
  - core\_libs, [209](#)
- Galileo\_Utc\_Model, [657](#)
  - A0, [658](#)
  - A1, [658](#)
  - A\_0G, [659](#)
  - A\_1G, [659](#)
  - Delta\_tLS, [659](#)
  - Delta\_tLSF, [659](#)
  - DN, [659](#)
  - flag\_utc\_model, [659](#)
  - Galileo\_Utc\_Model, [658](#)
  - GST\_to\_UTC\_time, [658](#)
  - serialize, [658](#)
  - t\_0G, [659](#)
  - tot, [659](#)
  - WN\_0G, [659](#)
  - WN\_LSF, [659](#)
  - WNot, [659](#)
- galileo\_utc\_model
  - Rtklib\_Solver, [1373](#)
- galileo\_utc\_model.h, [2620](#), [2621](#)
- GalileoE1BTelemetryDecoder, [660](#)
  - connect, [663](#)
  - disconnect, [663](#)



- get\_left\_block, [663](#)
- get\_right\_block, [663](#)
- implementation, [663](#)
- item\_size, [663](#)
- reset, [663](#)
- role, [664](#)
- set\_channel, [664](#)
- set\_satellite, [664](#)
- GalileoE1DlIPllVemlTracking, [664](#)
  - connect, [667](#)
  - disconnect, [667](#)
  - get\_left\_block, [667](#)
  - get\_right\_block, [667](#)
  - implementation, [667](#)
  - item\_size, [667](#)
  - role, [668](#)
  - set\_channel, [668](#)
  - set\_gnss\_synchro, [668](#)
  - start\_tracking, [668](#)
  - stop\_tracking, [668](#)
- GalileoE1DlIPllVemlTrackingFpga, [668](#)
  - ~GalileoE1DlIPllVemlTrackingFpga, [672](#)
  - connect, [672](#)
  - disconnect, [672](#)
  - GalileoE1DlIPllVemlTrackingFpga, [671](#)
  - get\_left\_block, [672](#)
  - get\_right\_block, [672](#)
  - implementation, [672](#)
  - item\_size, [672](#)
  - role, [672](#)
  - set\_channel, [672](#)
  - set\_gnss\_synchro, [672](#)
  - start\_tracking, [673](#)
  - stop\_tracking, [673](#)
- GalileoE1Pcps8msAmbiguousAcquisition, [673](#)
  - connect, [676](#)
  - disconnect, [676](#)
  - get\_left\_block, [677](#)
  - get\_right\_block, [677](#)
  - implementation, [677](#)
  - init, [677](#)
  - item\_size, [677](#)
  - mag, [677](#)
  - reset, [677](#)
  - role, [677](#)
  - set\_channel, [677](#)
  - set\_channel\_fsm, [678](#)
  - set\_doppler\_max, [678](#)
  - set\_doppler\_step, [678](#)
  - set\_gnss\_synchro, [678](#)
  - set\_local\_code, [678](#)
  - set\_resampler\_latency, [678](#)
  - set\_state, [678](#)
  - set\_threshold, [678](#)
  - stop\_acquisition, [679](#)
- GalileoE1PcpsAmbiguousAcquisition, [679](#)
  - connect, [683](#)
  - disconnect, [683](#)
- get\_left\_block, [683](#)
- get\_right\_block, [683](#)
- implementation, [683](#)
- init, [683](#)
- item\_size, [683](#)
- mag, [683](#)
- reset, [683](#)
- role, [683](#)
- set\_channel, [683](#)
- set\_channel\_fsm, [684](#)
- set\_doppler\_center, [684](#)
- set\_doppler\_max, [684](#)
- set\_doppler\_step, [684](#)
- set\_gnss\_synchro, [684](#)
- set\_local\_code, [684](#)
- set\_resampler\_latency, [684](#)
- set\_state, [684](#)
- set\_threshold, [685](#)
- stop\_acquisition, [685](#)
- GalileoE1PcpsAmbiguousAcquisitionFpga, [685](#)
  - ~GalileoE1PcpsAmbiguousAcquisitionFpga, [689](#)
  - connect, [689](#)
  - disconnect, [689](#)
  - GalileoE1PcpsAmbiguousAcquisitionFpga, [689](#)
  - get\_left\_block, [689](#)
  - get\_right\_block, [689](#)
  - implementation, [689](#)
  - init, [690](#)
  - item\_size, [690](#)
  - mag, [690](#)
  - reset, [690](#)
  - role, [690](#)
  - set\_channel, [690](#)
  - set\_channel\_fsm, [690](#)
  - set\_doppler\_center, [690](#)
  - set\_doppler\_max, [691](#)
  - set\_doppler\_step, [691](#)
  - set\_gnss\_synchro, [691](#)
  - set\_local\_code, [691](#)
  - set\_resampler\_latency, [691](#)
  - set\_state, [691](#)
  - set\_threshold, [691](#)
  - stop\_acquisition, [691](#)
- GalileoE1PcpsCccwsrAmbiguousAcquisition, [692](#)
  - connect, [694](#)
  - disconnect, [694](#)
  - get\_left\_block, [695](#)
  - get\_right\_block, [695](#)
  - implementation, [695](#)
  - init, [695](#)
  - item\_size, [695](#)
  - mag, [695](#)
  - reset, [695](#)
  - role, [695](#)
  - set\_channel, [695](#)
  - set\_channel\_fsm, [696](#)
  - set\_doppler\_max, [696](#)
  - set\_doppler\_step, [696](#)

- set\_gnss\_synchro, 696
  - set\_local\_code, 696
  - set\_resampler\_latency, 696
  - set\_state, 696
  - set\_threshold, 696
  - stop\_acquisition, 697
- GalileoE1PcpsQuickSyncAmbiguousAcquisition, 697
  - connect, 700
  - disconnect, 700
  - get\_left\_block, 701
  - get\_right\_block, 701
  - implementation, 701
  - init, 701
  - item\_size, 701
  - mag, 701
  - reset, 701
  - role, 701
  - set\_channel, 701
  - set\_channel\_fsm, 702
  - set\_doppler\_max, 702
  - set\_doppler\_step, 702
  - set\_gnss\_synchro, 702
  - set\_local\_code, 702
  - set\_resampler\_latency, 702
  - set\_state, 702
  - set\_threshold, 702
  - stop\_acquisition, 703
- GalileoE1PcpsTongAmbiguousAcquisition, 703
  - connect, 706
  - disconnect, 706
  - get\_left\_block, 707
  - get\_right\_block, 707
  - implementation, 707
  - init, 707
  - item\_size, 707
  - mag, 707
  - reset, 707
  - role, 707
  - set\_channel, 707
  - set\_channel\_fsm, 708
  - set\_doppler\_max, 708
  - set\_doppler\_step, 708
  - set\_gnss\_synchro, 708
  - set\_local\_code, 708
  - set\_resampler\_latency, 708
  - set\_state, 708
  - set\_threshold, 708
  - stop\_acquisition, 709
- GalileoE1TcpConnectorTracking, 709
  - connect, 712
  - disconnect, 712
  - get\_left\_block, 712
  - get\_right\_block, 712
  - implementation, 712
  - item\_size, 712
  - role, 713
  - set\_channel, 713
  - set\_gnss\_synchro, 713
  - start\_tracking, 713
  - stop\_tracking, 713
- GalileoE5aDIIPIITracking, 713
  - connect, 716
  - disconnect, 716
  - get\_left\_block, 716
  - get\_right\_block, 716
  - implementation, 716
  - item\_size, 716
  - role, 717
  - set\_channel, 717
  - set\_gnss\_synchro, 717
  - start\_tracking, 717
  - stop\_tracking, 717
- GalileoE5aDIIPIITrackingFpga, 717
  - ~GalileoE5aDIIPIITrackingFpga, 720
  - connect, 721
  - disconnect, 721
  - GalileoE5aDIIPIITrackingFpga, 720
  - get\_left\_block, 721
  - get\_right\_block, 721
  - implementation, 721
  - item\_size, 721
  - role, 721
  - set\_channel, 721
  - set\_gnss\_synchro, 721
  - start\_tracking, 722
  - stop\_tracking, 722
- GalileoE5aNoncoherentIQAquisitionCaf, 723
  - connect, 725
  - disconnect, 725
  - get\_left\_block, 726
  - get\_right\_block, 726
  - implementation, 726
  - init, 726
  - item\_size, 726
  - mag, 726
  - reset, 726
  - role, 726
  - set\_channel, 726
  - set\_channel\_fsm, 727
  - set\_doppler\_max, 727
  - set\_doppler\_step, 727
  - set\_gnss\_synchro, 727
  - set\_local\_code, 727
  - set\_resampler\_latency, 727
  - set\_state, 727
  - set\_threshold, 728
  - stop\_acquisition, 728
- GalileoE5aPcpsAcquisition, 729
  - connect, 731
  - disconnect, 731
  - get\_left\_block, 732
  - get\_right\_block, 732
  - implementation, 732
  - init, 732
  - item\_size, 732
  - mag, 732

- reset, [732](#)
- role, [732](#)
- set\_channel, [732](#)
- set\_channel\_fsm, [732](#)
- set\_doppler\_center, [733](#)
- set\_doppler\_max, [733](#)
- set\_doppler\_step, [733](#)
- set\_gnss\_synchro, [733](#)
- set\_local\_code, [733](#)
- set\_resampler\_latency, [733](#)
- set\_state, [733](#)
- set\_threshold, [733](#)
- stop\_acquisition, [734](#)
- GalileoE5aPcpsAcquisitionFpga, [734](#)
  - ~GalileoE5aPcpsAcquisitionFpga, [738](#)
  - connect, [738](#)
  - disconnect, [738](#)
  - GalileoE5aPcpsAcquisitionFpga, [738](#)
  - get\_left\_block, [738](#)
  - get\_right\_block, [739](#)
  - implementation, [739](#)
  - init, [739](#)
  - item\_size, [739](#)
  - mag, [739](#)
  - reset, [739](#)
  - role, [739](#)
  - set\_channel, [739](#)
  - set\_channel\_fsm, [739](#)
  - set\_doppler\_center, [740](#)
  - set\_doppler\_max, [740](#)
  - set\_doppler\_step, [740](#)
  - set\_gnss\_synchro, [740](#)
  - set\_local\_code, [740](#)
  - set\_resampler\_latency, [740](#)
  - set\_single\_doppler\_flag, [740](#)
  - set\_state, [740](#)
  - set\_threshold, [741](#)
  - stop\_acquisition, [741](#)
- GalileoE5aTelemetryDecoder, [741](#)
  - connect, [744](#)
  - disconnect, [744](#)
  - get\_left\_block, [744](#)
  - get\_right\_block, [744](#)
  - implementation, [744](#)
  - item\_size, [744](#)
  - reset, [744](#)
  - role, [745](#)
  - set\_channel, [745](#)
  - set\_satellite, [745](#)
- GalileoE5bDIPIITracking, [745](#)
  - connect, [748](#)
  - disconnect, [748](#)
  - get\_left\_block, [748](#)
  - get\_right\_block, [748](#)
  - implementation, [748](#)
  - item\_size, [749](#)
  - role, [749](#)
  - set\_channel, [749](#)
  - set\_gnss\_synchro, [749](#)
  - start\_tracking, [749](#)
  - stop\_tracking, [749](#)
- GalileoE5bPcpsAcquisition, [750](#)
  - ~GalileoE5bPcpsAcquisition, [753](#)
  - connect, [753](#)
  - disconnect, [753](#)
  - GalileoE5bPcpsAcquisition, [753](#)
  - get\_left\_block, [753](#)
  - get\_right\_block, [753](#)
  - implementation, [753](#)
  - init, [754](#)
  - item\_size, [754](#)
  - mag, [754](#)
  - reset, [754](#)
  - role, [754](#)
  - set\_channel, [754](#)
  - set\_channel\_fsm, [754](#)
  - set\_doppler\_center, [754](#)
  - set\_doppler\_max, [755](#)
  - set\_doppler\_step, [755](#)
  - set\_gnss\_synchro, [755](#)
  - set\_local\_code, [755](#)
  - set\_resampler\_latency, [755](#)
  - set\_state, [755](#)
  - set\_threshold, [755](#)
  - stop\_acquisition, [755](#)
- GalileoE5bPcpsAcquisitionFpga, [756](#)
  - ~GalileoE5bPcpsAcquisitionFpga, [760](#)
  - connect, [760](#)
  - disconnect, [760](#)
  - GalileoE5bPcpsAcquisitionFpga, [760](#)
  - get\_left\_block, [760](#)
  - get\_right\_block, [761](#)
  - implementation, [761](#)
  - init, [761](#)
  - item\_size, [761](#)
  - mag, [761](#)
  - reset, [761](#)
  - role, [761](#)
  - set\_channel, [761](#)
  - set\_channel\_fsm, [761](#)
  - set\_doppler\_center, [762](#)
  - set\_doppler\_max, [762](#)
  - set\_doppler\_step, [762](#)
  - set\_gnss\_synchro, [762](#)
  - set\_local\_code, [762](#)
  - set\_resampler\_latency, [762](#)
  - set\_single\_doppler\_flag, [762](#)
  - set\_state, [762](#)
  - set\_threshold, [763](#)
  - stop\_acquisition, [763](#)
- GalileoE5bTelemetryDecoder, [763](#)
  - connect, [766](#)
  - disconnect, [766](#)
  - get\_left\_block, [766](#)
  - get\_right\_block, [766](#)
  - implementation, [766](#)

- item\_size, [766](#)
  - reset, [767](#)
  - role, [767](#)
  - set\_channel, [767](#)
  - set\_satellite, [767](#)
- GalileoE6DIIPIITracking, [767](#)
  - connect, [770](#)
  - disconnect, [770](#)
  - get\_left\_block, [770](#)
  - get\_right\_block, [770](#)
  - implementation, [770](#)
  - item\_size, [771](#)
  - role, [771](#)
  - set\_channel, [771](#)
  - set\_gnss\_synchro, [771](#)
  - start\_tracking, [771](#)
  - stop\_tracking, [771](#)
- GalileoE6PcpsAcquisition, [771](#)
  - connect, [775](#)
  - disconnect, [775](#)
  - get\_left\_block, [775](#)
  - get\_right\_block, [775](#)
  - implementation, [775](#)
  - init, [775](#)
  - item\_size, [775](#)
  - mag, [775](#)
  - reset, [775](#)
  - role, [775](#)
  - set\_channel, [775](#)
  - set\_channel\_fsm, [776](#)
  - set\_doppler\_center, [776](#)
  - set\_doppler\_max, [776](#)
  - set\_doppler\_step, [776](#)
  - set\_gnss\_synchro, [776](#)
  - set\_local\_code, [776](#)
  - set\_resampler\_latency, [776](#)
  - set\_state, [776](#)
  - set\_threshold, [777](#)
  - stop\_acquisition, [777](#)
- GalileoE6TelemetryDecoder, [777](#)
  - connect, [780](#)
  - disconnect, [780](#)
  - get\_left\_block, [780](#)
  - get\_right\_block, [780](#)
  - implementation, [780](#)
  - item\_size, [780](#)
  - reset, [781](#)
  - role, [781](#)
  - set\_channel, [781](#)
  - set\_satellite, [781](#)
- gamn
  - geph\_t, [789](#)
- GAP\_RESION
  - algorithms\_libs\_rtklib, [144](#)
- gdop
  - Monitor\_Pvt, [1173](#)
- gen\_signal\_source.h, [2008](#), [2009](#)
- general\_work
  - beidou\_b1i\_telemetry\_decoder\_gs, [371](#)
  - beidou\_b3i\_telemetry\_decoder\_gs, [373](#)
  - dll\_pll\_veml\_tracking\_fpga, [508](#)
  - galileo\_e5a\_noncoherentIQ\_acquisition\_caf\_cc, [611](#)
  - galileo\_pcps\_8ms\_acquisition\_cc, [648](#)
  - galileo\_telemetry\_decoder\_gs, [655](#)
  - glonass\_l1\_ca\_telemetry\_decoder\_gs, [818](#)
  - glonass\_l2\_ca\_telemetry\_decoder\_gs, [825](#)
  - gps\_l1\_ca\_telemetry\_decoder\_gs, [953](#)
  - gps\_l2c\_telemetry\_decoder\_gs, [955](#)
  - pcps\_acquisition, [1244](#)
  - pcps\_acquisition\_fine\_doppler\_cc, [1249](#)
  - pcps\_assisted\_acquisition\_cc, [1259](#)
  - pcps\_cccwsr\_acquisition\_cc, [1264](#)
  - pcps\_openc1\_acquisition\_cc, [1269](#)
  - pcps\_quicksync\_acquisition\_cc, [1275](#)
  - pcps\_tong\_acquisition\_cc, [1281](#)
  - sbas\_l1\_telemetry\_decoder\_gs, [1389](#)
- GenSignalSource, [781](#)
  - connect, [784](#)
  - disconnect, [784](#)
  - GenSignalSource, [784](#)
  - get\_left\_block, [784](#)
  - get\_right\_block, [784](#)
  - getRfChannels, [784](#)
  - implementation, [784](#)
  - item\_size, [785](#)
  - role, [785](#)
  - signal\_generator, [785](#)
- Geo\_to\_ECEF
  - algorithms\_libs, [104](#)
- geofunctions.h, [1740](#), [1742](#)
- Geohash, [785](#)
  - decode, [786](#)
  - encode, [786](#)
- geohash
  - Monitor\_Pvt, [1173](#)
- geohash.h, [1893](#)
- geoid
  - solopt\_t, [1443](#)
- geojson\_output\_enabled
  - Pvt\_Conf, [1310](#)
- geojson\_output\_path
  - Pvt\_Conf, [1310](#)
- GeoJSON\_Printer, [786](#)
- geojson\_printer.h, [1894](#), [1895](#)
- geojson\_rate\_ms
  - Pvt\_Conf, [1310](#)
- geph
  - nav\_t, [1192](#)
- geph\_t, [788](#)
  - acc, [789](#)
  - age, [789](#)
  - dtaun, [789](#)
  - frq, [789](#)
  - gamn, [789](#)
  - iode, [789](#)

- pos, [789](#)
- sat, [789](#)
- sva, [789](#)
- svh, [789](#)
- taun, [789](#)
- toe, [789](#)
- tof, [790](#)
- vel, [790](#)
- Get
  - INIReader, [1108](#)
- get
  - algorithms\_libs, [104](#)
- get\_A0G
  - Galileo\_Inav\_Message, [639](#)
- get\_A1G
  - Galileo\_Inav\_Message, [639](#)
- get\_alm\_satellite\_slot\_number
  - Glionass\_Gnav\_Navigation\_Message, [806](#)
- get\_almanac
  - Glionass\_Gnav\_Navigation\_Message, [806](#)
  - Gps\_Navigation\_Message, [959](#)
- get\_beidou\_dnav\_almanac\_map
  - rtklib\_pvt\_gs, [1367](#)
- get\_beidou\_dnav\_ephemeris\_map
  - rtklib\_pvt\_gs, [1367](#)
- get\_block
  - Gnss\_Satellite, [880](#)
- get\_buffer\_address
  - Fpga\_DMA, [582](#)
- get\_clock\_correction\_mult\_m
  - Galileo\_HAS\_data, [629](#)
- get\_clock\_drift\_ppm
  - Pvt\_Solution, [1318](#)
- get\_clock\_subset\_correction\_mult\_m
  - Galileo\_HAS\_data, [629](#)
- get\_code\_bias\_m
  - Galileo\_HAS\_data, [629](#)
- get\_code\_nco
  - Tracking\_2nd\_DLL\_filter, [1494](#)
- get\_course\_over\_ground
  - Pvt\_Solution, [1318](#)
- get\_current\_status\_map
  - channel\_status\_msg\_receiver, [434](#)
- get\_current\_status\_pvt
  - channel\_status\_msg\_receiver, [435](#)
- get\_delta\_clock\_correction\_m
  - Galileo\_HAS\_data, [629](#)
- get\_delta\_clock\_subset\_correction\_m
  - Galileo\_HAS\_data, [629](#), [630](#)
- get\_delta\_cross\_track\_m
  - Galileo\_HAS\_data, [630](#)
- get\_delta\_in\_track\_m
  - Galileo\_HAS\_data, [630](#)
- get\_delta\_radial\_m
  - Galileo\_HAS\_data, [630](#)
- get\_ephemeris
  - Beidou\_Dnav\_Navigation\_Message, [389](#)
  - Glionass\_Gnav\_Navigation\_Message, [806](#)
  - Gps\_CNAV\_Navigation\_Message, [933](#)
  - Gps\_Navigation\_Message, [959](#)
- get\_flag\_CRC\_test
  - Beidou\_Dnav\_Navigation\_Message, [389](#)
  - Galileo\_Cnav\_Message, [606](#)
  - Galileo\_Fnav\_Message, [624](#)
  - Galileo\_Inav\_Message, [639](#)
  - Glionass\_Gnav\_Navigation\_Message, [807](#)
- get\_flag\_GGTO
  - Galileo\_Inav\_Message, [639](#)
- get\_flag\_iono\_valid
  - Gps\_Navigation\_Message, [959](#)
- get\_flag\_new\_SOW\_available
  - Beidou\_Dnav\_Navigation\_Message, [389](#)
- get\_flag\_TOW\_new
  - Glionass\_Gnav\_Navigation\_Message, [807](#)
- get\_flag\_TOW\_set
  - Galileo\_Fnav\_Message, [624](#)
  - Galileo\_Inav\_Message, [640](#)
- get\_flag\_update\_slot\_number
  - Glionass\_Gnav\_Navigation\_Message, [807](#)
- get\_flag\_utc\_model\_valid
  - Gps\_Navigation\_Message, [959](#)
- get\_frame\_number
  - Glionass\_Gnav\_Navigation\_Message, [807](#)
- get\_galileo\_almanac
  - Rtklib\_Pvt, [1363](#)
- get\_galileo\_almanac\_map
  - rtklib\_pvt\_gs, [1367](#)
- get\_galileo\_ephemeris
  - Rtklib\_Pvt, [1363](#)
- get\_galileo\_ephemeris\_map
  - rtklib\_pvt\_gs, [1367](#)
- get\_Galileo\_week
  - Galileo\_Inav\_Message, [640](#)
- get\_gdop
  - Rtklib\_Solver, [1372](#)
- get\_gnss\_id
  - Galileo\_HAS\_data, [631](#)
- get\_gnss\_iod
  - Galileo\_HAS\_data, [631](#)
- get\_gps\_almanac
  - Rtklib\_Pvt, [1363](#)
- get\_gps\_almanac\_map
  - rtklib\_pvt\_gs, [1367](#)
- get\_gps\_ephemeris
  - Rtklib\_Pvt, [1364](#)
- get\_gps\_ephemeris\_map
  - rtklib\_pvt\_gs, [1367](#)
- get\_GPS\_week
  - Gps\_Navigation\_Message, [959](#)
- get\_HAS\_encoded\_page
  - Galileo\_Cnav\_Message, [606](#)
- get\_hdop
  - Rtklib\_Solver, [1372](#)
- get\_height
  - Pvt\_Solution, [1318](#)
- get\_IOD\_nav

- OSNMA\_NavData, [1235](#)
- get\_iono
  - Beidou\_Dnav\_Navigation\_Message, [389](#)
  - Gps\_CNAV\_Navigation\_Message, [933](#)
  - Gps\_Navigation\_Message, [959](#)
- get\_last\_received\_TOW
  - OSNMA\_NavData, [1235](#)
- get\_latest\_PVT
  - Rtklib\_Pvt, [1364](#)
  - rtklib\_pvt\_gs, [1367](#)
- get\_latitude
  - Pvt\_Solution, [1318](#)
- get\_left\_block
  - Ad936xCustomSignalSource, [344](#)
  - Adrv9361z7035SignalSourceFPGA, [348](#)
  - ArraySignalConditioner, [362](#)
  - BeamformerFilter, [368](#)
  - BeidouB1iDIIPIITracking, [396](#)
  - BeidouB1iPcpsAcquisition, [401](#)
  - BeidouB1iTelemetryDecoder, [406](#)
  - BeidouB3iDIIPIITracking, [410](#)
  - BeidouB3iPcpsAcquisition, [415](#)
  - BeidouB3iTelemetryDecoder, [420](#)
  - ByteToShort, [425](#)
  - Channel, [429](#)
  - ChannelInterface, [439](#)
  - CshortToGrComplex, [475](#)
  - CustomUDPSignalSource, [480](#)
  - DirectResamplerConditioner, [491](#)
  - DMASignalSourceFPGA, [511](#)
  - FifoSignalSource, [535](#)
  - FileSourceBase, [552](#)
  - FirFilter, [560](#)
  - FlexibandSignalSource, [564](#)
  - Fmcomms2SignalSource, [568](#)
  - Fmcomms5SignalSourceFPGA, [571](#)
  - FreqXlatingFirFilter, [593](#)
  - GalileoE1BTelemetryDecoder, [663](#)
  - GalileoE1DIIPIIVemITracking, [667](#)
  - GalileoE1DIIPIIVemITrackingFpga, [672](#)
  - GalileoE1Pcps8msAmbiguousAcquisition, [677](#)
  - GalileoE1PcpsAmbiguousAcquisition, [683](#)
  - GalileoE1PcpsAmbiguousAcquisitionFpga, [689](#)
  - GalileoE1PcpsCccwsrAmbiguousAcquisition, [695](#)
  - GalileoE1PcpsQuickSyncAmbiguousAcquisition, [701](#)
  - GalileoE1PcpsTongAmbiguousAcquisition, [707](#)
  - GalileoE1TcpConnectorTracking, [712](#)
  - GalileoE5aDIIPIITracking, [716](#)
  - GalileoE5aDIIPIITrackingFpga, [721](#)
  - GalileoE5aNoncoherentIQAcquisitionCaf, [726](#)
  - GalileoE5aPcpsAcquisition, [732](#)
  - GalileoE5aPcpsAcquisitionFpga, [738](#)
  - GalileoE5aTelemetryDecoder, [744](#)
  - GalileoE5bDIIPIITracking, [748](#)
  - GalileoE5bPcpsAcquisition, [753](#)
  - GalileoE5bPcpsAcquisitionFpga, [760](#)
  - GalileoE5bTelemetryDecoder, [766](#)
  - GalileoE6DIIPIITracking, [770](#)
  - GalileoE6PcpsAcquisition, [775](#)
  - GalileoE6TelemetryDecoder, [780](#)
  - GenSignalSource, [784](#)
  - GlonassL1CaDIIPIICAidTracking, [828](#)
  - GlonassL1CaDIIPIITracking, [832](#)
  - GlonassL1CaPcpsAcquisition, [837](#)
  - GlonassL1CaTelemetryDecoder, [841](#)
  - GlonassL2CaDIIPIICAidTracking, [845](#)
  - GlonassL2CaDIIPIITracking, [849](#)
  - GlonassL2CaPcpsAcquisition, [854](#)
  - GlonassL2CaTelemetryDecoder, [858](#)
  - GNSSBlockInterface, [910](#)
  - GpsL1CaDIIPIITracking, [967](#)
  - GpsL1CaDIIPIITrackingFpga, [972](#)
  - GpsL1CaDIIPIITrackingGPU, [976](#)
  - GpsL1CaGaussianTracking, [980](#)
  - GpsL1CaKfTracking, [984](#)
  - GpsL1CaPcpsAcquisition, [989](#)
  - GpsL1CaPcpsAcquisitionFineDoppler, [995](#)
  - GpsL1CaPcpsAcquisitionFpga, [1000](#)
  - GpsL1CaPcpsAssistedAcquisition, [1007](#)
  - GpsL1CaPcpsOpenCIAcquisition, [1012](#)
  - GpsL1CaPcpsQuickSyncAcquisition, [1018](#)
  - GpsL1CaPcpsTongAcquisition, [1023](#)
  - GpsL1CaTcpConnectorTracking, [1027](#)
  - GpsL1CaTelemetryDecoder, [1031](#)
  - GpsL2CTelemetryDecoder, [1035](#)
  - GpsL2MDIIPIITracking, [1039](#)
  - GpsL2MDIIPIITrackingFpga, [1043](#)
  - GpsL2MPcpsAcquisition, [1048](#)
  - GpsL2MPcpsAcquisitionFpga, [1054](#)
  - GpsL5DIIPIITracking, [1058](#)
  - GpsL5DIIPIITrackingFpga, [1063](#)
  - GpsL5iPcpsAcquisition, [1068](#)
  - GpsL5iPcpsAcquisitionFpga, [1074](#)
  - GpsL5TelemetryDecoder, [1079](#)
  - HybridObservables, [1096](#)
  - IbyteToCbyte, [1100](#)
  - IbyteToComplex, [1103](#)
  - IbyteToCshort, [1106](#)
  - IONGSMSSignalSource, [1124](#)
  - IshortToComplex, [1128](#)
  - IshortToCshort, [1131](#)
  - LabsatSignalSource, [1145](#)
  - LimesdrSignalSource, [1155](#)
  - MAX2771EVKITSignalSourceFPGA, [1165](#)
  - MmseResamplerConditioner, [1168](#)
  - MultichannelFileSignalSource, [1183](#)
  - NotchFilter, [1201](#)
  - NotchFilterLite, [1204](#)
  - OsmosdrSignalSource, [1225](#)
  - Pass\_Through, [1240](#)
  - PlutosdrSignalSource, [1291](#)
  - PulseBlankingFilter, [1306](#)
  - RawArraySignalSource, [1330](#)
  - Rtklib\_Pvt, [1364](#)
  - RtlTcpSignalSource, [1385](#)

- SbasL1TelemetryDecoder, [1393](#)
- SignalConditioner, [1427](#)
- SignalGenerator, [1431](#)
- SignalSourceBase, [1434](#)
- SpirGSS6450FileSignalSource, [1456](#)
- UhdSignalSource, [1514](#)
- get\_left\_block\_acq
  - Channel, [429](#)
  - ChannelInterface, [439](#)
- get\_left\_block\_trk
  - Channel, [429](#)
  - ChannelInterface, [439](#)
- get\_LLH
  - TcpCmdInterface, [1480](#)
- get\_longitude
  - Pvt\_Solution, [1318](#)
- get\_map\_copy
  - Concurrent\_Map< Data >, [456](#)
- get\_merkle\_root
  - Gnss\_Crypto, [869](#)
- get\_navfilename
  - Rinex\_Printer, [1334](#)
- get\_nsat
  - Galileo\_HAS\_data, [631](#)
- get\_nsat\_sub
  - Galileo\_HAS\_data, [631](#)
- get\_num\_satellites
  - Galileo\_HAS\_data, [631](#)
- get\_num\_subset\_satellites
  - Galileo\_HAS\_data, [631](#)
- get\_num\_valid\_observations
  - Pvt\_Solution, [1318](#)
- get\_obsfilename
  - Rinex\_Printer, [1334](#)
- get\_p dop
  - Rtklib\_Solver, [1372](#)
- get\_phase\_bias\_cycle
  - Galileo\_HAS\_data, [631](#)
- get\_PRN
  - Gnss\_Satellite, [880](#)
- get\_prn\_d
  - OSNMA\_NavData, [1235](#)
- get\_PRNs\_in\_mask
  - Galileo\_HAS\_data, [631](#), [632](#)
- get\_PRNs\_in\_submask
  - Galileo\_HAS\_data, [632](#)
- get\_public\_key\_type
  - Gnss\_Crypto, [869](#)
- get\_pvt
  - GNSSFlowgraph, [914](#)
- get\_rf\_link
  - Gnss\_Satellite, [881](#)
- get\_right\_block
  - Ad936xCustomSignalSource, [344](#)
  - Adrv9361z7035SignalSourceFPGA, [348](#)
  - ArraySignalConditioner, [363](#)
  - BeamformerFilter, [368](#)
  - BeidouB1iDIIPIITracking, [396](#)
  - BeidouB1iPcpsAcquisition, [401](#)
  - BeidouB1iTelemetryDecoder, [406](#)
  - BeidouB3iDIIPIITracking, [410](#)
  - BeidouB3iPcpsAcquisition, [415](#)
  - BeidouB3iTelemetryDecoder, [420](#)
  - ByteToShort, [425](#)
  - Channel, [429](#)
  - ChannelInterface, [439](#)
  - CshortToGrComplex, [475](#)
  - CustomUDPSignalSource, [480](#)
  - DirectResamplerConditioner, [491](#)
  - DMASignalSourceFPGA, [511](#)
  - FifoSignalSource, [536](#)
  - FileSourceBase, [552](#)
  - FirFilter, [560](#)
  - FlexibandSignalSource, [564](#)
  - Fmcomms2SignalSource, [568](#)
  - Fmcomms5SignalSourceFPGA, [571](#)
  - FreqXlatingFirFilter, [593](#)
  - GalileoE1BTelemetryDecoder, [663](#)
  - GalileoE1DIIPIIVemITracking, [667](#)
  - GalileoE1DIIPIIVemITrackingFpga, [672](#)
  - GalileoE1Pcps8msAmbiguousAcquisition, [677](#)
  - GalileoE1PcpsAmbiguousAcquisition, [683](#)
  - GalileoE1PcpsAmbiguousAcquisitionFpga, [689](#)
  - GalileoE1PcpsCccwsrAmbiguousAcquisition, [695](#)
  - GalileoE1PcpsQuickSyncAmbiguousAcquisition, [701](#)
  - GalileoE1PcpsTongAmbiguousAcquisition, [707](#)
  - GalileoE1TcpConnectorTracking, [712](#)
  - GalileoE5aDIIPIITracking, [716](#)
  - GalileoE5aDIIPIITrackingFpga, [721](#)
  - GalileoE5aNoncoherentIQAcquisitionCaf, [726](#)
  - GalileoE5aPcpsAcquisition, [732](#)
  - GalileoE5aPcpsAcquisitionFpga, [739](#)
  - GalileoE5aTelemetryDecoder, [744](#)
  - GalileoE5bDIIPIITracking, [748](#)
  - GalileoE5bPcpsAcquisition, [753](#)
  - GalileoE5bPcpsAcquisitionFpga, [761](#)
  - GalileoE5bTelemetryDecoder, [766](#)
  - GalileoE6DIIPIITracking, [770](#)
  - GalileoE6PcpsAcquisition, [775](#)
  - GalileoE6TelemetryDecoder, [780](#)
  - GenSignalSource, [784](#)
  - GlionassL1CaDIIPIICAidTracking, [828](#)
  - GlionassL1CaDIIPIITracking, [832](#)
  - GlionassL1CaPcpsAcquisition, [837](#)
  - GlionassL1CaTelemetryDecoder, [841](#)
  - GlionassL2CaDIIPIICAidTracking, [845](#)
  - GlionassL2CaDIIPIITracking, [849](#)
  - GlionassL2CaPcpsAcquisition, [854](#)
  - GlionassL2CaTelemetryDecoder, [858](#)
  - GNSSBlockInterface, [910](#)
  - GpsL1CaDIIPIITracking, [967](#)
  - GpsL1CaDIIPIITrackingFpga, [972](#)
  - GpsL1CaDIIPIITrackingGPU, [976](#)
  - GpsL1CaGaussianTracking, [980](#)
  - GpsL1CaKfTracking, [984](#)



- GpsL1CaPcpsAcquisition, 989
- GpsL1CaPcpsAcquisitionFineDoppler, 995
- GpsL1CaPcpsAcquisitionFpga, 1000
- GpsL1CaPcpsAssistedAcquisition, 1007
- GpsL1CaPcpsOpenCIAcquisition, 1012
- GpsL1CaPcpsQuickSyncAcquisition, 1018
- GpsL1CaPcpsTongAcquisition, 1023
- GpsL1CaTcpConnectorTracking, 1027
- GpsL1CaTelemetryDecoder, 1031
- GpsL2CTelemetryDecoder, 1035
- GpsL2MDIIPITracking, 1039
- GpsL2MDIIPITrackingFpga, 1043
- GpsL2MPcpsAcquisition, 1048
- GpsL2MPcpsAcquisitionFpga, 1054
- GpsL5DIIPITracking, 1058
- GpsL5DIIPITrackingFpga, 1063
- GpsL5IPcpsAcquisition, 1068
- GpsL5IPcpsAcquisitionFpga, 1074
- GpsL5TelemetryDecoder, 1079
- HybridObservables, 1096
- lbyteToCbyte, 1100
- lbyteToComplex, 1103
- lbyteToCshort, 1106
- IONGSMSSignalSource, 1124
- lshortToComplex, 1128
- lshortToCshort, 1131
- LabsatSignalSource, 1145
- LimesdrSignalSource, 1155
- MAX2771EVKITSignalSourceFPGA, 1165
- MmseResamplerConditioner, 1168
- MultichannelFileSignalSource, 1183
- NotchFilter, 1201
- NotchFilterLite, 1204
- OsmosdrSignalSource, 1225
- Pass\_Through, 1240
- PlutosdrSignalSource, 1291
- PulseBlankingFilter, 1306
- RawArraySignalSource, 1330
- Rtklib\_Pvt, 1364
- RtlTcpSignalSource, 1385
- SbasL1TelemetryDecoder, 1393
- SignalConditioner, 1428
- SignalGenerator, 1431
- SpirGSS6450FileSignalSource, 1456, 1457
- UhdSignalSource, 1514
- ZmqSignalSource, 1536
- get\_right\_block\_acq
  - Channel, 429
  - ChannelInterface, 439
- get\_right\_block\_trk
  - Channel, 429
  - ChannelInterface, 439
- get\_satellite
  - Gnss\_Signal, 897
- get\_satellite\_PRN
  - Gps\_Navigation\_Message, 960
- get\_signal
  - Channel, 430
- get\_signal\_str
  - Gnss\_Signal, 897
- get\_signals\_in\_mask
  - Galileo\_HAS\_data, 632
- get\_SOW
  - Beidou\_Dnav\_Navigation\_Message, 389
- get\_speed\_over\_ground
  - Pvt\_Solution, 1318
- get\_system
  - Gnss\_Satellite, 881
- get\_system\_short
  - Gnss\_Satellite, 881
- get\_systems\_string
  - Galileo\_HAS\_data, 632
- get\_systems\_subset\_string
  - Galileo\_HAS\_data, 632
- get\_t0G
  - Galileo\_Inav\_Message, 640
- get\_time\_offset\_s
  - Pvt\_Solution, 1319
- get\_TOW
  - Gps\_Navigation\_Message, 960
- get\_TOW0
  - Galileo\_Inav\_Message, 640
- get\_TOW1
  - Galileo\_Fnav\_Message, 624
- get\_TOW2
  - Galileo\_Fnav\_Message, 624
- get\_TOW3
  - Galileo\_Fnav\_Message, 624
- get\_TOW4
  - Galileo\_Fnav\_Message, 624
- get\_TOW5
  - Galileo\_Inav\_Message, 640
- get\_TOW6
  - Galileo\_Inav\_Message, 640
- get\_tow\_sf0
  - OSNMA\_NavData, 1235
- get\_tuner\_gain\_count
  - Rtl\_Tcp\_Dongle\_Info, 1380
- get\_tuner\_type
  - Rtl\_Tcp\_Dongle\_Info, 1380
- get\_utc\_model
  - Beidou\_Dnav\_Navigation\_Message, 389
  - Glonass\_Gnav\_Navigation\_Message, 807
  - Gps\_CNAV\_Navigation\_Message, 933
  - Gps\_Navigation\_Message, 960
- get\_utc\_time
  - TcpCmdInterface, 1480
- get\_validity\_interval\_s
  - Galileo\_HAS\_data, 632
- get\_vdop
  - Rtklib\_Solver, 1372
- get\_verified\_bits
  - OSNMA\_NavData, 1235
- get\_verified\_status
  - OSNMA\_NavData, 1235
- get\_WN0G



- Galileo\_Inav\_Message, [640](#)
- GetBlock
  - GNSSBlockFactory, [908](#)
- GetInteger
  - INIReader, [1108](#)
- getRfChannels
  - GenSignalSource, [784](#)
  - LabsatSignalSource, [1146](#)
  - SignalSourceBase, [1434](#)
- GetYIntercept
  - tracking\_libs, [203](#)
- GetYInterceptAbs
  - tracking\_libs, [203](#)
- gf
  - ssat\_t, [1459](#)
- gf2
  - ssat\_t, [1459](#)
- give
  - sbsigp\_t, [1398](#)
- glo\_cpbias
  - nav\_t, [1192](#)
- glo\_fcn
  - nav\_t, [1192](#)
- glo\_gnav\_utc
  - Gnss\_Sdr\_Supl\_Client, [890](#)
- gloicb
  - exterr\_t, [529](#)
- glomodear
  - prcopt\_t, [1299](#)
- GLONASS\_C20
  - core\_system\_parameters, [289](#)
- GLONASS\_CA\_NBR\_SATS
  - core\_system\_parameters, [289](#)
- GLONASS\_EARTH\_INCLINATION
  - core\_system\_parameters, [289](#)
- GLONASS\_EARTH\_RADIUS
  - core\_system\_parameters, [289](#)
- GLONASS\_F\_M\_A
  - core\_system\_parameters, [289](#)
- GLONASS\_FLATTENING
  - core\_system\_parameters, [289](#)
- GLONASS\_GM
  - core\_system\_parameters, [289](#)
- Glonass\_Gnav\_Almanac, [790](#)
  - d\_C\_n, [792](#)
  - d\_Delta\_i\_n\_A, [792](#)
  - d\_Delta\_T\_n\_A, [792](#)
  - d\_Delta\_T\_n\_A\_dot, [792](#)
  - d\_epsilon\_n\_A, [792](#)
  - d\_H\_n\_A, [792](#)
  - d\_KP, [793](#)
  - d\_I\_n, [793](#)
  - d\_lambda\_n\_A, [793](#)
  - d\_M\_n\_A, [793](#)
  - d\_n\_A, [793](#)
  - d\_omega\_n\_A, [793](#)
  - d\_t\_lambda\_n\_A, [793](#)
  - d\_tau\_n\_A, [793](#)
  - Glonass\_Gnav\_Almanac, [792](#)
  - i\_satellite\_freq\_channel, [794](#)
  - i\_satellite\_slot\_number, [794](#)
  - PRN, [794](#)
  - serialize, [792](#)
- glonass\_gnav\_almanac
  - Rtklib\_Solver, [1373](#)
- glonass\_gnav\_almanac.h, [2622](#), [2624](#)
- GLONASS\_GNAV\_CRC\_I\_INDEX
  - core\_system\_parameters, [289](#)
- GLONASS\_GNAV\_CRC\_J\_INDEX
  - core\_system\_parameters, [290](#)
- GLONASS\_GNAV\_CRC\_K\_INDEX
  - core\_system\_parameters, [290](#)
- GLONASS\_GNAV\_CRC\_L\_INDEX
  - core\_system\_parameters, [290](#)
- GLONASS\_GNAV\_CRC\_M\_INDEX
  - core\_system\_parameters, [290](#)
- GLONASS\_GNAV\_CRC\_N\_INDEX
  - core\_system\_parameters, [290](#)
- GLONASS\_GNAV\_CRC\_P\_INDEX
  - core\_system\_parameters, [290](#)
- GLONASS\_GNAV\_CRC\_Q\_INDEX
  - core\_system\_parameters, [290](#)
- GLONASS\_GNAV\_DATA\_SYMBOLS
  - core\_system\_parameters, [290](#)
- GLONASS\_GNAV\_ECC\_LOCATOR
  - core\_system\_parameters, [290](#)
- Glonass\_Gnav\_Ephemeris, [794](#)
  - compute\_GLONASS\_time, [798](#)
  - d\_AXn, [799](#)
  - d\_AYn, [799](#)
  - d\_AZn, [800](#)
  - d\_B\_n, [800](#)
  - d\_Delta\_tau\_n, [800](#)
  - d\_dtr, [800](#)
  - d\_E\_n, [800](#)
  - d\_F\_T, [800](#)
  - d\_gamma\_n, [800](#)
  - d\_iode, [800](#)
  - d\_l3rd\_n, [800](#)
  - d\_l5th\_n, [801](#)
  - d\_M, [801](#)
  - d\_m, [801](#)
  - d\_n, [801](#)
  - d\_N\_T, [801](#)
  - d\_P, [801](#)
  - d\_P\_1, [801](#)
  - d\_P\_2, [801](#)
  - d\_P\_3, [802](#)
  - d\_P\_4, [802](#)
  - d\_satClkDrift, [802](#)
  - d\_t\_b, [802](#)
  - d\_t\_k, [802](#)
  - d\_tau\_c, [802](#)
  - d\_tau\_n, [802](#)
  - d\_tod, [802](#)
  - d\_TOW, [803](#)

- d\_VXn, [803](#)
- d\_VYn, [803](#)
- d\_VZn, [803](#)
- d\_WN, [803](#)
- d\_Xn, [803](#)
- d\_Yn, [803](#)
- d\_yr, [803](#)
- d\_Zn, [803](#)
- Glonass\_Gnav\_Ephemeris, [797](#)
- glot\_to\_gpst, [798](#)
- glot\_to\_utc, [798](#)
- i\_satellite\_freq\_channel, [804](#)
- i\_satellite\_slot\_number, [804](#)
- PRN, [804](#)
- serialize, [798](#)
- sv\_clock\_drift, [799](#)
- glonass\_gnav\_ephemeris.h, [2625](#), [2627](#)
- glonass\_gnav\_ephemeris\_map
  - Gnss\_Sdr\_Supl\_Client, [890](#)
  - Rtklib\_Solver, [1373](#)
- GLONASS\_GNAV\_HAMMING\_CODE\_BITS
  - core\_system\_parameters, [291](#)
- Glonass\_Gnav\_Navigation\_Message, [804](#)
  - CRC\_test, [806](#)
  - get\_alm\_satellite\_slot\_number, [806](#)
  - get\_almanac, [806](#)
  - get\_ephemeris, [806](#)
  - get\_flag\_CRC\_test, [807](#)
  - get\_flag\_TOW\_new, [807](#)
  - get\_flag\_update\_slot\_number, [807](#)
  - get\_frame\_number, [807](#)
  - get\_utc\_model, [807](#)
  - Glonass\_Gnav\_Navigation\_Message, [806](#)
  - have\_new\_almanac, [807](#)
  - have\_new\_ephemeris, [807](#)
  - have\_new\_utc\_model, [807](#)
  - is\_flag\_TOW\_set, [807](#)
  - set\_flag\_ephemeris\_str\_1, [807](#)
  - set\_flag\_ephemeris\_str\_2, [808](#)
  - set\_flag\_ephemeris\_str\_3, [808](#)
  - set\_flag\_ephemeris\_str\_4, [808](#)
  - set\_flag\_TOW\_new, [808](#)
  - set\_flag\_update\_slot\_number, [808](#)
  - set\_rf\_link, [808](#)
  - string\_decoder, [808](#)
- glonass\_gnav\_navigation\_message.h, [2629](#), [2631](#)
- GLONASS\_GNAV\_PREAMBLE
  - core\_system\_parameters, [246](#)
- GLONASS\_GNAV\_PREAMBLE\_DURATION\_S
  - core\_system\_parameters, [291](#)
- GLONASS\_GNAV\_PREAMBLE\_LENGTH\_BITS
  - core\_system\_parameters, [291](#)
- GLONASS\_GNAV\_PREAMBLE\_LENGTH\_SYMBOLS
  - core\_system\_parameters, [291](#)
- GLONASS\_GNAV\_PREAMBLE\_PERIOD\_SYMBOLS
  - core\_system\_parameters, [291](#)
- GLONASS\_GNAV\_STRING\_BITS
  - core\_system\_parameters, [291](#)
- GLONASS\_GNAV\_STRING\_SYMBOLS
  - core\_system\_parameters, [291](#)
- GLONASS\_GNAV\_TELEMETRY\_RATE\_BITS\_SECOND
  - core\_system\_parameters, [291](#)
- GLONASS\_GNAV\_TELEMETRY\_RATE\_SYMBOLS\_SECOND
  - core\_system\_parameters, [291](#)
- GLONASS\_GNAV\_TELEMETRY\_SYMBOLS\_PER\_BIT
  - core\_system\_parameters, [291](#)
- GLONASS\_GNAV\_TELEMETRY\_SYMBOLS\_PER\_PREAMBLE\_BIT
  - core\_system\_parameters, [292](#)
- Glonass\_Gnav\_Utc\_Model, [809](#)
  - d\_B1, [810](#)
  - d\_B2, [810](#)
  - d\_N\_4, [810](#)
  - d\_N\_A, [810](#)
  - d\_tau\_c, [811](#)
  - d\_tau\_gps, [811](#)
  - Glonass\_Gnav\_Utc\_Model, [810](#)
  - serialize, [810](#)
  - utc\_time, [810](#)
  - valid, [811](#)
- glonass\_gnav\_utc\_model
  - Rtklib\_Solver, [1373](#)
- glonass\_gnav\_utc\_model.h, [2633](#), [2635](#)
- GLONASS\_GRAVITY
  - core\_system\_parameters, [292](#)
- GLONASS\_GRAVITY\_CORRECTION
  - core\_system\_parameters, [292](#)
- GLONASS\_J2
  - core\_system\_parameters, [292](#)
- GLONASS\_J4
  - core\_system\_parameters, [292](#)
- GLONASS\_J6
  - core\_system\_parameters, [292](#)
- GLONASS\_J8
  - core\_system\_parameters, [292](#)
- GLONASS\_L1\_CA\_CHIP\_PERIOD\_S
  - core\_system\_parameters, [292](#)
- glonass\_l1\_ca\_code\_gen\_complex
  - algorithms\_libs, [104](#)
- glonass\_l1\_ca\_code\_gen\_complex\_sampled
  - algorithms\_libs, [105](#)
- GLONASS\_L1\_CA\_CODE\_LENGTH\_CHIPS
  - core\_system\_parameters, [292](#)
- GLONASS\_L1\_CA\_CODE\_PERIOD\_S
  - core\_system\_parameters, [292](#)
- GLONASS\_L1\_CA\_CODE\_RATE\_CPS
  - core\_system\_parameters, [293](#)
- GLONASS\_L1\_CA\_DFREQ\_HZ
  - core\_system\_parameters, [293](#)
- glonass\_l1\_ca\_dll\_pll\_c\_aid\_tracking.h, [2197](#), [2198](#)
- glonass\_l1\_ca\_dll\_pll\_c\_aid\_tracking\_cc, [811](#)
- glonass\_l1\_ca\_dll\_pll\_c\_aid\_tracking\_cc.h, [2237](#), [2238](#)
- glonass\_l1\_ca\_dll\_pll\_c\_aid\_tracking\_cc\_sptr
  - tracking\_gr\_blocks, [197](#)
- glonass\_l1\_ca\_dll\_pll\_c\_aid\_tracking\_sc, [813](#)
- glonass\_l1\_ca\_dll\_pll\_c\_aid\_tracking\_sc.h, [2240](#), [2242](#)
- glonass\_l1\_ca\_dll\_pll\_c\_aid\_tracking\_sc\_sptr

- tracking\_gr\_blocks, 197
- glonass\_l1\_ca\_dll\_pll\_tracking.h, 2199, 2200
- Glonass\_L1\_Ca\_Dll\_Pll\_Tracking\_cc, 814
- glonass\_l1\_ca\_dll\_pll\_tracking\_cc.h, 2244, 2246
- glonass\_l1\_ca\_dll\_pll\_tracking\_cc\_sptr
  - tracking\_gr\_blocks, 197
- GLONASS\_L1\_CA\_FREQ\_HZ
  - core\_system\_parameters, 293
- GLONASS\_L1\_CA\_HISTORY\_DEEP
  - core\_system\_parameters, 293
- glonass\_l1\_ca\_pcps\_acquisition.h, 1581
- GLONASS\_L1\_CA\_SYMBOL\_RATE\_BPS
  - core\_system\_parameters, 293
- glonass\_l1\_ca\_telemetry\_decoder.h, 2124
- glonass\_l1\_ca\_telemetry\_decoder\_gs, 816
  - ~glonass\_l1\_ca\_telemetry\_decoder\_gs, 818
  - general\_work, 818
  - reset, 818
  - set\_channel, 818
  - set\_satellite, 818
- glonass\_l1\_ca\_telemetry\_decoder\_gs.h, 2143, 2145
- glonass\_l1\_ca\_telemetry\_decoder\_gs\_sptr
  - telemetry\_decoder\_gr\_blocks, 188
- GLONASS\_L1\_L2\_CA.h, 2636, 2641
- glonass\_l1\_signal\_replica.h, 1744
- GLONASS\_L2\_CA\_CHIP\_PERIOD\_S
  - core\_system\_parameters, 293
- glonass\_l2\_ca\_code\_gen\_complex
  - algorithms\_libs, 105
- glonass\_l2\_ca\_code\_gen\_complex\_sampled
  - algorithms\_libs, 105
- GLONASS\_L2\_CA\_CODE\_LENGTH\_CHIPS
  - core\_system\_parameters, 293
- GLONASS\_L2\_CA\_CODE\_PERIOD\_S
  - core\_system\_parameters, 293
- GLONASS\_L2\_CA\_CODE\_RATE\_CPS
  - core\_system\_parameters, 293
- GLONASS\_L2\_CA\_DFREQ\_HZ
  - core\_system\_parameters, 293
- glonass\_l2\_ca\_dll\_pll\_c\_aid\_tracking.h, 2201, 2202
- glonass\_l2\_ca\_dll\_pll\_c\_aid\_tracking\_cc, 818
- glonass\_l2\_ca\_dll\_pll\_c\_aid\_tracking\_cc.h, 2248, 2249
- glonass\_l2\_ca\_dll\_pll\_c\_aid\_tracking\_cc\_sptr
  - tracking\_gr\_blocks, 197
- glonass\_l2\_ca\_dll\_pll\_c\_aid\_tracking\_sc, 820
- glonass\_l2\_ca\_dll\_pll\_c\_aid\_tracking\_sc.h, 2251, 2253
- glonass\_l2\_ca\_dll\_pll\_c\_aid\_tracking\_sc\_sptr
  - tracking\_gr\_blocks, 197
- glonass\_l2\_ca\_dll\_pll\_tracking.h, 2203, 2204
- Glonass\_L2\_Ca\_Dll\_Pll\_Tracking\_cc, 821
- glonass\_l2\_ca\_dll\_pll\_tracking\_cc.h, 2255, 2256
- glonass\_l2\_ca\_dll\_pll\_tracking\_cc\_sptr
  - tracking\_gr\_blocks, 197
- GLONASS\_L2\_CA\_FREQ\_HZ
  - core\_system\_parameters, 294
- glonass\_l2\_ca\_pcps\_acquisition.h, 1583, 1584
- GLONASS\_L2\_CA\_SYMBOL\_RATE\_BPS
  - core\_system\_parameters, 294
- glonass\_l2\_ca\_telemetry\_decoder.h, 2126
- glonass\_l2\_ca\_telemetry\_decoder\_gs, 823
  - ~glonass\_l2\_ca\_telemetry\_decoder\_gs, 825
  - general\_work, 825
  - reset, 825
  - set\_channel, 825
  - set\_satellite, 825
- glonass\_l2\_ca\_telemetry\_decoder\_gs.h, 2146, 2147
- glonass\_l2\_ca\_telemetry\_decoder\_gs\_sptr
  - telemetry\_decoder\_gr\_blocks, 189
- glonass\_l2\_signal\_replica.h, 1745, 1746
- GLONASS\_LEAP\_SECONDS
  - core\_system\_parameters, 294
- GLONASS\_MOON\_ECCENTRICITY
  - core\_system\_parameters, 294
- GLONASS\_MOON\_GM
  - core\_system\_parameters, 294
- GLONASS\_MOON\_INCLINATION
  - core\_system\_parameters, 294
- GLONASS\_MOON\_OMEGA\_0
  - core\_system\_parameters, 294
- GLONASS\_MOON\_OMEGA\_1
  - core\_system\_parameters, 295
- GLONASS\_MOON\_Q0
  - core\_system\_parameters, 295
- GLONASS\_MOON\_Q1
  - core\_system\_parameters, 295
- GLONASS\_MOON\_SEMI\_MAJOR\_AXIS
  - core\_system\_parameters, 295
- GLONASS\_OMEGA\_EARTH\_DOT
  - core\_system\_parameters, 295
- GLONASS\_PRN
  - core\_system\_parameters, 295
- GLONASS\_SEMI\_MAJOR\_AXIS
  - core\_system\_parameters, 295
- GLONASS\_SUN\_ECCENTRICITY
  - core\_system\_parameters, 295
- GLONASS\_SUN\_GM
  - core\_system\_parameters, 295
- GLONASS\_SUN\_OMEGA
  - core\_system\_parameters, 295
- GLONASS\_SUN\_Q0
  - core\_system\_parameters, 296
- GLONASS\_SUN\_Q1
  - core\_system\_parameters, 296
- GLONASS\_SUN\_SEMI\_MAJOR\_AXIS
  - core\_system\_parameters, 296
- GLONASS\_TAU\_0
  - core\_system\_parameters, 296
- GLONASS\_TAU\_1
  - core\_system\_parameters, 296
- GLONASS\_U0
  - core\_system\_parameters, 296
- GlonassL1CaDllPllCAidTracking, 825
  - connect, 828
  - disconnect, 828
  - get\_left\_block, 828
  - get\_right\_block, 828

- implementation, 828
- item\_size, 828
- role, 829
- set\_channel, 829
- set\_gnss\_synchro, 829
- start\_tracking, 829
- stop\_tracking, 829
- GlonassL1CaDIPIITracking, 829
  - connect, 832
  - disconnect, 832
  - get\_left\_block, 832
  - get\_right\_block, 832
  - implementation, 832
  - item\_size, 832
  - role, 833
  - set\_channel, 833
  - set\_gnss\_synchro, 833
  - start\_tracking, 833
  - stop\_tracking, 833
- GlonassL1CaPcpsAcquisition, 833
  - connect, 836
  - disconnect, 836
  - get\_left\_block, 837
  - get\_right\_block, 837
  - implementation, 837
  - init, 837
  - item\_size, 837
  - mag, 837
  - reset, 837
  - role, 837
  - set\_channel, 837
  - set\_channel\_fsm, 837
  - set\_doppler\_max, 838
  - set\_doppler\_step, 838
  - set\_gnss\_synchro, 838
  - set\_local\_code, 838
  - set\_resampler\_latency, 838
  - set\_state, 838
  - set\_threshold, 838
  - stop\_acquisition, 838
- GlonassL1CaTelemetryDecoder, 839
  - connect, 841
  - disconnect, 841
  - get\_left\_block, 841
  - get\_right\_block, 841
  - implementation, 841
  - item\_size, 841
  - reset, 841
  - role, 842
  - set\_channel, 842
  - set\_satellite, 842
- GlonassL2CaDIPIICAidTracking, 842
  - connect, 845
  - disconnect, 845
  - get\_left\_block, 845
  - get\_right\_block, 845
  - implementation, 845
  - item\_size, 845
  - role, 846
  - set\_channel, 846
  - set\_gnss\_synchro, 846
  - start\_tracking, 846
  - stop\_tracking, 846
- GlonassL2CaDIPIITracking, 846
  - connect, 849
  - disconnect, 849
  - get\_left\_block, 849
  - get\_right\_block, 849
  - implementation, 849
  - item\_size, 849
  - role, 850
  - set\_channel, 850
  - set\_gnss\_synchro, 850
  - start\_tracking, 850
  - stop\_tracking, 850
- GlonassL2CaPcpsAcquisition, 850
  - connect, 853
  - disconnect, 853
  - get\_left\_block, 854
  - get\_right\_block, 854
  - implementation, 854
  - init, 854
  - item\_size, 854
  - mag, 854
  - reset, 854
  - role, 854
  - set\_channel, 854
  - set\_channel\_fsm, 854
  - set\_doppler\_max, 855
  - set\_doppler\_step, 855
  - set\_gnss\_synchro, 855
  - set\_local\_code, 855
  - set\_resampler\_latency, 855
  - set\_state, 855
  - set\_threshold, 855
  - stop\_acquisition, 855
- GlonassL2CaTelemetryDecoder, 856
  - connect, 858
  - disconnect, 858
  - get\_left\_block, 858
  - get\_right\_block, 858
  - implementation, 858
  - item\_size, 858
  - reset, 858
  - role, 859
  - set\_channel, 859
  - set\_satellite, 859
- glot\_to\_gpst
  - Glonass\_Gnav\_Ephemeris, 798
- glot\_to\_utc
  - Glonass\_Gnav\_Ephemeris, 798
- GME
  - rtklib\_tides.h, 1871
- GMM
  - rtklib\_tides.h, 1871
- GMS

- rtklib\_tides.h, [1871](#)
- GNSS block interfaces, [206](#)
  - ~SignalSourceInterface, [207](#)
  - gnss\_make\_shared, [207](#)
  - gnss\_shared\_ptr, [206](#)
  - SignalSourceInterface, [207](#)
- GNSS-SDR, [1](#)
- Gnss\_Almanac, [859](#)
  - af0, [863](#)
  - af1, [863](#)
  - delta\_i, [863](#)
  - ecc, [863](#)
  - Gnss\_Almanac, [862](#)
  - M\_0, [863](#)
  - omega, [863](#)
  - OMEGA\_0, [863](#)
  - OMEGAdot, [864](#)
  - predicted\_doppler, [862](#)
  - PRN, [864](#)
  - satellitePosVelComputation, [863](#)
  - sqrtA, [864](#)
  - System, [864](#)
  - toa, [864](#)
  - WNa, [864](#)
- gnss\_almanac.h, [2645](#), [2646](#)
- gnss\_block\_factory.h, [2388](#), [2389](#)
- gnss\_block\_interface.h, [2317](#), [2318](#)
- Gnss\_circular\_deque
  - algorithms\_libs, [105](#)
- Gnss\_circular\_deque< T >, [865](#)
- gnss\_circular\_deque.h, [1747](#), [1748](#)
- Gnss\_Crypto, [867](#)
  - ~Gnss\_Crypto, [868](#)
  - compute\_CMAC\_AES, [868](#)
  - compute\_HMAC\_SHA\_256, [868](#)
  - compute\_SHA3\_256, [869](#)
  - compute\_SHA\_256, [869](#)
  - get\_merkle\_root, [869](#)
  - get\_public\_key\_type, [869](#)
  - Gnss\_Crypto, [868](#)
  - have\_public\_key, [869](#)
  - set\_merkle\_root, [869](#)
  - set\_public\_key, [869](#)
  - set\_public\_key\_type, [869](#)
  - store\_public\_key, [869](#)
  - verify\_signature\_ecdsa\_p256, [869](#)
  - verify\_signature\_ecdsa\_p521, [870](#)
- gnss\_crypto.h, [2337](#)
- Gnss\_Ephemeris, [870](#)
  - af0, [874](#)
  - af1, [874](#)
  - af2, [874](#)
  - Cic, [874](#)
  - Cis, [874](#)
  - Crc, [874](#)
  - Crs, [874](#)
  - Cuc, [875](#)
  - Cus, [875](#)
  - delta\_n, [875](#)
  - dtr, [875](#)
  - ecc, [875](#)
  - i\_0, [875](#)
  - idot, [875](#)
  - M\_0, [876](#)
  - omega, [876](#)
  - OMEGA\_0, [876](#)
  - OMEGAdot, [876](#)
  - predicted\_doppler, [873](#)
  - PRN, [876](#)
  - satClkDrift, [876](#)
  - satellitePosition, [873](#)
  - satpos\_X, [876](#)
  - satpos\_Y, [876](#)
  - satpos\_Z, [877](#)
  - satvel\_X, [877](#)
  - satvel\_Y, [877](#)
  - satvel\_Z, [877](#)
  - sqrtA, [877](#)
  - sv\_clock\_drift, [873](#)
  - System, [877](#)
  - toc, [877](#)
  - toe, [877](#)
  - tow, [878](#)
  - WN, [878](#)
- gnss\_ephemeris.h, [2647](#), [2648](#)
- gnss\_fft\_complex\_fwd
  - gnss\_sdr\_fft.h, [1751](#)
- gnss\_fft\_complex\_rev
  - gnss\_sdr\_fft.h, [1751](#)
- gnss\_fft\_fwd\_make\_unique
  - gnss\_sdr\_fft.h, [1752](#)
- gnss\_fft\_fwd\_unique\_ptr
  - gnss\_sdr\_fft.h, [1752](#)
- gnss\_fft\_rev\_make\_unique
  - gnss\_sdr\_fft.h, [1752](#)
- gnss\_fft\_rev\_unique\_ptr
  - gnss\_sdr\_fft.h, [1752](#)
- gnss\_flowgraph.h, [2390](#), [2391](#)
- gnss\_frequencies.h, [2650](#), [2651](#)
- gnss\_id\_clock\_subset
  - Galileo\_HAS\_data, [633](#)
- gnss\_id\_mask
  - Galileo\_HAS\_data, [633](#)
- gnss\_iod
  - Galileo\_HAS\_data, [633](#)
- gnss\_make\_shared
  - GNSS block interfaces, [207](#)
- gnss\_obs\_codes.h, [2652](#), [2655](#)
- GNSS\_OMEGA\_EARTH\_DOT
  - core\_system\_parameters, [296](#)
- GNSS\_PI
  - core\_system\_parameters, [296](#)
- Gnss\_Satellite, [878](#)
  - ~Gnss\_Satellite, [880](#)
  - get\_block, [880](#)
  - get\_PRN, [880](#)

- get\_rf\_link, 881
- get\_system, 881
- get\_system\_short, 881
- Gnss\_Satellite, 879, 880
- operator<<, 882
- operator=, 881
- operator==, 882
- update\_PRN, 881
- what\_block, 882
- gnss\_satellite.h, 2656, 2657
- GNSS\_SDR\_BEAMFORMER\_CHANNELS
  - input\_filter\_gr\_blocks, 89
- gnss\_sdr\_create\_directory.h, 1749, 1750
- gnss\_sdr\_fft.h, 1751, 1752
  - gnss\_fft\_complex\_fwd, 1751
  - gnss\_fft\_complex\_rev, 1751
  - gnss\_fft\_fwd\_make\_unique, 1752
  - gnss\_fft\_fwd\_unique\_ptr, 1752
  - gnss\_fft\_rev\_make\_unique, 1752
  - gnss\_fft\_rev\_unique\_ptr, 1752
- gnss\_sdr\_filesystem.h, 1753, 1754
- gnss\_sdr\_flags, 113
  - ABSL\_DECLARE\_FLAG, 114–117
  - DEFAULT\_CARRIER\_SMOOTHING\_FACTOR, 118
- gnss\_sdr\_flags.h, 1755, 1756
- gnss\_sdr\_fpga\_sample\_counter, 882
- gnss\_sdr\_fpga\_sample\_counter.h, 2339, 2340
- gnss\_sdr\_fpga\_sample\_counter\_sptr
  - core\_libs, 209
- gnss\_sdr\_make\_unique.h, 1759
- gnss\_sdr\_sample\_counter, 884
- gnss\_sdr\_sample\_counter.h, 2341, 2342
- gnss\_sdr\_sample\_counter\_sptr
  - core\_libs, 209
- gnss\_sdr\_string\_literals.h, 1760
- Gnss\_Sdr\_Supl\_Client, 885
  - gal\_almanac\_map, 890
  - gal\_ephemeris\_map, 890
  - gal\_iono, 890
  - gal\_utc, 890
  - glo\_gnav\_utc, 890
  - glonass\_gnav\_ephemeris\_map, 890
  - gps\_acq\_map, 891
  - gps\_almanac\_map, 891
  - gps\_cnav\_ephemeris\_map, 891
  - gps\_cnav\_utc, 891
  - gps\_ephemeris\_map, 891
  - gps\_iono, 891
  - gps\_ref\_loc, 891
  - gps\_time, 891
  - gps\_utc, 891
  - load\_cnav\_ephemeris\_xml, 887
  - load\_cnav\_utc\_xml, 887
  - load\_ephemeris\_xml, 887
  - load\_gal\_almanac\_xml, 887
  - load\_gal\_ephemeris\_xml, 887
  - load\_gal\_iono\_xml, 887
  - load\_gal\_utc\_xml, 888
  - load\_glo\_utc\_xml, 888
  - load\_gnav\_ephemeris\_xml, 888
  - load\_gps\_almanac\_xml, 888
  - load\_iono\_xml, 888
  - load\_ref\_location\_xml, 888
  - load\_ref\_time\_xml, 888
  - load\_utc\_xml, 888
  - request, 891
  - save\_cnav\_ephemeris\_map\_xml, 888
  - save\_cnav\_utc\_xml, 888
  - save\_ephemeris\_map\_xml, 889
  - save\_gal\_almanac\_xml, 889
  - save\_gal\_ephemeris\_map\_xml, 889
  - save\_gal\_iono\_xml, 889
  - save\_gal\_utc\_xml, 889
  - save\_glo\_utc\_xml, 889
  - save\_gnav\_ephemeris\_map\_xml, 889
  - save\_gps\_almanac\_xml, 889
  - save\_iono\_xml, 890
  - save\_ref\_location\_xml, 890
  - save\_ref\_time\_xml, 890
  - save\_utc\_xml, 890
  - server\_name, 891
  - server\_port, 891
- gnss\_sdr\_supl\_client.h, 2343, 2344
- gnss\_sdr\_time\_counter, 892
- gnss\_sdr\_time\_counter.h, 2347, 2348
- gnss\_sdr\_time\_counter\_sptr
  - core\_libs, 209
- Gnss\_Sdr\_Timestamp, 893
- gnss\_sdr\_timestamp.h, 2088, 2089
- Gnss\_Sdr\_Valve, 894
- gnss\_sdr\_valve.h, 2090, 2091
- gnss\_shared\_ptr
  - GNSS block interfaces, 206
- Gnss\_Signal, 896
  - get\_satellite, 897
  - get\_signal\_str, 897
  - operator<<, 897
  - operator==, 897
- gnss\_signal.h, 2658, 2660
- gnss\_signal\_replica.h, 1761, 1762
- Gnss\_Synchro, 897
  - ~Gnss\_Synchro, 900
  - Acq\_delay\_samples, 902
  - Acq\_doppler\_hz, 902
  - Acq\_doppler\_step, 902
  - Acq\_samplestamp\_samples, 902
  - Carrier\_Doppler\_hz, 902
  - Carrier\_phase\_rads, 902
  - Channel\_ID, 902
  - CN0\_dB\_hz, 902
  - Code\_phase\_samples, 903
  - correlation\_length\_ms, 903
  - Flag\_PLL\_180\_deg\_phase\_locked, 903
  - Flag\_valid\_acquisition, 903
  - Flag\_valid\_pseudorange, 903

- Flag\_valid\_symbol\_output, 903
- Flag\_valid\_word, 903
- fs, 903
- Gnss\_Synchro, 900
- interp\_TOW\_ms, 904
- operator=, 901
- PRN, 904
- Prompt\_I, 904
- Prompt\_Q, 904
- Pseudorange\_m, 904
- RX\_time, 904
- serialize, 901
- Signal, 904
- System, 904
- TOW\_at\_current\_symbol\_ms, 904
- Tracking\_sample\_counter, 905
- gnss\_synchro.h, 2660, 2661
- gnss\_synchro\_monitor, 905
  - ~gnss\_synchro\_monitor, 906
- gnss\_synchro\_monitor.h, 2371, 2372
- gnss\_synchro\_monitor\_sptr
  - core\_monitor, 211
- Gnss\_Synchro\_Udp\_Sink, 906
- gnss\_synchro\_udp\_sink.h, 2373, 2374
- gnss\_time.h, 1763, 1764
- GNSSBlockFactory, 907
  - GetBlock, 908
- GNSSBlockInterface, 908
  - connect, 909
  - disconnect, 909
  - get\_left\_block, 910
  - get\_right\_block, 910
  - implementation, 910
  - item\_size, 910
  - role, 911
  - start, 911
- GNSSFlowgraph, 911
  - ~GNSSFlowgraph, 913
  - acquisition\_manager, 913
  - apply\_action, 913
  - connect, 913
  - connected, 914
  - disconnect, 914
  - get\_pvt, 914
  - GNSSFlowgraph, 913
  - priorize\_satellites, 914
  - running, 914
  - send\_telemetry\_msg, 914
  - set\_configuration, 914
  - start, 914
  - stop, 914
  - wait, 914
- GnssTime, 915
  - rx\_time, 915
  - tow\_ms, 915
  - tow\_ms\_fraction, 915
  - week, 915
- Gps\_Acq\_Assist, 916
  - Azimuth, 917
  - Code\_Phase, 917
  - Code\_Phase\_int, 917
  - Code\_Phase\_window, 917
  - Doppler0, 917
  - Doppler1, 917
  - dopplerUncertainty, 918
  - Elevation, 918
  - Gps\_Acq\_Assist, 917
  - GPS\_Bit\_Number, 918
  - PRN, 918
  - tow, 918
- gps\_acq\_assist.h, 2664, 2665
- gps\_acq\_map
  - Gnss\_Sdr\_Supl\_Client, 891
- Gps\_Almanac, 918
  - AS\_status, 922
  - Gps\_Almanac, 921
  - serialize, 922
  - SV\_health, 922
- gps\_almanac.h, 2666, 2667
- gps\_almanac\_map
  - Gnss\_Sdr\_Supl\_Client, 891
  - Rtklib\_Solver, 1373
- GPS\_Bit\_Number
  - Gps\_Acq\_Assist, 918
- GPS\_CA\_PREAMBLE
  - core\_system\_parameters, 296
- GPS\_CA\_PREAMBLE\_DURATION\_MS
  - core\_system\_parameters, 296
- GPS\_CA\_PREAMBLE\_DURATION\_S
  - core\_system\_parameters, 297
- GPS\_CA\_PREAMBLE\_LENGTH\_BITS
  - core\_system\_parameters, 297
- GPS\_CA\_PREAMBLE\_LENGTH\_SYMBOLS
  - core\_system\_parameters, 297
- GPS\_CA\_PREAMBLE\_SYMBOLS\_STR
  - core\_system\_parameters, 297
- GPS\_CA\_TELEMETRY\_RATE\_BITS\_SECOND
  - core\_system\_parameters, 297
- GPS\_CA\_TELEMETRY\_RATE\_SYMBOLS\_SECOND
  - core\_system\_parameters, 297
- GPS\_CA\_TELEMETRY\_SYMBOLS\_PER\_BIT
  - core\_system\_parameters, 297
- GPS\_CNAV.h, 2668, 2671
- GPS\_CNAV\_DATA\_PAGE\_BITS
  - core\_system\_parameters, 297
- Gps\_CNAV\_Ephemeris, 922
  - Adot, 928
  - alert\_flag, 928
  - antispoofing\_flag, 928
  - delta\_A, 928
  - delta\_ndot, 928
  - delta\_OMEGAAdot, 928
  - Gps\_CNAV\_Ephemeris, 927
  - integrity\_status\_flag, 928
  - ISCL1, 928
  - ISCL2, 928



- ISCL5I, [929](#)
- ISCL5Q, [929](#)
- l2c\_phasing\_flag, [929](#)
- serialize, [927](#)
- signal\_health, [929](#)
- TGD, [929](#)
- toe1, [929](#)
- toe2, [929](#)
- top, [929](#)
- URA, [929](#)
- URA0, [929](#)
- URA1, [930](#)
- URA2, [930](#)
- gps\_cnav\_ephemeris.h, [2673](#), [2675](#)
- gps\_cnav\_ephemeris\_map
  - Gnss\_Sdr\_Supl\_Client, [891](#)
  - Rtklib\_Solver, [1373](#)
- Gps\_CNAV\_Iono, [930](#)
  - Gps\_CNAV\_Iono, [932](#)
- gps\_cnav\_iono
  - Rtklib\_Solver, [1373](#)
- gps\_cnav\_iono.h, [2676](#), [2678](#)
- Gps\_CNAV\_Navigation\_Message, [932](#)
  - get\_ephemeris, [933](#)
  - get\_iono, [933](#)
  - get\_utc\_model, [933](#)
  - Gps\_CNAV\_Navigation\_Message, [933](#)
  - have\_new\_ephemeris, [933](#)
  - have\_new\_iono, [933](#)
  - have\_new\_utc\_model, [933](#)
- gps\_cnav\_navigation\_message.h, [2678](#), [2679](#)
- gps\_cnav\_utc
  - Gnss\_Sdr\_Supl\_Client, [891](#)
- Gps\_CNAV\_Utc\_Model, [934](#)
  - Gps\_CNAV\_Utc\_Model, [936](#)
- gps\_cnav\_utc\_model
  - Rtklib\_Solver, [1373](#)
- gps\_cnav\_utc\_model.h, [2681](#), [2682](#)
- Gps\_Ephemeris, [936](#)
  - alert\_flag, [942](#)
  - antispoofing\_flag, [942](#)
  - AODO, [942](#)
  - code\_on\_L2, [942](#)
  - fit\_interval\_flag, [942](#)
  - Gps\_Ephemeris, [941](#)
  - integrity\_status\_flag, [942](#)
  - IODC, [942](#)
  - IODE\_SF2, [942](#)
  - IODE\_SF3, [943](#)
  - L2\_P\_data\_flag, [943](#)
  - satelliteBlock, [943](#)
  - serialize, [941](#)
  - spare1, [943](#)
  - spare2, [943](#)
  - SV\_accuracy, [943](#)
  - SV\_health, [943](#)
  - TGD, [943](#)
- gps\_ephemeris.h, [2682](#), [2683](#)
- gps\_ephemeris\_map
  - Gnss\_Sdr\_Supl\_Client, [891](#)
  - Rtklib\_Solver, [1373](#)
- GPS\_F
  - core\_system\_parameters, [297](#)
- GPS\_GM
  - core\_system\_parameters, [297](#)
- Gps\_Iono, [944](#)
  - alpha0, [946](#)
  - alpha1, [946](#)
  - alpha2, [946](#)
  - alpha3, [946](#)
  - beta0, [946](#)
  - beta1, [947](#)
  - beta2, [947](#)
  - beta3, [947](#)
  - Gps\_Iono, [946](#)
  - serialize, [946](#)
  - valid, [947](#)
- gps\_iono
  - Gnss\_Sdr\_Supl\_Client, [891](#)
  - Rtklib\_Solver, [1373](#)
- gps\_iono.h, [2685](#), [2686](#)
- GPS\_L1\_CA.h, [2687](#), [2692](#)
- GPS\_L1\_CA\_BIT\_PERIOD\_MS
  - core\_system\_parameters, [297](#)
- GPS\_L1\_CA\_CHIP\_PERIOD\_S
  - core\_system\_parameters, [298](#)
- gps\_l1\_ca\_code\_gen\_complex
  - algorithms\_libs, [106](#)
- gps\_l1\_ca\_code\_gen\_complex\_sampled
  - algorithms\_libs, [106](#)
- gps\_l1\_ca\_code\_gen\_float
  - algorithms\_libs, [106](#)
- gps\_l1\_ca\_code\_gen\_int
  - algorithms\_libs, [106](#)
- GPS\_L1\_CA\_CODE\_LENGTH\_CHIPS
  - core\_system\_parameters, [298](#)
- GPS\_L1\_CA\_CODE\_PERIOD\_MS
  - core\_system\_parameters, [298](#)
- GPS\_L1\_CA\_CODE\_PERIOD\_S
  - core\_system\_parameters, [298](#)
- GPS\_L1\_CA\_CODE\_RATE\_CPS
  - core\_system\_parameters, [298](#)
- gps\_l1\_ca\_dll\_pll\_tracking.h, [2205](#), [2206](#)
- gps\_l1\_ca\_dll\_pll\_tracking\_fpga.h, [2207](#), [2208](#)
- gps\_l1\_ca\_dll\_pll\_tracking\_gpu.h, [2210](#)
- Gps\_L1\_Ca\_Dll\_Pll\_Tracking\_GPU\_cc, [947](#)
- gps\_l1\_ca\_dll\_pll\_tracking\_gpu\_cc.h, [2258](#), [2260](#)
- gps\_l1\_ca\_dll\_pll\_tracking\_gpu\_cc\_sptr
  - tracking\_gr\_blocks, [197](#)
- gps\_l1\_ca\_gaussian\_tracking.h, [2212](#)
- Gps\_L1\_Ca\_Gaussian\_Tracking\_cc, [949](#)
- gps\_l1\_ca\_gaussian\_tracking\_cc.h, [2262](#), [2263](#)
- gps\_l1\_ca\_gaussian\_tracking\_cc\_sptr
  - tracking\_gr\_blocks, [198](#)
- GPS\_L1\_CA\_HISTORY\_DEEP
  - core\_system\_parameters, [298](#)



- gps\_l1\_ca\_kf\_tracking.h, 2214
- GPS\_L1\_CA\_OPT\_ACQ\_FS\_SPS
  - core\_system\_parameters, 298
- gps\_l1\_ca\_pcps\_acquisition.h, 1586, 1587
- gps\_l1\_ca\_pcps\_acquisition\_fine\_doppler.h, 1589, 1590
- gps\_l1\_ca\_pcps\_acquisition\_fpga.h, 1592, 1593
- gps\_l1\_ca\_pcps\_assisted\_acquisition.h, 1595, 1596
- gps\_l1\_ca\_pcps\_openc1\_acquisition.h, 1598, 1599
- gps\_l1\_ca\_pcps\_quicksync\_acquisition.h, 1601, 1602
- gps\_l1\_ca\_pcps\_tong\_acquisition.h, 1604, 1605
- gps\_l1\_ca\_tcp\_connector\_tracking.h, 2215, 2216
- Gps\_L1\_Ca\_Tcp\_Connector\_Tracking\_cc, 950
- gps\_l1\_ca\_tcp\_connector\_tracking\_cc.h, 2266, 2267
- gps\_l1\_ca\_tcp\_connector\_tracking\_cc\_sptr
  - tracking\_gr\_blocks, 198
- gps\_l1\_ca\_telemetry\_decoder.h, 2127, 2128
- gps\_l1\_ca\_telemetry\_decoder\_gs, 952
  - general\_work, 953
  - set\_channel, 954
  - set\_satellite, 954
- gps\_l1\_ca\_telemetry\_decoder\_gs.h, 2149, 2150
- gps\_l1\_ca\_telemetry\_decoder\_gs\_sptr
  - telemetry\_decoder\_gr\_blocks, 189
- GPS\_L1\_FREQ\_HZ
  - core\_system\_parameters, 298
- GPS\_L2\_CNAV\_DATA\_PAGE\_BITS
  - core\_system\_parameters, 298
- GPS\_L2\_CNAV\_DATA\_PAGE\_DURATION\_S
  - core\_system\_parameters, 298
- GPS\_L2\_CNAV\_DATA\_PAGE\_SYMBOLS
  - core\_system\_parameters, 299
- GPS\_L2\_FREQ\_HZ
  - core\_system\_parameters, 299
- GPS\_L2\_L\_CODE\_LENGTH\_CHIPS
  - core\_system\_parameters, 299
- GPS\_L2\_L\_CODE\_RATE\_CPS
  - core\_system\_parameters, 299
- GPS\_L2\_L\_PERIOD\_S
  - core\_system\_parameters, 299
- GPS\_L2\_M\_CODE\_LENGTH\_CHIPS
  - core\_system\_parameters, 299
- GPS\_L2\_M\_CODE\_RATE\_CPS
  - core\_system\_parameters, 299
- gps\_l2\_m\_dll\_pll\_tracking.h, 2217, 2218
- gps\_l2\_m\_dll\_pll\_tracking\_fpga.h, 2219, 2220
- gps\_l2\_m\_pcps\_acquisition.h, 1607
- gps\_l2\_m\_pcps\_acquisition\_fpga.h, 1610
- GPS\_L2\_M\_PERIOD\_S
  - core\_system\_parameters, 299
- GPS\_L2\_SAMPLES\_PER\_SYMBOL
  - core\_system\_parameters, 299
- GPS\_L2\_SYMBOLS\_PER\_BIT
  - core\_system\_parameters, 299
- GPS\_L2\_V27\_HISTORY\_LENGTH\_BITS
  - telemetry\_decoder\_libswiftcnv, 191
- GPS\_L2C.h, 2695, 2697
- GPS\_L2C\_HISTORY\_DEEP
  - core\_system\_parameters, 299
- gps\_l2c\_m\_code\_gen\_complex
  - algorithms\_libs, 107
- gps\_l2c\_m\_code\_gen\_complex\_sampled
  - algorithms\_libs, 107
- gps\_l2c\_m\_code\_gen\_float
  - algorithms\_libs, 107
- GPS\_L2C\_M\_INIT\_REG
  - core\_system\_parameters, 300
- GPS\_L2C\_OPT\_ACQ\_FS\_SPS
  - core\_system\_parameters, 300
- gps\_l2c\_signal\_replica.h, 1764, 1765
- gps\_l2c\_telemetry\_decoder.h, 2129, 2130
- gps\_l2c\_telemetry\_decoder\_gs, 954
  - general\_work, 955
  - set\_channel, 955
  - set\_satellite, 956
- gps\_l2c\_telemetry\_decoder\_gs.h, 2152, 2153
- gps\_l2c\_telemetry\_decoder\_gs\_sptr
  - telemetry\_decoder\_gr\_blocks, 189
- GPS\_L2C\_V27\_DECODE\_BITS
  - telemetry\_decoder\_libswiftcnv, 191
- GPS\_L2C\_V27\_DELAY\_BITS
  - telemetry\_decoder\_libswiftcnv, 191
- GPS\_L2C\_V27\_INIT\_BITS
  - telemetry\_decoder\_libswiftcnv, 191
- GPS\_L5.h, 2698, 2699
- GPS\_L5\_CNAV\_DATA\_PAGE\_BITS
  - core\_system\_parameters, 300
- GPS\_L5\_CNAV\_DATA\_PAGE\_DURATION\_S
  - core\_system\_parameters, 300
- GPS\_L5\_CNAV\_DATA\_PAGE\_SYMBOLS
  - core\_system\_parameters, 300
- gps\_l5\_dll\_pll\_tracking.h, 2221, 2222
- gps\_l5\_dll\_pll\_tracking\_fpga.h, 2223, 2224
- GPS\_L5\_FREQ\_HZ
  - core\_system\_parameters, 300
- GPS\_L5\_HISTORY\_DEEP
  - core\_system\_parameters, 300
- GPS\_L5\_OPT\_ACQ\_FS\_SPS
  - core\_system\_parameters, 301
- GPS\_L5\_SAMPLES\_PER\_SYMBOL
  - core\_system\_parameters, 301
- gps\_l5\_signal\_replica.h, 1766
- GPS\_L5\_SYMBOLS\_PER\_BIT
  - core\_system\_parameters, 301
- gps\_l5\_telemetry\_decoder.h, 2131, 2132
- gps\_l5\_telemetry\_decoder\_gs, 956
  - set\_channel, 957
  - set\_satellite, 957
- gps\_l5\_telemetry\_decoder\_gs.h, 2154, 2156
- gps\_l5\_telemetry\_decoder\_gs\_sptr
  - telemetry\_decoder\_gr\_blocks, 189
- gps\_l5i\_code\_gen\_complex
  - algorithms\_libs, 107
- gps\_l5i\_code\_gen\_complex\_sampled
  - algorithms\_libs, 107
- gps\_l5i\_code\_gen\_float

- algorithms\_libs, [107](#)
- GPS\_L5I\_CODE\_LENGTH\_CHIPS
  - core\_system\_parameters, [301](#)
- GPS\_L5I\_CODE\_RATE\_CPS
  - core\_system\_parameters, [301](#)
- GPS\_L5I\_INIT\_REG
  - core\_system\_parameters, [301](#)
- GPS\_L5I\_NH\_CODE
  - core\_system\_parameters, [301](#)
- GPS\_L5I\_NH\_CODE\_LENGTH
  - core\_system\_parameters, [301](#)
- GPS\_L5I\_NH\_CODE\_STR
  - core\_system\_parameters, [301](#)
- gps\_l5i\_pcps\_acquisition.h, [1613](#)
- gps\_l5i\_pcps\_acquisition\_fpga.h, [1615](#), [1616](#)
- GPS\_L5I\_PERIOD\_MS
  - core\_system\_parameters, [301](#)
- GPS\_L5I\_PERIOD\_S
  - core\_system\_parameters, [301](#)
- GPS\_L5I\_SYMBOL\_PERIOD\_MS
  - core\_system\_parameters, [302](#)
- GPS\_L5I\_SYMBOL\_PERIOD\_S
  - core\_system\_parameters, [302](#)
- gps\_l5q\_code\_gen\_complex
  - algorithms\_libs, [108](#)
- gps\_l5q\_code\_gen\_complex\_sampled
  - algorithms\_libs, [108](#)
- gps\_l5q\_code\_gen\_float
  - algorithms\_libs, [108](#)
- GPS\_L5Q\_CODE\_LENGTH\_CHIPS
  - core\_system\_parameters, [302](#)
- GPS\_L5Q\_CODE\_RATE\_CPS
  - core\_system\_parameters, [302](#)
- GPS\_L5Q\_INIT\_REG
  - core\_system\_parameters, [302](#)
- GPS\_L5Q\_NH\_CODE
  - core\_system\_parameters, [302](#)
- GPS\_L5Q\_NH\_CODE\_LENGTH
  - core\_system\_parameters, [302](#)
- GPS\_L5Q\_NH\_CODE\_STR
  - core\_system\_parameters, [302](#)
- GPS\_L5Q\_PERIOD\_S
  - core\_system\_parameters, [302](#)
- Gps\_Navigation\_Message, [958](#)
  - get\_almanac, [959](#)
  - get\_ephemeris, [959](#)
  - get\_flag\_iono\_valid, [959](#)
  - get\_flag\_utc\_model\_valid, [959](#)
  - get\_GPS\_week, [959](#)
  - get\_iono, [959](#)
  - get\_satellite\_PRN, [960](#)
  - get\_TOW, [960](#)
  - get\_utc\_model, [960](#)
- Gps\_Navigation\_Message, [959](#)
  - set\_channel, [960](#)
  - set\_satellite\_PRN, [960](#)
  - subframe\_decoder, [960](#)
  - utc\_time, [960](#)
- gps\_navigation\_message.h, [2701](#), [2703](#)
- gps\_ref\_loc
  - Gnss\_Sdr\_Supl\_Client, [891](#)
- gps\_satellites
  - sdr\_gnss\_packet\_t, [1408](#)
- gps\_sdr\_signal\_replica.h, [1767](#), [1768](#)
- GPS\_SUBFRAME\_BITS
  - core\_system\_parameters, [302](#)
- GPS\_SUBFRAME\_LENGTH
  - core\_system\_parameters, [303](#)
- GPS\_SUBFRAME\_MS
  - core\_system\_parameters, [303](#)
- GPS\_SUBFRAME\_SECONDS
  - core\_system\_parameters, [303](#)
- gps\_time
  - Gnss\_Sdr\_Supl\_Client, [891](#)
- gps\_utc
  - Gnss\_Sdr\_Supl\_Client, [891](#)
- Gps\_Utc\_Model, [960](#)
  - A0, [963](#)
  - A1, [963](#)
  - A2, [963](#)
  - DeltaT\_LS, [963](#)
  - DeltaT\_LSF, [963](#)
  - DN, [963](#)
  - Gps\_Utc\_Model, [963](#)
  - serialize, [963](#)
  - tot, [963](#)
  - valid, [964](#)
  - WN\_LSF, [964](#)
  - WN\_T, [964](#)
- gps\_utc\_model
  - Rtklib\_Solver, [1374](#)
- gps\_utc\_model.h, [2706](#), [2707](#)
- GPS\_WORD\_BITS
  - core\_system\_parameters, [303](#)
- GPS\_WORD\_LENGTH
  - core\_system\_parameters, [303](#)
- gpsglob
  - exterr\_t, [529](#)
- GpsL1CaDIIPIITracking, [964](#)
  - connect, [967](#)
  - disconnect, [967](#)
  - get\_left\_block, [967](#)
  - get\_right\_block, [967](#)
  - implementation, [967](#)
  - item\_size, [967](#)
  - role, [968](#)
  - set\_channel, [968](#)
  - set\_gnss\_synchro, [968](#)
  - start\_tracking, [968](#)
  - stop\_tracking, [968](#)
- GpsL1CaDIIPIITrackingFpga, [968](#)
  - ~GpsL1CaDIIPIITrackingFpga, [972](#)
  - connect, [972](#)
  - disconnect, [972](#)
  - get\_left\_block, [972](#)
  - get\_right\_block, [972](#)

- GpsL1CaDIIPIITrackingFpga, 971
  - implementation, 972
  - item\_size, 972
  - role, 972
  - set\_channel, 972
  - set\_gnss\_synchro, 972
  - start\_tracking, 973
  - stop\_tracking, 973
- GpsL1CaDIIPIITrackingGPU, 973
  - connect, 976
  - disconnect, 976
  - get\_left\_block, 976
  - get\_right\_block, 976
  - implementation, 976
  - item\_size, 976
  - role, 977
  - set\_channel, 977
  - set\_gnss\_synchro, 977
  - start\_tracking, 977
  - stop\_tracking, 977
- GpsL1CaGaussianTracking, 977
  - connect, 980
  - disconnect, 980
  - get\_left\_block, 980
  - get\_right\_block, 980
  - implementation, 980
  - item\_size, 980
  - role, 981
  - set\_channel, 981
  - set\_gnss\_synchro, 981
  - start\_tracking, 981
  - stop\_tracking, 981
- GpsL1CaKfTracking, 981
  - connect, 984
  - disconnect, 984
  - get\_left\_block, 984
  - get\_right\_block, 984
  - implementation, 984
  - item\_size, 984
  - role, 985
  - set\_channel, 985
  - set\_gnss\_synchro, 985
  - start\_tracking, 985
  - stop\_tracking, 985
- GpsL1CaPcpsAcquisition, 985
  - connect, 989
  - disconnect, 989
  - get\_left\_block, 989
  - get\_right\_block, 989
  - implementation, 989
  - init, 989
  - item\_size, 989
  - mag, 989
  - reset, 989
  - role, 989
  - set\_channel, 989
  - set\_channel\_fsm, 990
  - set\_doppler\_center, 990
  - set\_doppler\_max, 990
  - set\_doppler\_step, 990
  - set\_gnss\_synchro, 990
  - set\_local\_code, 990
  - set\_resampler\_latency, 990
  - set\_state, 990
  - set\_threshold, 991
  - stop\_acquisition, 991
- GpsL1CaPcpsAcquisitionFineDoppler, 991
  - get\_left\_block, 995
  - get\_right\_block, 995
  - implementation, 995
  - init, 995
  - item\_size, 995
  - mag, 995
  - reset, 995
  - role, 995
  - set\_channel, 995
  - set\_channel\_fsm, 995
  - set\_doppler\_max, 996
  - set\_doppler\_step, 996
  - set\_gnss\_synchro, 996
  - set\_local\_code, 996
  - set\_resampler\_latency, 996
  - set\_state, 996
  - set\_threshold, 996
  - stop\_acquisition, 996
- GpsL1CaPcpsAcquisitionFpga, 997
  - ~GpsL1CaPcpsAcquisitionFpga, 1000
  - connect, 1000
  - disconnect, 1000
  - get\_left\_block, 1000
  - get\_right\_block, 1000
  - GpsL1CaPcpsAcquisitionFpga, 1000
  - implementation, 1001
  - init, 1001
  - item\_size, 1001
  - item\_type\_, 1003
  - mag, 1001
  - reset, 1001
  - role, 1001
  - set\_channel, 1001
  - set\_channel\_fsm, 1001
  - set\_doppler\_center, 1002
  - set\_doppler\_max, 1002
  - set\_doppler\_step, 1002
  - set\_gnss\_synchro, 1002
  - set\_local\_code, 1002
  - set\_resampler\_latency, 1002
  - set\_state, 1002
  - set\_threshold, 1002
  - stop\_acquisition, 1002
- GpsL1CaPcpsAssistedAcquisition, 1003
  - connect, 1006
  - disconnect, 1006
  - get\_left\_block, 1007
  - get\_right\_block, 1007
  - implementation, 1007

- init, [1007](#)
- item\_size, [1007](#)
- mag, [1007](#)
- reset, [1007](#)
- role, [1007](#)
- set\_channel, [1007](#)
- set\_channel\_fsm, [1007](#)
- set\_doppler\_max, [1008](#)
- set\_doppler\_step, [1008](#)
- set\_gnss\_synchro, [1008](#)
- set\_local\_code, [1008](#)
- set\_resampler\_latency, [1008](#)
- set\_state, [1008](#)
- set\_threshold, [1008](#)
- stop\_acquisition, [1008](#)
- GpsL1CaPcpsOpenCIACquisition, [1009](#)
  - connect, [1011](#)
  - disconnect, [1011](#)
  - get\_left\_block, [1012](#)
  - get\_right\_block, [1012](#)
  - implementation, [1012](#)
  - init, [1012](#)
  - item\_size, [1012](#)
  - mag, [1012](#)
  - openc1\_ready, [1012](#)
  - reset, [1012](#)
  - role, [1012](#)
  - set\_channel, [1012](#)
  - set\_channel\_fsm, [1013](#)
  - set\_doppler\_max, [1013](#)
  - set\_doppler\_step, [1013](#)
  - set\_gnss\_synchro, [1013](#)
  - set\_local\_code, [1013](#)
  - set\_resampler\_latency, [1013](#)
  - set\_state, [1013](#)
  - set\_threshold, [1013](#)
  - stop\_acquisition, [1014](#)
- GpsL1CaPcpsQuickSyncAcquisition, [1014](#)
  - connect, [1017](#)
  - disconnect, [1017](#)
  - get\_left\_block, [1018](#)
  - get\_right\_block, [1018](#)
  - implementation, [1018](#)
  - init, [1018](#)
  - item\_size, [1018](#)
  - mag, [1018](#)
  - reset, [1018](#)
  - role, [1018](#)
  - set\_channel, [1018](#)
  - set\_channel\_fsm, [1018](#)
  - set\_doppler\_max, [1019](#)
  - set\_doppler\_step, [1019](#)
  - set\_gnss\_synchro, [1019](#)
  - set\_local\_code, [1019](#)
  - set\_resampler\_latency, [1019](#)
  - set\_state, [1019](#)
  - set\_threshold, [1019](#)
  - stop\_acquisition, [1019](#)
- GpsL1CaPcpsTongAcquisition, [1020](#)
  - connect, [1022](#)
  - disconnect, [1022](#)
  - get\_left\_block, [1023](#)
  - get\_right\_block, [1023](#)
  - implementation, [1023](#)
  - init, [1023](#)
  - item\_size, [1023](#)
  - mag, [1023](#)
  - reset, [1023](#)
  - role, [1023](#)
  - set\_channel, [1023](#)
  - set\_channel\_fsm, [1023](#)
  - set\_doppler\_max, [1024](#)
  - set\_doppler\_step, [1024](#)
  - set\_gnss\_synchro, [1024](#)
  - set\_local\_code, [1024](#)
  - set\_resampler\_latency, [1024](#)
  - set\_state, [1024](#)
  - set\_threshold, [1024](#)
  - stop\_acquisition, [1024](#)
- GpsL1CaTcpConnectorTracking, [1025](#)
  - connect, [1027](#)
  - disconnect, [1027](#)
  - get\_left\_block, [1027](#)
  - get\_right\_block, [1027](#)
  - implementation, [1027](#)
  - item\_size, [1027](#)
  - role, [1028](#)
  - set\_channel, [1028](#)
  - set\_gnss\_synchro, [1028](#)
  - start\_tracking, [1028](#)
  - stop\_tracking, [1028](#)
- GpsL1CaTelemetryDecoder, [1028](#)
  - connect, [1031](#)
  - disconnect, [1031](#)
  - get\_left\_block, [1031](#)
  - get\_right\_block, [1031](#)
  - implementation, [1031](#)
  - item\_size, [1031](#)
  - reset, [1031](#)
  - role, [1032](#)
  - set\_channel, [1032](#)
  - set\_satellite, [1032](#)
- GpsL2CTelemetryDecoder, [1032](#)
  - connect, [1035](#)
  - disconnect, [1035](#)
  - get\_left\_block, [1035](#)
  - get\_right\_block, [1035](#)
  - implementation, [1035](#)
  - item\_size, [1035](#)
  - reset, [1035](#)
  - role, [1036](#)
  - set\_channel, [1036](#)
  - set\_satellite, [1036](#)
- GpsL2MDIIPITracking, [1036](#)
  - connect, [1039](#)
  - disconnect, [1039](#)

- get\_left\_block, 1039
- get\_right\_block, 1039
- implementation, 1039
- item\_size, 1039
- role, 1040
- set\_channel, 1040
- set\_gnss\_synchro, 1040
- start\_tracking, 1040
- stop\_tracking, 1040
- GpsL2MDIIPITrackingFpga, 1040
  - connect, 1043
  - disconnect, 1043
  - get\_left\_block, 1043
  - get\_right\_block, 1043
  - implementation, 1043
  - item\_size, 1043
  - role, 1044
  - set\_channel, 1044
  - set\_gnss\_synchro, 1044
  - start\_tracking, 1044
  - stop\_tracking, 1044
- GpsL2MPcpsAcquisition, 1044
  - connect, 1048
  - disconnect, 1048
  - get\_left\_block, 1048
  - get\_right\_block, 1048
  - implementation, 1048
  - init, 1048
  - item\_size, 1048
  - mag, 1048
  - reset, 1048
  - role, 1048
  - set\_channel, 1048
  - set\_channel\_fsm, 1049
  - set\_doppler\_center, 1049
  - set\_doppler\_max, 1049
  - set\_doppler\_step, 1049
  - set\_gnss\_synchro, 1049
  - set\_local\_code, 1049
  - set\_resampler\_latency, 1049
  - set\_state, 1049
  - set\_threshold, 1050
  - stop\_acquisition, 1050
- GpsL2MPcpsAcquisitionFpga, 1050
  - connect, 1053
  - disconnect, 1053
  - get\_left\_block, 1054
  - get\_right\_block, 1054
  - implementation, 1054
  - init, 1054
  - item\_size, 1054
  - mag, 1054
  - reset, 1054
  - role, 1054
  - set\_channel, 1054
  - set\_channel\_fsm, 1054
  - set\_doppler\_max, 1055
  - set\_doppler\_step, 1055
  - set\_gnss\_synchro, 1055
  - set\_local\_code, 1055
  - set\_resampler\_latency, 1055
  - set\_state, 1055
  - set\_threshold, 1055
  - stop\_acquisition, 1055
- GpsL5DIIPITracking, 1056
  - connect, 1058
  - disconnect, 1058
  - get\_left\_block, 1058
  - get\_right\_block, 1058
  - implementation, 1058
  - item\_size, 1058
  - role, 1059
  - set\_channel, 1059
  - set\_gnss\_synchro, 1059
  - start\_tracking, 1059
  - stop\_tracking, 1059
- GpsL5DIIPITrackingFpga, 1059
  - ~GpsL5DIIPITrackingFpga, 1063
  - connect, 1063
  - disconnect, 1063
  - get\_left\_block, 1063
  - get\_right\_block, 1063
  - GpsL5DIIPITrackingFpga, 1062
  - implementation, 1063
  - item\_size, 1063
  - role, 1063
  - set\_channel, 1063
  - set\_gnss\_synchro, 1063
  - start\_tracking, 1064
  - stop\_tracking, 1064
- GpsL5iPcpsAcquisition, 1064
  - connect, 1068
  - disconnect, 1068
  - get\_left\_block, 1068
  - get\_right\_block, 1068
  - implementation, 1068
  - init, 1068
  - item\_size, 1068
  - mag, 1068
  - reset, 1068
  - role, 1068
  - set\_channel, 1068
  - set\_channel\_fsm, 1069
  - set\_doppler\_center, 1069
  - set\_doppler\_max, 1069
  - set\_doppler\_step, 1069
  - set\_gnss\_synchro, 1069
  - set\_local\_code, 1069
  - set\_resampler\_latency, 1069
  - set\_state, 1069
  - set\_threshold, 1070
  - stop\_acquisition, 1070
- GpsL5iPcpsAcquisitionFpga, 1070
  - ~GpsL5iPcpsAcquisitionFpga, 1074
  - connect, 1074
  - disconnect, 1074

- get\_left\_block, [1074](#)
- get\_right\_block, [1074](#)
- GpsL5IPcpsAcquisitionFpga, [1074](#)
- implementation, [1074](#)
- init, [1075](#)
- item\_size, [1075](#)
- mag, [1075](#)
- reset, [1075](#)
- role, [1075](#)
- set\_channel, [1075](#)
- set\_channel\_fsm, [1075](#)
- set\_doppler\_center, [1075](#)
- set\_doppler\_max, [1076](#)
- set\_doppler\_step, [1076](#)
- set\_gnss\_synchro, [1076](#)
- set\_local\_code, [1076](#)
- set\_resampler\_latency, [1076](#)
- set\_state, [1076](#)
- set\_threshold, [1076](#)
- stop\_acquisition, [1076](#)
- GpsL5TelemetryDecoder, [1077](#)
- connect, [1079](#)
- disconnect, [1079](#)
- get\_left\_block, [1079](#)
- get\_right\_block, [1079](#)
- implementation, [1079](#)
- item\_size, [1079](#)
- reset, [1079](#)
- role, [1080](#)
- set\_channel, [1080](#)
- set\_satellite, [1080](#)
- GPU\_Complex, [1080](#)
- GPU\_Complex, [1081](#)
- i, [1082](#)
- magnitude2, [1081](#)
- multiply\_acc, [1081](#)
- operator+, [1081](#)
- operator+=", [1081](#)
- operator\*, [1081](#)
- r, [1082](#)
- GPU\_Complex\_Short, [1082](#)
- GPU\_Complex\_Short, [1082](#)
- i, [1083](#)
- magnitude2, [1083](#)
- operator+, [1083](#)
- operator\*, [1083](#)
- r, [1083](#)
- gpx\_output\_enabled
  - Pvt\_Conf, [1310](#)
- gpx\_output\_path
  - Pvt\_Conf, [1310](#)
- Gpx\_Printer, [1083](#)
- gpx\_printer.h, [1896](#), [1897](#)
- gpx\_rate\_ms
  - Pvt\_Conf, [1310](#)
- Gr\_Complex\_Ip\_Packet\_Source, [1084](#)
- sptr, [1085](#)
- gr\_complex\_ip\_packet\_source.h, [2046](#), [2047](#)
- Gravity\_ECEF
  - algorithms\_libs, [108](#)
- great\_circle\_distance
  - algorithms\_libs, [108](#)
- GST\_START\_EPOCH
  - Osnma\_Helper, [1230](#)
- GST\_to\_UTC\_time
  - Galileo\_Utc\_Model, [658](#)
- gtime\_t, [1086](#)
  - sec, [1086](#)
  - time, [1086](#)
- half
  - ssat\_t, [1459](#)
- half\_cyc\_tag, [1087](#)
  - corr, [1087](#)
  - freq, [1087](#)
  - next, [1087](#)
  - sat, [1088](#)
  - te, [1088](#)
  - ts, [1088](#)
  - valid, [1088](#)
- HALF\_PI
  - core\_system\_parameters, [303](#)
- halfc
  - raw\_t, [1325](#)
- has\_clock\_correction\_m
  - eph\_t, [521](#)
- HAS\_clock\_corrections, [1088](#)
  - clock\_correction\_m, [1089](#)
  - valid\_until, [1089](#)
- has\_message\_string
  - Galileo\_HAS\_page, [637](#)
- HAS\_MSG\_CODE\_BIAS\_LENGTH
  - core\_system\_parameters, [303](#)
- HAS\_MSG\_CODE\_BIAS\_SCALE\_FACTOR
  - core\_system\_parameters, [303](#)
- HAS\_MSG\_DELTA\_CLOCK\_CORRECTION\_LENGTH
  - core\_system\_parameters, [303](#)
- HAS\_MSG\_DELTA\_CLOCK\_CORRECTION\_SUBSET\_LENGTH
  - core\_system\_parameters, [303](#)
- HAS\_MSG\_DELTA\_CLOCK\_MULTIPLIER\_LENGTH
  - core\_system\_parameters, [303](#)
- HAS\_MSG\_DELTA\_CLOCK\_MULTIPLIER\_SUBSET\_LENGTH
  - core\_system\_parameters, [304](#)
- HAS\_MSG\_DELTA\_CLOCK\_SCALE\_FACTOR
  - core\_system\_parameters, [304](#)
- HAS\_MSG\_DELTA\_CROSS\_TRACK\_LENGTH
  - core\_system\_parameters, [304](#)
- HAS\_MSG\_DELTA\_CROSS\_TRACK\_SCALE\_FACTOR
  - core\_system\_parameters, [304](#)
- HAS\_MSG\_DELTA\_IN\_TRACK\_LENGTH
  - core\_system\_parameters, [304](#)
- HAS\_MSG\_DELTA\_IN\_TRACK\_SCALE\_FACTOR
  - core\_system\_parameters, [304](#)
- HAS\_MSG\_DELTA\_RADIAL\_LENGTH
  - core\_system\_parameters, [304](#)
- HAS\_MSG\_DELTA\_RADIAL\_SCALE\_FACTOR
  - core\_system\_parameters, [304](#)

- HAS\_MSG\_GALILEO\_SYSTEM
  - core\_system\_parameters, [304](#)
- HAS\_MSG\_GPS\_SYSTEM
  - core\_system\_parameters, [304](#)
- HAS\_MSG\_ID\_CLOCK\_SUBSET\_LENGTH
  - core\_system\_parameters, [304](#)
- HAS\_MSG\_ID\_MASK\_LENGTH
  - core\_system\_parameters, [304](#)
- HAS\_MSG\_IOD\_GAL\_LENGTH
  - core\_system\_parameters, [305](#)
- HAS\_MSG\_IOD\_GPS\_LENGTH
  - core\_system\_parameters, [305](#)
- HAS\_MSG\_MASK\_RESERVED\_LENGTH
  - core\_system\_parameters, [305](#)
- HAS\_MSG\_NAV\_MESSAGE\_LENGTH
  - core\_system\_parameters, [305](#)
- HAS\_MSG\_NSYS\_LENGTH
  - core\_system\_parameters, [305](#)
- HAS\_MSG\_NSYS\_SUB\_LENGTH
  - core\_system\_parameters, [305](#)
- HAS\_MSG\_NUMBER\_GNSS\_IDS
  - core\_system\_parameters, [305](#)
- HAS\_MSG\_NUMBER\_MASK\_IDS
  - core\_system\_parameters, [305](#)
- HAS\_MSG\_NUMBER\_MAX\_TOH
  - core\_system\_parameters, [305](#)
- HAS\_MSG\_NUMBER\_MESSAGE\_IDS
  - core\_system\_parameters, [305](#)
- HAS\_MSG\_NUMBER\_SATELLITE\_IDS
  - core\_system\_parameters, [305](#)
- HAS\_MSG\_NUMBER\_SIGNAL\_MASKS
  - core\_system\_parameters, [305](#)
- HAS\_MSG\_PHASE\_BIAS\_LENGTH
  - core\_system\_parameters, [306](#)
- HAS\_MSG\_PHASE\_BIAS\_SCALE\_FACTOR
  - core\_system\_parameters, [306](#)
- HAS\_MSG\_PHASE\_DISCONTINUITY\_INDICATOR\_LENGTH
  - core\_system\_parameters, [306](#)
- HAS\_MSG\_SATELLITE\_MASK\_LENGTH
  - core\_system\_parameters, [306](#)
- HAS\_MSG\_SIGNAL\_MASK\_LENGTH
  - core\_system\_parameters, [306](#)
- HAS\_MSG\_VALIDITY\_INDEX\_LENGTH
  - core\_system\_parameters, [306](#)
- HAS\_MSG\_WRONG\_SYSTEM
  - core\_system\_parameters, [306](#)
- HAS\_obs\_corrections, [1089](#)
  - code\_bias\_m, [1089](#)
  - phase\_bias\_cycle, [1089](#)
- HAS\_orbit\_corrections, [1090](#)
  - cross\_track\_m, [1090](#)
  - in\_track\_m, [1090](#)
  - iod, [1090](#)
  - radial\_m, [1090](#)
  - valid\_until, [1090](#)
- has\_orbit\_cross\_track\_correction\_m
  - eph\_t, [522](#)
- has\_orbit\_in\_track\_correction\_m
  - eph\_t, [522](#)
- has\_orbit\_radial\_correction\_m
  - eph\_t, [522](#)
- has\_output\_file\_path
  - Pvt\_Conf, [1310](#)
- HAS\_SIGNAL\_INDEX\_TABLE
  - core\_system\_parameters, [306](#)
- Has\_Simple\_Printer, [1091](#)
- has\_simple\_printer.h, [1897](#), [1898](#)
- has\_status
  - Galileo\_HAS\_data, [634](#)
  - Galileo\_HAS\_page, [637](#)
- HAS\_VALIDITY\_INTERVALS
  - core\_system\_parameters, [306](#)
- HasSection
  - INIReader, [1108](#)
- HasValue
  - INIReader, [1108](#)
- have\_new\_almanac
  - Beidou\_Dnav\_Navigation\_Message, [390](#)
  - Glionass\_Gnav\_Navigation\_Message, [807](#)
- have\_new\_ephemeris
  - Beidou\_Dnav\_Navigation\_Message, [390](#)
  - Glionass\_Gnav\_Navigation\_Message, [807](#)
  - Gps\_CNAV\_Navigation\_Message, [933](#)
- have\_new\_HAS\_page
  - Galileo\_Cnav\_Message, [606](#)
- have\_new\_iono
  - Beidou\_Dnav\_Navigation\_Message, [390](#)
  - Gps\_CNAV\_Navigation\_Message, [933](#)
- have\_new\_utc\_model
  - Beidou\_Dnav\_Navigation\_Message, [390](#)
  - Glionass\_Gnav\_Navigation\_Message, [807](#)
  - Gps\_CNAV\_Navigation\_Message, [933](#)
- have\_public\_key
  - Gnss\_Crypto, [869](#)
- hbb
  - Monitor\_Pvt, [1173](#)
- header
  - an\_packet\_t, [359](#)
  - Galileo\_HAS\_data, [634](#)
  - MACK\_message, [1160](#)
- health
  - lexeph\_t, [1149](#)
- height
  - Monitor\_Pvt, [1173](#)
  - sdr\_gnss\_packet\_t, [1408](#)
  - solopt\_t, [1443](#)
- hex\_to\_bin
  - Rtcm, [1341](#)
- hex\_to\_binary\_converter
  - algorithms\_libs, [108](#)
- hex\_to\_binary\_string
  - algorithms\_libs, [109](#)
- hex\_to\_int
  - Rtcm, [1341](#)
- hex\_to\_uint
  - Rtcm, [1341](#)



- hf
  - DSM\_KROOT\_message, [514](#)
- hgt
  - sta\_t, [1465](#)
- hgts
  - tec\_t, [1482](#)
- high\_dyn
  - Dll\_Pll\_Conf, [495](#)
  - Dll\_Pll\_Conf\_Fpga, [502](#)
  - Kf\_Conf, [1137](#)
- HION
  - algorithms\_libs\_rtklib, [144](#)
- hkroot
  - OSNMA\_msg, [1231](#)
- hrclk
  - ssr\_t, [1462](#)
- hybrid\_observables.h, [1875](#), [1876](#)
- hybrid\_observables\_gs, [1091](#)
- hybrid\_observables\_gs.h, [1877](#), [1878](#)
- hybrid\_observables\_gs\_sptr
  - obs\_gr\_blocks, [174](#)
- HybridObservables, [1093](#)
  - connect, [1096](#)
  - disconnect, [1096](#)
  - get\_left\_block, [1096](#)
  - get\_right\_block, [1096](#)
  - implementation, [1096](#)
  - item\_size, [1096](#)
  - reset, [1096](#)
  - role, [1097](#)
- i
  - GPU\_Complex, [1082](#)
  - GPU\_Complex\_Short, [1083](#)
- i0
  - alm\_t, [354](#)
  - eph\_t, [522](#)
- i\_0
  - Gnss\_Ephemeris, [875](#)
- I\_0\_2\_LSB
  - core\_system\_parameters, [307](#)
- I\_0\_LSB
  - core\_system\_parameters, [307](#)
- I\_DOT\_2\_LSB
  - core\_system\_parameters, [307](#)
- I\_DOT\_LSB
  - core\_system\_parameters, [307](#)
- i\_prn
  - Sbas\_Ephemeris, [1387](#)
- i\_satellite\_freq\_channel
  - Glonass\_Gnav\_Almanac, [794](#)
  - Glonass\_Gnav\_Ephemeris, [804](#)
- i\_satellite\_slot\_number
  - Glonass\_Gnav\_Almanac, [794](#)
  - Glonass\_Gnav\_Ephemeris, [804](#)
- i\_sv\_ura
  - Sbas\_Ephemeris, [1388](#)
- i\_t0
  - Sbas\_Ephemeris, [1388](#)
- IB\_PPP
  - algorithms\_libs\_rtklib, [131](#)
- IB\_RTK
  - algorithms\_libs\_rtklib, [131](#)
- ibyte\_to\_cbyte.h, [1683](#), [1684](#)
- ibyte\_to\_complex.h, [1685](#), [1686](#)
- ibyte\_to\_cshort.h, [1687](#), [1688](#)
- lbyteToCbyte, [1097](#)
  - connect, [1100](#)
  - disconnect, [1100](#)
  - get\_left\_block, [1100](#)
  - get\_right\_block, [1100](#)
  - implementation, [1100](#)
  - item\_size, [1100](#)
  - role, [1100](#)
- lbyteToComplex, [1101](#)
  - connect, [1103](#)
  - disconnect, [1103](#)
  - get\_left\_block, [1103](#)
  - get\_right\_block, [1103](#)
  - implementation, [1103](#)
  - item\_size, [1103](#)
  - role, [1103](#)
- lbyteToCshort, [1104](#)
  - connect, [1106](#)
  - disconnect, [1106](#)
  - get\_left\_block, [1106](#)
  - get\_right\_block, [1106](#)
  - implementation, [1106](#)
  - item\_size, [1106](#)
  - role, [1106](#)
- IC\_PPP
  - algorithms\_libs\_rtklib, [131](#)
- icpc
  - raw\_t, [1325](#)
- icpp
  - raw\_t, [1325](#)
- id\_counter
  - Tag, [1474](#)
- idot
  - eph\_t, [522](#)
  - Gnss\_Ephemeris, [875](#)
- igp
  - sbsion\_t, [1400](#)
- IGPBAND1
  - rtklib\_sbas.h, [1857](#)
- IGPBAND2
  - rtklib\_sbas.h, [1857](#)
- II\_RTK
  - algorithms\_libs\_rtklib, [131](#)
- IIO\_DEFAULTAD936XAPIFIFOSIZE\_SAMPLES
  - ad936x\_iio\_samples.h, [2076](#)
- IIO\_INPUTTRAMFIFOSIZE
  - ad936x\_iio\_samples.h, [2076](#)
- IL\_RTK
  - algorithms\_libs\_rtklib, [132](#)
- imag
  - cIFFT\_Complex, [444](#)



- cIFFT\_SplitComplex, [446](#)
- implementation
  - ArraySignalConditioner, [363](#)
  - BeamformerFilter, [368](#)
  - BeidouB1iDIIPIITracking, [396](#)
  - BeidouB1iPcpsAcquisition, [401](#)
  - BeidouB1iTelemetryDecoder, [406](#)
  - BeidouB3iDIIPIITracking, [410](#)
  - BeidouB3iPcpsAcquisition, [415](#)
  - BeidouB3iTelemetryDecoder, [420](#)
  - ByteToShort, [425](#)
  - Channel, [430](#)
  - CshortToGrComplex, [475](#)
  - DirectResamplerConditioner, [491](#)
  - FirFilter, [560](#)
  - FreqXlatingFirFilter, [593](#)
  - GalileoE1BTelemetryDecoder, [663](#)
  - GalileoE1DIIPIIVemITracking, [667](#)
  - GalileoE1DIIPIIVemITrackingFpga, [672](#)
  - GalileoE1Pcps8msAmbiguousAcquisition, [677](#)
  - GalileoE1PcpsAmbiguousAcquisition, [683](#)
  - GalileoE1PcpsAmbiguousAcquisitionFpga, [689](#)
  - GalileoE1PcpsCccwsrAmbiguousAcquisition, [695](#)
  - GalileoE1PcpsQuickSyncAmbiguousAcquisition, [701](#)
  - GalileoE1PcpsTongAmbiguousAcquisition, [707](#)
  - GalileoE1TcpConnectorTracking, [712](#)
  - GalileoE5aDIIPIITracking, [716](#)
  - GalileoE5aDIIPIITrackingFpga, [721](#)
  - GalileoE5aNoncoherentLQAcquisitionCaf, [726](#)
  - GalileoE5aPcpsAcquisition, [732](#)
  - GalileoE5aPcpsAcquisitionFpga, [739](#)
  - GalileoE5aTelemetryDecoder, [744](#)
  - GalileoE5bDIIPIITracking, [748](#)
  - GalileoE5bPcpsAcquisition, [753](#)
  - GalileoE5bPcpsAcquisitionFpga, [761](#)
  - GalileoE5bTelemetryDecoder, [766](#)
  - GalileoE6DIIPIITracking, [770](#)
  - GalileoE6PcpsAcquisition, [775](#)
  - GalileoE6TelemetryDecoder, [780](#)
  - GenSignalSource, [784](#)
  - GlonassL1CaDIIPIICAidTracking, [828](#)
  - GlonassL1CaDIIPIITracking, [832](#)
  - GlonassL1CaPcpsAcquisition, [837](#)
  - GlonassL1CaTelemetryDecoder, [841](#)
  - GlonassL2CaDIIPIICAidTracking, [845](#)
  - GlonassL2CaDIIPIITracking, [849](#)
  - GlonassL2CaPcpsAcquisition, [854](#)
  - GlonassL2CaTelemetryDecoder, [858](#)
  - GNSSBlockInterface, [910](#)
  - GpsL1CaDIIPIITracking, [967](#)
  - GpsL1CaDIIPIITrackingFpga, [972](#)
  - GpsL1CaDIIPIITrackingGPU, [976](#)
  - GpsL1CaGaussianTracking, [980](#)
  - GpsL1CaKfTracking, [984](#)
  - GpsL1CaPcpsAcquisition, [989](#)
  - GpsL1CaPcpsAcquisitionFineDoppler, [995](#)
  - GpsL1CaPcpsAcquisitionFpga, [1001](#)
  - GpsL1CaPcpsAssistedAcquisition, [1007](#)
  - GpsL1CaPcpsOpenCIAcquisition, [1012](#)
  - GpsL1CaPcpsQuickSyncAcquisition, [1018](#)
  - GpsL1CaPcpsTongAcquisition, [1023](#)
  - GpsL1CaTcpConnectorTracking, [1027](#)
  - GpsL1CaTelemetryDecoder, [1031](#)
  - GpsL2CTelemetryDecoder, [1035](#)
  - GpsL2MDIIPIITracking, [1039](#)
  - GpsL2MDIIPIITrackingFpga, [1043](#)
  - GpsL2MPcpsAcquisition, [1048](#)
  - GpsL2MPcpsAcquisitionFpga, [1054](#)
  - GpsL5DIIPIITracking, [1058](#)
  - GpsL5DIIPIITrackingFpga, [1063](#)
  - GpsL5iPcpsAcquisition, [1068](#)
  - GpsL5iPcpsAcquisitionFpga, [1074](#)
  - GpsL5TelemetryDecoder, [1079](#)
  - HybridObservables, [1096](#)
  - lbyteToCbyte, [1100](#)
  - lbyteToComplex, [1103](#)
  - lbyteToCshort, [1106](#)
  - lshortToComplex, [1128](#)
  - lshortToCshort, [1131](#)
  - MmseResamplerConditioner, [1168](#)
  - NotchFilter, [1201](#)
  - NotchFilterLite, [1204](#)
  - Pass\_Through, [1240](#)
  - PulseBlankingFilter, [1306](#)
  - Rtklib\_Pvt, [1364](#)
  - SbasL1TelemetryDecoder, [1393](#)
  - SignalConditioner, [1428](#)
  - SignalGenerator, [1431](#)
  - SignalSourceBase, [1435](#)
  - in\_memory\_configuration.h, [2395](#), [2396](#)
  - in\_place\_possible
    - kernel\_info\_t, [1132](#)
  - in\_track\_m
    - HAS\_orbit\_corrections, [1090](#)
  - INAV\_RS\_BUFFER\_LENGTH
    - core\_system\_parameters, [307](#)
  - INAV\_RS\_INFO\_VECTOR\_LENGTH
    - core\_system\_parameters, [307](#)
  - INAV\_RS\_PARITY\_VECTOR\_LENGTH
    - core\_system\_parameters, [307](#)
  - INAV\_RS\_SUBVECTOR\_LENGTH
    - core\_system\_parameters, [307](#)
  - inb
    - stream\_t, [1469](#)
  - inbt
    - stream\_t, [1469](#)
  - inc
    - tled\_t, [1488](#)
  - index
    - pclk\_t, [1241](#)
    - peph\_t, [1288](#)
  - ini.h, [2349](#), [2350](#)
  - INI\_ALLOW\_MULTILINE
    - core\_libs, [209](#)
  - ini\_parse

- core\_libs, 210
- INIRReader, 1107
  - Get, 1108
  - GetInteger, 1108
  - HasSection, 1108
  - HasValue, 1108
  - INIRReader, 1107
  - ParseError, 1108
- INIRReader.h, 2351, 2352
- init
  - AcquisitionInterface, 334
  - BeidouB1iPcpsAcquisition, 401
  - BeidouB3iPcpsAcquisition, 415
  - cnav\_v27\_part\_t, 451
  - FileSourceBase, 552
  - Fpga\_Acquisition, 578
  - galileo\_e5a\_noncoherentIQ\_acquisition\_caf\_cc, 611
  - galileo\_pcps\_8ms\_acquisition\_cc, 648
  - GalileoE1Pcps8msAmbiguousAcquisition, 677
  - GalileoE1PcpsAmbiguousAcquisition, 683
  - GalileoE1PcpsAmbiguousAcquisitionFpga, 690
  - GalileoE1PcpsCccwsrAmbiguousAcquisition, 695
  - GalileoE1PcpsQuickSyncAmbiguousAcquisition, 701
  - GalileoE1PcpsTongAmbiguousAcquisition, 707
  - GalileoE5aNoncoherentIQAcquisitionCaf, 726
  - GalileoE5aPcpsAcquisition, 732
  - GalileoE5aPcpsAcquisitionFpga, 739
  - GalileoE5bPcpsAcquisition, 754
  - GalileoE5bPcpsAcquisitionFpga, 761
  - GalileoE6PcpsAcquisition, 775
  - GlonassL1CaPcpsAcquisition, 837
  - GlonassL2CaPcpsAcquisition, 854
  - GpsL1CaPcpsAcquisition, 989
  - GpsL1CaPcpsAcquisitionFineDoppler, 995
  - GpsL1CaPcpsAcquisitionFpga, 1001
  - GpsL1CaPcpsAssistedAcquisition, 1007
  - GpsL1CaPcpsOpenCIAcquisition, 1012
  - GpsL1CaPcpsQuickSyncAcquisition, 1018
  - GpsL1CaPcpsTongAcquisition, 1023
  - GpsL2MPcpsAcquisition, 1048
  - GpsL2MPcpsAcquisitionFpga, 1054
  - GpsL5iPcpsAcquisition, 1068
  - GpsL5iPcpsAcquisitionFpga, 1075
  - pcps\_acquisition, 1244
  - pcps\_acquisition\_fine\_doppler\_cc, 1249
  - pcps\_acquisition\_fpga, 1254
  - pcps\_assisted\_acquisition\_cc, 1259
  - pcps\_cccwsr\_acquisition\_cc, 1264
  - pcps\_openc1\_acquisition\_cc, 1269
  - pcps\_quicksync\_acquisition\_cc, 1275
  - pcps\_tong\_acquisition\_cc, 1281
- init\_carrier\_freq\_rate\_sd\_hz\_s
  - Kf\_Conf, 1137
- init\_carrier\_freq\_sd\_hz
  - Kf\_Conf, 1137
- init\_carrier\_phase\_sd\_rad
  - Kf\_Conf, 1137
- init\_code\_phase\_sd\_chips
  - Kf\_Conf, 1137
- init\_PRN
  - Galileo\_Inav\_Message, 640
- INIT\_ZWD
  - algorithms\_libs\_rtklib, 144
- initialize
  - Tlm\_CRC\_Stats, 1492
  - Tracking\_2nd\_DLL\_filter, 1494
- initialize\_secondary\_code
  - Fpga\_Multicorrelator\_8sc, 586
- initlock
  - algorithms\_libs\_rtklib, 132
- initrst
  - prcopt\_t, 1299
- InMemoryConfiguration, 1108
  - property, 1111, 1112
  - set\_property, 1112
- Input Filter, 87
- input\_filter
  - ArraySignalConditioner, 363
  - SignalConditioner, 1428
- input\_filter\_adapters, 87
- input\_filter\_gr\_blocks, 88
  - beamformer\_sptr, 89
  - GNSS\_SDR\_BEAMFORMER\_CHANNELS, 89
  - notch\_lite\_sptr, 89
  - notch\_sptr, 89
  - pulse\_blanking\_cc\_sptr, 89
- inr
  - stream\_t, 1469
- INT\_SWAP\_STAT
  - algorithms\_libs\_rtklib, 145
- INT\_SWAP\_TRAC
  - algorithms\_libs\_rtklib, 145
- integrity\_status\_flag
  - Beidou\_Dnav\_Ephemeris, 383
  - Gps\_CNAV\_Ephemeris, 928
  - Gps\_Ephemeris, 942
- interleaved\_byte\_to\_complex\_byte, 1112
- interleaved\_byte\_to\_complex\_byte.h, 1694, 1695
- interleaved\_byte\_to\_complex\_byte\_sptr
  - data\_type\_gr\_blocks, 86
- interleaved\_byte\_to\_complex\_short, 1114
- interleaved\_byte\_to\_complex\_short.h, 1696, 1697
- interleaved\_byte\_to\_complex\_short\_sptr
  - data\_type\_gr\_blocks, 86
- interleaved\_short\_to\_complex\_short, 1115
- interleaved\_short\_to\_complex\_short.h, 1698, 1699
- interleaved\_short\_to\_complex\_short\_sptr
  - data\_type\_gr\_blocks, 86
- interp\_TOW\_ms
  - Gnss\_Synchro, 904
- intpref
  - prcopt\_t, 1299
- invert
  - cnav\_v27\_part\_t, 451

- iod
  - dgps\_t, [483](#)
  - HAS\_orbit\_corrections, [1090](#)
  - msm\_h\_t, [1178](#)
  - raw\_t, [1325](#)
  - ssr\_t, [1462](#)
- IOD\_a\_10
  - Galileo\_Almanac\_Helper, [603](#)
- IOD\_a\_7
  - Galileo\_Almanac\_Helper, [603](#)
- IOD\_a\_8
  - Galileo\_Almanac\_Helper, [603](#)
- IOD\_a\_9
  - Galileo\_Almanac\_Helper, [603](#)
- IOD\_ephemeris
  - Galileo\_Ephemeris, [622](#)
- IOD\_nav
  - Galileo\_Ephemeris, [622](#)
- iod\_set\_id
  - mt1\_header, [1180](#)
- IODa
  - Galileo\_Almanac, [599](#)
- IODC
  - Gps\_Ephemeris, [942](#)
- iodc
  - eph\_t, [522](#)
- iodcrc
  - ssr\_t, [1462](#)
- iode
  - eph\_t, [522](#)
  - geph\_t, [789](#)
  - sbslcorr\_t, [1402](#)
  - ssr\_t, [1462](#)
- IODE\_SF2
  - Gps\_Ephemeris, [942](#)
- IODE\_SF3
  - Gps\_Ephemeris, [943](#)
- iodev
  - signal\_source\_libs, [185](#)
- iodf
  - sbsfcorr\_t, [1396](#)
- iodi
  - sbsion\_t, [1400](#)
- IODnav
  - Galileo\_Reduced\_CED, [652](#)
- iodp
  - sbssat\_t, [1405](#)
- ion
  - stec\_t, [1466](#)
- ion\_cmp
  - nav\_t, [1192](#)
- ion\_gal
  - nav\_t, [1192](#)
- ion\_gps
  - nav\_t, [1192](#)
- ion\_gsms.h, [2049](#), [2050](#)
- ion\_gsms\_chunk\_data.h, [2092](#), [2094](#)
  - bits\_to\_item\_size, [2094](#)
  - with\_word\_type, [2094](#)
- ion\_gsms\_chunk\_unpacking\_ctx.h, [2096](#), [2097](#)
- ion\_gsms\_signal\_source.h, [2010](#)
- ion\_gsms\_stream\_encodings.h, [2099](#), [2104](#)
  - encoding\_from\_string, [2101](#)
  - five\_bit\_look\_up, [2101](#)
  - four\_bit\_look\_up, [2102](#)
  - FP, [2102](#)
  - MS, [2102](#)
  - MSA, [2102](#)
  - OB, [2102](#)
  - OBA, [2102](#)
  - OG, [2102](#)
  - OGA, [2102](#)
  - SIGN, [2102](#)
  - SM, [2103](#)
  - SMA, [2103](#)
  - StreamEncoding, [2101](#)
  - TC, [2103](#)
  - TCA, [2103](#)
  - three\_bit\_look\_up, [2103](#)
  - two\_bit\_look\_up, [2103](#)
- ion\_irn
  - nav\_t, [1192](#)
- ion\_qzs
  - nav\_t, [1193](#)
- IONGSMSChunkData, [1117](#)
- IONGSMSChunkUnpackingCtx
  - IONGSMSChunkUnpackingCtx< WT >, [1119](#)
- IONGSMSChunkUnpackingCtx< WT >, [1118](#)
  - advance\_word, [1119](#)
  - bitshift\_, [1119](#)
  - current\_word\_, [1119](#)
  - IONGSMSChunkUnpackingCtx, [1119](#)
  - iterator\_, [1119](#)
  - shift\_current\_word, [1119](#)
  - shift\_padding, [1119](#)
  - shift\_sample, [1119](#)
  - word\_bitsize\_, [1120](#)
  - word\_shift\_direction\_, [1120](#)
- IONGSMSFileSource, [1120](#)
  - sptr, [1121](#)
- IONGSMSSignalSource, [1122](#)
  - connect, [1124](#)
  - disconnect, [1124](#)
  - get\_left\_block, [1124](#)
  - get\_right\_block, [1124](#)
  - item\_size, [1125](#)
- ionoopt
  - prcopt\_t, [1299](#)
- IONOOPT\_BRDC
  - algorithms\_libs\_rtklib, [145](#)
- IONOOPT\_EST
  - algorithms\_libs\_rtklib, [145](#)
- IONOOPT\_IFLC
  - algorithms\_libs\_rtklib, [145](#)
- IONOOPT\_LEX
  - algorithms\_libs\_rtklib, [145](#)

- IONOOPT\_OFF
  - algorithms\_libs\_rtklib, 146
- IONOOPT\_QZS
  - algorithms\_libs\_rtklib, 146
- IONOOPT\_SBAS
  - algorithms\_libs\_rtklib, 146
- IONOOPT\_STEC
  - algorithms\_libs\_rtklib, 146
- IONOOPT\_TEC
  - algorithms\_libs\_rtklib, 146
- is\_connected
  - pps\_tcp\_rx, 1294
- is\_flag\_TOW\_set
  - Glonass\_Gnav\_Navigation\_Message, 807
- is\_HAS\_in\_test\_mode
  - Galileo\_Cnav\_Message, 606
- is\_HAS\_page\_dummy
  - Galileo\_Cnav\_Message, 607
- is\_rinex\_header\_written
  - Rinex\_Printer, 1334
- is\_server\_running
  - Rtcm, 1341
- is\_TOW0\_set
  - Galileo\_Inav\_Message, 640
- is\_TOW1\_set
  - Galileo\_Fnav\_Message, 624
- is\_TOW2\_set
  - Galileo\_Fnav\_Message, 624
- is\_TOW3\_set
  - Galileo\_Fnav\_Message, 625
- is\_TOW4\_set
  - Galileo\_Fnav\_Message, 625
- is\_TOW5\_set
  - Galileo\_Inav\_Message, 640
- is\_TOW6\_set
  - Galileo\_Inav\_Message, 640
- isc
  - eph\_t, 522
  - lexeph\_t, 1149
- ISCL1
  - Gps\_CNAV\_Ephemeris, 928
- ISCL2
  - Gps\_CNAV\_Ephemeris, 928
- ISCL5I
  - Gps\_CNAV\_Ephemeris, 929
- ISCL5Q
  - Gps\_CNAV\_Ephemeris, 929
- ishort\_to\_complex.h, 1689
- ishort\_to\_cshort.h, 1691
- IshortToComplex, 1125
  - connect, 1128
  - disconnect, 1128
  - get\_left\_block, 1128
  - get\_right\_block, 1128
  - implementation, 1128
  - item\_size, 1128
  - role, 1128
- IshortToCshort, 1129
  - connect, 1131
  - disconnect, 1131
  - get\_left\_block, 1131
  - get\_right\_block, 1131
  - implementation, 1131
  - item\_size, 1131
  - role, 1131
- IT\_PPP
  - algorithms\_libs\_rtklib, 132
- IT\_RTK
  - algorithms\_libs\_rtklib, 132
- it\_size
  - Acq\_Conf, 326
- item\_size
  - Ad936xCustomSignalSource, 345
  - Adrv9361z7035SignalSourceFPGA, 348
  - ArraySignalConditioner, 363
  - BeamformerFilter, 368
  - BeidouB1iDIPIITracking, 396
  - BeidouB1iPcpsAcquisition, 401
  - BeidouB1iTelemetryDecoder, 406
  - BeidouB3iDIPIITracking, 410
  - BeidouB3iPcpsAcquisition, 415
  - BeidouB3iTelemetryDecoder, 420
  - ByteToShort, 425
  - Channel, 430
  - CshortToGrComplex, 475
  - CustomUDPSignalSource, 481
  - DirectResamplerConditioner, 491
  - DMASignalSourceFPGA, 511
  - FifoSignalSource, 536
  - FileSourceBase, 552
  - FirFilter, 561
  - FlexibandSignalSource, 565
  - Fmcomms2SignalSource, 568
  - Fmcomms5SignalSourceFPGA, 571
  - FreqXlatingFirFilter, 593
  - GalileoE1BTelemetryDecoder, 663
  - GalileoE1DIPIIVemITracking, 667
  - GalileoE1DIPIIVemITrackingFpga, 672
  - GalileoE1Pcps8msAmbiguousAcquisition, 677
  - GalileoE1PcpsAmbiguousAcquisition, 683
  - GalileoE1PcpsAmbiguousAcquisitionFpga, 690
  - GalileoE1PcpsCccwsrAmbiguousAcquisition, 695
  - GalileoE1PcpsQuickSyncAmbiguousAcquisition, 701
  - GalileoE1PcpsTongAmbiguousAcquisition, 707
  - GalileoE1TcpConnectorTracking, 712
  - GalileoE5aDIPIITracking, 716
  - GalileoE5aDIPIITrackingFpga, 721
  - GalileoE5aNoncoherentIQAcquisitionCaf, 726
  - GalileoE5aPcpsAcquisition, 732
  - GalileoE5aPcpsAcquisitionFpga, 739
  - GalileoE5aTelemetryDecoder, 744
  - GalileoE5bDIPIITracking, 749
  - GalileoE5bPcpsAcquisition, 754
  - GalileoE5bPcpsAcquisitionFpga, 761
  - GalileoE5bTelemetryDecoder, 766

- GalileoE6DIIPIITracking, [771](#)
- GalileoE6PcpsAcquisition, [775](#)
- GalileoE6TelemetryDecoder, [780](#)
- GenSignalSource, [785](#)
- GlonassL1CaDIIPIICAidTracking, [828](#)
- GlonassL1CaDIIPIITracking, [832](#)
- GlonassL1CaPcpsAcquisition, [837](#)
- GlonassL1CaTelemetryDecoder, [841](#)
- GlonassL2CaDIIPIICAidTracking, [845](#)
- GlonassL2CaDIIPIITracking, [849](#)
- GlonassL2CaPcpsAcquisition, [854](#)
- GlonassL2CaTelemetryDecoder, [858](#)
- GNSSBlockInterface, [910](#)
- GpsL1CaDIIPIITracking, [967](#)
- GpsL1CaDIIPIITrackingFpga, [972](#)
- GpsL1CaDIIPIITrackingGPU, [976](#)
- GpsL1CaGaussianTracking, [980](#)
- GpsL1CaKfTracking, [984](#)
- GpsL1CaPcpsAcquisition, [989](#)
- GpsL1CaPcpsAcquisitionFineDoppler, [995](#)
- GpsL1CaPcpsAcquisitionFpga, [1001](#)
- GpsL1CaPcpsAssistedAcquisition, [1007](#)
- GpsL1CaPcpsOpenCIAcquisition, [1012](#)
- GpsL1CaPcpsQuickSyncAcquisition, [1018](#)
- GpsL1CaPcpsTongAcquisition, [1023](#)
- GpsL1CaTcpConnectorTracking, [1027](#)
- GpsL1CaTelemetryDecoder, [1031](#)
- GpsL2CTelemetryDecoder, [1035](#)
- GpsL2MDIIPIITracking, [1039](#)
- GpsL2MDIIPIITrackingFpga, [1043](#)
- GpsL2MPcpsAcquisition, [1048](#)
- GpsL2MPcpsAcquisitionFpga, [1054](#)
- GpsL5DIIPIITracking, [1058](#)
- GpsL5DIIPIITrackingFpga, [1063](#)
- GpsL5iPcpsAcquisition, [1068](#)
- GpsL5iPcpsAcquisitionFpga, [1075](#)
- GpsL5TelemetryDecoder, [1079](#)
- HybridObservables, [1096](#)
- lbyteToCbyte, [1100](#)
- lbyteToComplex, [1103](#)
- lbyteToCshort, [1106](#)
- IONGSMSSignalSource, [1125](#)
- lshortToComplex, [1128](#)
- lshortToCshort, [1131](#)
- LabsatSignalSource, [1146](#)
- LimesdrSignalSource, [1155](#)
- MAX2771EVKITSignalSourceFPGA, [1165](#)
- MmseResamplerConditioner, [1168](#)
- MultichannelFileSignalSource, [1184](#)
- NotchFilter, [1201](#)
- NotchFilterLite, [1204](#)
- OsmosdrSignalSource, [1225](#)
- Pass\_Through, [1240](#)
- PlutosdrSignalSource, [1291](#)
- PulseBlankingFilter, [1306](#)
- RawArraySignalSource, [1330](#)
- Rtllib\_Pvt, [1364](#)
- RtlTcpSignalSource, [1385](#)
- SbasL1TelemetryDecoder, [1393](#)
- SignalConditioner, [1428](#)
- SignalGenerator, [1431](#)
- SpirGSS6450FileSignalSource, [1457](#)
- UhdSignalSource, [1514](#)
- ZmqSignalSource, [1536](#)
- item\_type
  - Acq\_Conf, [326](#)
  - DII\_PII\_Conf, [495](#)
  - FileSourceBase, [552](#)
  - Kf\_Conf, [1138](#)
  - MultichannelFileSignalSource, [1184](#)
  - Pass\_Through, [1240](#)
  - SpirGSS6450FileSignalSource, [1457](#)
- item\_type\_
  - GpsL1CaPcpsAcquisitionFpga, [1003](#)
- item\_type\_converter\_t
  - algorithms\_libs, [96](#)
- item\_type\_helpers.h, [1769](#), [1770](#)
- item\_type\_is\_complex
  - algorithms\_libs, [109](#)
- item\_type\_size
  - algorithms\_libs, [109](#)
- item\_type\_valid
  - algorithms\_libs, [109](#)
- itemTypeToSize
  - FileSourceBase, [552](#)
  - FourBitCpxFileSignalSource, [575](#)
  - NsrFileSignalSource, [1210](#)
  - SpirFileSignalSource, [1452](#)
  - TwoBitCpxFileSignalSource, [1505](#)
  - TwoBitPackedFileSignalSource, [1510](#)
- iterator\_
  - IONGSMSSChunkUnpackingCtx< WT >, [1119](#)
- itn
  - DSM\_PKR\_message, [518](#)
- itrf
  - sta\_t, [1465](#)
- jerk
  - lexeph\_t, [1149](#)
- kernel
  - kernel\_info\_t, [1132](#)
- kernel\_dir\_t
  - fft\_internal.h, [1779](#)
- kernel\_info
  - cl\_fft\_plan, [442](#)
- kernel\_info\_t, [1132](#)
  - dir, [1132](#)
  - in\_place\_possible, [1132](#)
  - kernel, [1132](#)
  - kernel\_name, [1132](#)
  - lmem\_size, [1133](#)
  - next, [1133](#)
  - num\_workgroups, [1133](#)
  - num\_workitems\_per\_workgroup, [1133](#)
  - num\_xforms\_per\_workgroup, [1133](#)
- kernel\_name

- kernel\_info\_t, [1132](#)
- kernel\_string
  - cl\_fft\_plan, [442](#)
- key
  - MACK\_message, [1160](#)
- Kf\_Conf, [1134](#)
  - bit\_synchronization\_time\_limit\_s, [1135](#)
  - carrier\_disc\_sd\_rads, [1135](#)
  - carrier\_freq\_rate\_sd\_hz\_s, [1135](#)
  - carrier\_freq\_sd\_hz, [1136](#)
  - carrier\_lock\_test\_smoother\_alpha, [1136](#)
  - carrier\_lock\_test\_smoother\_samples, [1136](#)
  - carrier\_lock\_th, [1136](#)
  - carrier\_phase\_sd\_rad, [1136](#)
  - cn0\_min, [1136](#)
  - cn0\_samples, [1136](#)
  - cn0\_smoother\_alpha, [1136](#)
  - cn0\_smoother\_samples, [1136](#)
  - code\_disc\_sd\_chips, [1136](#)
  - code\_phase\_sd\_chips, [1136](#)
  - dump, [1136](#)
  - dump\_filename, [1137](#)
  - dump\_mat, [1137](#)
  - early\_late\_space\_chips, [1137](#)
  - early\_late\_space\_narrow\_chips, [1137](#)
  - enable\_doppler\_correction, [1137](#)
  - extend\_correlation\_symbols, [1137](#)
  - fs\_in, [1137](#)
  - high\_dyn, [1137](#)
  - init\_carrier\_freq\_rate\_sd\_hz\_s, [1137](#)
  - init\_carrier\_freq\_sd\_hz, [1137](#)
  - init\_carrier\_phase\_sd\_rad, [1137](#)
  - init\_code\_phase\_sd\_chips, [1137](#)
  - item\_type, [1138](#)
  - max\_carrier\_lock\_fail, [1138](#)
  - max\_code\_lock\_fail, [1138](#)
  - pull\_in\_time\_s, [1138](#)
  - signal, [1138](#)
  - slope, [1138](#)
  - smoother\_length, [1138](#)
  - spc, [1138](#)
  - system, [1138](#)
  - track\_pilot, [1138](#)
  - vector\_length, [1138](#)
  - very\_early\_late\_space\_chips, [1138](#)
  - very\_early\_late\_space\_narrow\_chips, [1139](#)
  - y\_intercept, [1139](#)
- kf\_conf.h, [2293](#)
- kf\_tracking, [1139](#)
- kf\_tracking.h, [2269](#)
- kml\_output\_enabled
  - Pvt\_Conf, [1310](#)
- kml\_output\_path
  - Pvt\_Conf, [1310](#)
- Kml\_Printer, [1140](#)
- kml\_printer.h, [1899](#), [1900](#)
- kml\_rate\_ms
  - Pvt\_Conf, [1310](#)
- kroot
  - DSM\_KROOT\_message, [514](#)
- ks
  - DSM\_KROOT\_message, [514](#)
- L
  - obsd\_t, [1218](#)
- L2\_P\_data\_flag
  - Gps\_Ephemeris, [943](#)
- l2c\_phasing\_flag
  - Gps\_CNAV\_Ephemeris, [929](#)
- labsat23\_source, [1141](#)
- labsat23\_source.h, [2050](#), [2051](#)
- labsat23\_source\_sptr
  - signal\_source\_gr\_blocks, [181](#)
- labsat\_signal\_source.h, [2011](#), [2012](#)
- LabsatSignalSource, [1143](#)
  - connect, [1145](#)
  - disconnect, [1145](#)
  - get\_left\_block, [1145](#)
  - get\_right\_block, [1145](#)
  - getRfChannels, [1146](#)
  - item\_size, [1146](#)
- lam
  - nav\_t, [1193](#)
- LAM\_CARR
  - algorithms\_libs\_rtklib, [146](#)
- lambda0red
  - Galileo\_Reduced\_CED, [653](#)
- last\_batch\_size
  - cl\_fft\_plan, [442](#)
- lat
  - Agnss\_Ref\_Location, [350](#)
  - sbsigp\_t, [1398](#)
- latitude
  - Monitor\_Pvt, [1173](#)
  - sdr\_gnss\_packet\_t, [1408](#)
- lats
  - tec\_t, [1482](#)
- LC
  - ambc\_t, [356](#)
- lcorr
  - sbssatp\_t, [1406](#)
- LCv
  - ambc\_t, [357](#)
- leaps
  - nav\_t, [1193](#)
- len
  - raw\_t, [1325](#)
  - rtcm\_t, [1354](#)
- length
  - fpga\_dma-proxy.h, [2081](#)
- lex\_t, [1146](#)
  - msgs, [1147](#)
  - n, [1147](#)
  - nmax, [1147](#)
- lexeph
  - nav\_t, [1193](#)
- lexeph\_t, [1148](#)

- acc, [1149](#)
- af0, [1149](#)
- af1, [1149](#)
- health, [1149](#)
- isc, [1149](#)
- jerk, [1149](#)
- pos, [1149](#)
- sat, [1149](#)
- tgdt, [1149](#)
- toe, [1149](#)
- tof, [1149](#)
- ura, [1149](#)
- vel, [1149](#)
- lexion
  - nav\_t, [1193](#)
- lexion\_t, [1150](#)
  - coef, [1150](#)
  - pos0, [1150](#)
  - t0, [1150](#)
  - tspan, [1151](#)
- lexmsg
  - raw\_t, [1325](#)
- lexmsg\_t, [1151](#)
  - alert, [1151](#)
  - msg, [1151](#)
  - prn, [1152](#)
  - snr, [1152](#)
  - stat, [1152](#)
  - ttr, [1152](#)
  - type, [1152](#)
- limesdr\_signal\_source.h, [2013](#)
- LimesdrSignalSource, [1152](#)
  - connect, [1155](#)
  - disconnect, [1155](#)
  - get\_left\_block, [1155](#)
  - get\_right\_block, [1155](#)
  - item\_size, [1155](#)
- LLI
  - obsd\_t, [1218](#)
- lltime
  - rtcm\_t, [1354](#)
- lmem\_size
  - kernel\_info\_t, [1133](#)
- lo\_hz
  - stream\_cfg, [1467](#)
- load\_cnav\_ephemeris\_xml
  - Gnss\_Sdr\_Supl\_Client, [887](#)
- load\_cnav\_utc\_xml
  - Gnss\_Sdr\_Supl\_Client, [887](#)
- load\_ephemeris\_xml
  - Gnss\_Sdr\_Supl\_Client, [887](#)
- load\_gal\_almanac\_xml
  - Gnss\_Sdr\_Supl\_Client, [887](#)
- load\_gal\_ephemeris\_xml
  - Gnss\_Sdr\_Supl\_Client, [887](#)
- load\_gal\_iono\_xml
  - Gnss\_Sdr\_Supl\_Client, [887](#)
- load\_gal\_utc\_xml
  - Gnss\_Sdr\_Supl\_Client, [888](#)
- load\_glo\_utc\_xml
  - Gnss\_Sdr\_Supl\_Client, [888](#)
- load\_gnav\_ephemeris\_xml
  - Gnss\_Sdr\_Supl\_Client, [888](#)
- load\_gps\_almanac\_xml
  - Gnss\_Sdr\_Supl\_Client, [888](#)
- load\_iono\_xml
  - Gnss\_Sdr\_Supl\_Client, [888](#)
- load\_ref\_location\_xml
  - Gnss\_Sdr\_Supl\_Client, [888](#)
- load\_ref\_time\_xml
  - Gnss\_Sdr\_Supl\_Client, [888](#)
- load\_utc\_xml
  - Gnss\_Sdr\_Supl\_Client, [888](#)
- local
  - ftp\_t, [595](#)
- lock
  - file\_t, [538](#)
  - rtcm\_t, [1354](#)
  - rtksvr\_t, [1377](#)
  - solstat\_t, [1446](#)
  - ssat\_t, [1459](#)
  - stream\_t, [1469](#)
- lock\_channel
  - Fpga\_Multicorrelator\_8sc, [586](#)
- lock\_detectors.h, [2294](#), [2295](#)
- lock\_t
  - algorithms\_libs\_rtklib, [132](#)
- lock\_time
  - Rtcm, [1341](#), [1342](#)
  - Rtcm\_Printer, [1351](#)
- lockt
  - raw\_t, [1325](#)
- lod
  - erpd\_t, [525](#)
- LOG\_PI
  - rtklib\_ppp.h, [1824](#)
- log\_rinex\_nav\_bds\_dnav
  - Rinex\_Printer, [1334](#)
- log\_rinex\_nav\_gal\_nav
  - Rinex\_Printer, [1334](#)
- log\_rinex\_nav\_glo\_gnav
  - Rinex\_Printer, [1335](#)
- log\_rinex\_nav\_gps\_cnav
  - Rinex\_Printer, [1335](#)
- log\_rinex\_nav\_gps\_nav
  - Rinex\_Printer, [1335](#)
- log\_source\_timetag
  - Pvt\_Conf, [1311](#)
- log\_source\_timetag\_file
  - Pvt\_Conf, [1311](#)
- lon
  - AgNSS\_Ref\_Location, [350](#)
  - sbsignt, [1398](#)
- longitude
  - Monitor\_Pvt, [1174](#)
  - sdr\_gnss\_packet\_t, [1408](#)



- lons
  - tec\_t, [1483](#)
- LOOPMAX
  - rtklib\_lambda.h, [1816](#)
- loss
  - rtcm\_t, [1355](#)
- M
  - tled\_t, [1488](#)
- M0
  - alm\_t, [354](#)
  - eph\_t, [522](#)
- M0\_10
  - Galileo\_Almanac\_Helper, [604](#)
- M0\_10\_LSB
  - core\_system\_parameters, [307](#)
- M0\_1\_LSB
  - core\_system\_parameters, [307](#)
- M0\_7
  - Galileo\_Almanac\_Helper, [604](#)
- M0\_7\_LSB
  - core\_system\_parameters, [307](#)
- M0\_9
  - Galileo\_Almanac\_Helper, [604](#)
- M0\_9\_LSB
  - core\_system\_parameters, [307](#)
- M\_0
  - Gnss\_Almanac, [863](#)
  - Gnss\_Ephemeris, [876](#)
- M\_0\_LSB
  - core\_system\_parameters, [308](#)
- mack
  - OSNMA\_msg, [1231](#)
- MACK\_header, [1156](#)
  - cop, [1156](#)
  - macseq, [1156](#)
  - tag0, [1156](#)
- Mack\_lookup, [1157](#)
  - Mack\_lookup, [1158](#)
  - msg, [1158](#)
  - nt, [1158](#)
  - sequence1, [1158](#)
  - sequence2, [1158](#)
- MACK\_message, [1159](#)
  - header, [1160](#)
  - key, [1160](#)
  - PRNa, [1160](#)
  - tag\_and\_info, [1160](#)
  - TOW, [1160](#)
  - WN, [1160](#)
- MACK\_tag\_and\_info, [1161](#)
  - counter, [1161](#)
  - tag, [1161](#)
  - tag\_info, [1161](#)
- MACK\_tag\_info, [1162](#)
  - ADKD, [1162](#)
  - cop, [1162](#)
  - PRN\_d, [1162](#)
- maclt
  - DSM\_KROOT\_message, [514](#)
- macseq
  - MACK\_header, [1156](#)
- mag
  - AcquisitionInterface, [334](#)
  - BeidouB1iPcpsAcquisition, [401](#)
  - BeidouB3iPcpsAcquisition, [415](#)
  - galileo\_e5a\_noncoherentIQ\_acquisition\_caf\_cc, [611](#)
  - galileo\_pcps\_8ms\_acquisition\_cc, [648](#)
  - GalileoE1Pcps8msAmbiguousAcquisition, [677](#)
  - GalileoE1PcpsAmbiguousAcquisition, [683](#)
  - GalileoE1PcpsAmbiguousAcquisitionFpga, [690](#)
  - GalileoE1PcpsCccwsrAmbiguousAcquisition, [695](#)
  - GalileoE1PcpsQuickSyncAmbiguousAcquisition, [701](#)
  - GalileoE1PcpsTongAmbiguousAcquisition, [707](#)
  - GalileoE5aNoncoherentIQAcquisitionCaf, [726](#)
  - GalileoE5aPcpsAcquisition, [732](#)
  - GalileoE5aPcpsAcquisitionFpga, [739](#)
  - GalileoE5bPcpsAcquisition, [754](#)
  - GalileoE5bPcpsAcquisitionFpga, [761](#)
  - GalileoE6PcpsAcquisition, [775](#)
  - GlionassL1CaPcpsAcquisition, [837](#)
  - GlionassL2CaPcpsAcquisition, [854](#)
  - GpsL1CaPcpsAcquisition, [989](#)
  - GpsL1CaPcpsAcquisitionFineDoppler, [995](#)
  - GpsL1CaPcpsAcquisitionFpga, [1001](#)
  - GpsL1CaPcpsAssistedAcquisition, [1007](#)
  - GpsL1CaPcpsOpenCIAcquisition, [1012](#)
  - GpsL1CaPcpsQuickSyncAcquisition, [1018](#)
  - GpsL1CaPcpsTongAcquisition, [1023](#)
  - GpsL2MPcpsAcquisition, [1048](#)
  - GpsL2MPcpsAcquisitionFpga, [1054](#)
  - GpsL5iPcpsAcquisition, [1068](#)
  - GpsL5iPcpsAcquisitionFpga, [1075](#)
  - pcps\_acquisition, [1244](#)
  - pcps\_acquisition\_fine\_doppler\_cc, [1249](#)
  - pcps\_acquisition\_fpga, [1254](#)
  - pcps\_assisted\_acquisition\_cc, [1259](#)
  - pcps\_cccwsr\_acquisition\_cc, [1264](#)
  - pcps\_opencl\_acquisition\_cc, [1269](#)
  - pcps\_quicksync\_acquisition\_cc, [1275](#)
  - pcps\_tong\_acquisition\_cc, [1281](#)
- magnitude2
  - GPU\_Complex, [1081](#)
  - GPU\_Complex\_Short, [1083](#)
- make\_2\_steps
  - Acq\_Conf, [326](#)
  - Acq\_Conf\_Fpga, [331](#)
- make\_vector\_converter
  - algorithms\_libs, [109](#)
- marker
  - sta\_t, [1465](#)
- mask
  - snrmask\_t, [1437](#)
- mask\_flag
  - mt1\_header, [1180](#)



mask\_id  
    mt1\_header, 1180  
MATH\_CONSTANTS.h, 2709, 2712  
MAX  
    telemetry\_decoder\_libswiftcnv, 191  
max2771\_evkit\_signal\_source\_fpga.h, 2014, 2015  
MAX2771EVKITSignalSourceFPGA, 1163  
    connect, 1165  
    disconnect, 1165  
    get\_left\_block, 1165  
    get\_right\_block, 1165  
    item\_size, 1165  
max\_carrier\_lock\_fail  
    Dll\_Pll\_Conf, 496  
    Dll\_Pll\_Conf\_Fpga, 502  
    Kf\_Conf, 1138  
max\_code\_lock\_fail  
    Dll\_Pll\_Conf, 496  
    Dll\_Pll\_Conf\_Fpga, 502  
    Kf\_Conf, 1138  
max\_dwells  
    Acq\_Conf, 327  
max\_localmem\_fft\_size  
    cl\_fft\_plan, 442  
max\_num\_acqs  
    Acq\_Conf\_Fpga, 331  
max\_obs\_block\_rx\_clock\_offset\_ms  
    Pvt\_Conf, 1311  
max\_radix  
    cl\_fft\_plan, 442  
MAX\_SECONDS\_REMEMBERING\_MID  
    core\_system\_parameters, 308  
MAX\_TOA\_DELAY\_MS  
    core\_system\_parameters, 308  
max\_work\_item\_per\_workgroup  
    cl\_fft\_plan, 442  
MAXAC  
    algorithms\_libs\_rtklib, 147  
MAXANT  
    algorithms\_libs\_rtklib, 147  
maxaveep  
    prcopt\_t, 1299  
MAXBAND  
    algorithms\_libs\_rtklib, 147  
MAXCLI  
    algorithms\_libs\_rtklib, 147  
    rtklib\_stream.h, 1866  
MAXCODE  
    core\_system\_parameters, 308  
MAXDTE  
    rtklib\_preceph.h, 1828  
MAXDTOE  
    algorithms\_libs\_rtklib, 147  
MAXDTOE\_BDS  
    algorithms\_libs\_rtklib, 147  
MAXDTOE\_GAL  
    algorithms\_libs\_rtklib, 148  
MAXDTOE\_GLO  
    algorithms\_libs\_rtklib, 148  
MAXDTOE\_QZS  
    algorithms\_libs\_rtklib, 148  
MAXDTOE\_S  
    algorithms\_libs\_rtklib, 148  
MAXDTOE\_SBS  
    algorithms\_libs\_rtklib, 148  
MAXERRMSG  
    algorithms\_libs\_rtklib, 148  
MAXEXFILE  
    algorithms\_libs\_rtklib, 149  
MAXFREQ  
    algorithms\_libs\_rtklib, 149  
MAXGDOP  
    algorithms\_libs\_rtklib, 149  
maxgdop  
    prcopt\_t, 1299  
maxinno  
    prcopt\_t, 1299  
MAXITR  
    rtklib\_pntpos.h, 1819  
MAXLEAPS  
    algorithms\_libs\_rtklib, 149  
MAXNGEO  
    algorithms\_libs\_rtklib, 149  
MAXNIGP  
    algorithms\_libs\_rtklib, 149  
MAXOBS  
    algorithms\_libs\_rtklib, 150  
MAXOBSBUF  
    algorithms\_libs\_rtklib, 150  
MAXOBSTYPE  
    algorithms\_libs\_rtklib, 150  
maxout  
    prcopt\_t, 1299  
MAXPRNBDS  
    algorithms\_libs\_rtklib, 150  
MAXPRNGAL  
    algorithms\_libs\_rtklib, 150  
MAXPRNGLO  
    algorithms\_libs\_rtklib, 150  
MAXPRNGPS  
    algorithms\_libs\_rtklib, 151  
MAXPRNIRN  
    algorithms\_libs\_rtklib, 151  
MAXPRNLEO  
    algorithms\_libs\_rtklib, 151  
MAXPRNQZS  
    algorithms\_libs\_rtklib, 151  
MAXPRNSBS  
    algorithms\_libs\_rtklib, 151  
MAXRAWLEN  
    algorithms\_libs\_rtklib, 151  
MAXRCV  
    algorithms\_libs\_rtklib, 152  
MAXSAT  
    algorithms\_libs\_rtklib, 152  
MAXSBSAGEF

- algorithms\_libs\_rtklib, [152](#)
- MAXSBSAGEL
  - algorithms\_libs\_rtklib, [152](#)
- MAXSBSMSG
  - algorithms\_libs\_rtklib, [152](#)
- MAXSBSURA
  - algorithms\_libs\_rtklib, [152](#)
- MAXSOLBUF
  - algorithms\_libs\_rtklib, [153](#)
- MAXSOLMSG
  - algorithms\_libs\_rtklib, [153](#)
- MAXSOLQ
  - algorithms\_libs\_rtklib, [153](#)
- maxsolstd
  - solopt\_t, [1443](#)
- MAXSTA
  - algorithms\_libs\_rtklib, [153](#)
- MAXSTATMSG
  - algorithms\_libs\_rtklib, [153](#)
  - rtklib\_stream.h, [1866](#)
- MAXSTRMSG
  - algorithms\_libs\_rtklib, [153](#)
- MAXSTRPATH
  - algorithms\_libs\_rtklib, [154](#)
- MAXSTRRTK
  - algorithms\_libs\_rtklib, [154](#)
- maxtdiff
  - prcopt\_t, [1299](#)
- me
  - ControlThread, [467](#)
- measures\_ecef\_pos\_sd\_m
  - Pvt\_Conf, [1311](#)
- measures\_ecef\_vel\_sd\_ms
  - Pvt\_Conf, [1311](#)
- message\_id
  - Galileo\_HAS\_data, [634](#)
  - Galileo\_HAS\_page, [637](#)
- message\_lock
  - cnav\_v27\_part\_t, [451](#)
- message\_page\_id
  - Galileo\_HAS\_page, [637](#)
- message\_size
  - Galileo\_HAS\_page, [637](#)
- message\_type
  - Galileo\_HAS\_page, [637](#)
- metrics1
  - v27\_t, [1530](#)
- metrics2
  - v27\_t, [1530](#)
- mf
  - DSM\_KROOT\_message, [514](#)
- microseconds
  - Agnss\_Ref\_Time, [351](#)
  - sdr\_gnss\_packet\_t, [1408](#)
- mid
  - DSM\_PKR\_message, [518](#)
- MIN
  - telemetry\_decoder\_libswiftcnav, [191](#)
- MIN\_ARC\_GAP
  - rtklib\_ppp.h, [1824](#)
- MIN\_EL
  - rtklib\_ionex.h, [1814](#)
- MIN\_HGT
  - rtklib\_ionex.h, [1814](#)
- min\_mem\_coalesce\_width
  - cl\_fft\_plan, [442](#)
- MIN\_PPP
  - rtklib\_ppp.h, [1823](#)
- minfix
  - prcopt\_t, [1300](#)
- minlock
  - prcopt\_t, [1300](#)
- MINPRNBDS
  - algorithms\_libs\_rtklib, [154](#)
- MINPRNGAL
  - algorithms\_libs\_rtklib, [154](#)
- MINPRNGLO
  - algorithms\_libs\_rtklib, [154](#)
- MINPRNGPS
  - algorithms\_libs\_rtklib, [154](#)
- MINPRNI RN
  - algorithms\_libs\_rtklib, [155](#)
- MINPRNLEO
  - algorithms\_libs\_rtklib, [155](#)
- MINPRNQZS
  - algorithms\_libs\_rtklib, [155](#)
- MINPRNQZS\_S
  - algorithms\_libs\_rtklib, [155](#)
- MINPRNSBS
  - algorithms\_libs\_rtklib, [155](#)
- mjd
  - erpd\_t, [525](#)
- mmse\_resampler\_conditioner.h, [1971](#)
- MmseResamplerConditioner, [1166](#)
  - connect, [1168](#)
  - disconnect, [1168](#)
  - get\_left\_block, [1168](#)
  - get\_right\_block, [1168](#)
  - implementation, [1168](#)
  - item\_size, [1168](#)
  - role, [1168](#)
- mntpnt
  - ntrip\_t, [1213](#)
- mode
  - file\_t, [538](#)
  - prcopt\_t, [1300](#)
  - stream\_t, [1469](#)
- modear
  - prcopt\_t, [1300](#)
- ModelFunction, [1169](#)
  - ModelFunction, [1169](#)
- moni
  - rtksvr\_t, [1377](#)
- monitor\_enabled
  - Pvt\_Conf, [1311](#)
- monitor\_ephemeris\_enabled

- Pvt\_Conf, [1311](#)
- Monitor\_Ephemeris\_Udp\_Sink, [1169](#)
- monitor\_ephemeris\_udp\_sink.h, [1900](#), [1901](#)
- Monitor\_Pvt, [1170](#)
  - AR\_ratio\_factor, [1172](#)
  - AR\_ratio\_threshold, [1172](#)
  - cog, [1172](#)
  - cov\_xx, [1173](#)
  - cov\_xy, [1173](#)
  - cov\_yy, [1173](#)
  - cov\_yz, [1173](#)
  - cov\_zx, [1173](#)
  - cov\_zz, [1173](#)
  - galhas\_status, [1173](#)
  - gdop, [1173](#)
  - geohash, [1173](#)
  - hdop, [1173](#)
  - height, [1173](#)
  - latitude, [1173](#)
  - longitude, [1174](#)
  - pdop, [1174](#)
  - pos\_x, [1174](#)
  - pos\_y, [1174](#)
  - pos\_z, [1174](#)
  - RX\_time, [1174](#)
  - serialize, [1172](#)
  - solution\_status, [1174](#)
  - solution\_type, [1174](#)
  - TOW\_at\_current\_symbol\_ms, [1174](#)
  - user\_clk\_drift\_ppm, [1174](#)
  - user\_clk\_offset, [1174](#)
  - utc\_time, [1174](#)
  - valid\_sats, [1175](#)
  - vdop, [1175](#)
  - vel\_e, [1175](#)
  - vel\_n, [1175](#)
  - vel\_u, [1175](#)
  - vel\_x, [1175](#)
  - vel\_y, [1175](#)
  - vel\_z, [1175](#)
  - week, [1175](#)
- monitor\_pvt.h, [1902](#), [1903](#)
- Monitor\_Pvt\_Udp\_Sink, [1176](#)
- monitor\_pvt\_udp\_sink.h, [1905](#), [1906](#)
- MS
  - ion\_gsms\_stream\_encodings.h, [2102](#)
- ms\_per\_code
  - Acq\_Conf, [327](#)
- MSA
  - ion\_gsms\_stream\_encodings.h, [2102](#)
- msg
  - lexmsg\_t, [1151](#)
  - Mack\_lookup, [1158](#)
  - rtcm\_t, [1355](#)
  - sbsmsg\_t, [1403](#)
  - stream\_t, [1469](#)
- MSG\_DISCONN
  - algorithms\_libs\_rtklib, [133](#)
- msg\_handler\_osnma
  - osnma\_msg\_receiver, [1233](#)
- msg\_id
  - cnav\_msg\_t, [449](#)
- msgs
  - lex\_t, [1147](#)
  - sbs\_t, [1395](#)
- msgtype
  - raw\_t, [1325](#)
  - rtcm\_t, [1355](#)
- msm\_h\_t, [1177](#)
  - cellmask, [1177](#)
  - clk\_ext, [1177](#)
  - clk\_str, [1177](#)
  - iod, [1178](#)
  - nsat, [1178](#)
  - nsig, [1178](#)
  - sats, [1178](#)
  - sigs, [1178](#)
  - smooth, [1178](#)
  - time\_s, [1178](#)
  - tint\_s, [1178](#)
- msmtype
  - rtcm\_t, [1355](#)
- mt1\_header, [1179](#)
  - clock\_fullset\_flag, [1179](#)
  - clock\_subset\_flag, [1179](#)
  - code\_bias\_flag, [1179](#)
  - iod\_set\_id, [1180](#)
  - mask\_flag, [1180](#)
  - mask\_id, [1180](#)
  - orbit\_correction\_flag, [1180](#)
  - phase\_bias\_flag, [1180](#)
  - reserved, [1180](#)
  - toh, [1180](#)
- multichannel\_file\_signal\_source.h, [2017](#), [2018](#)
- MultichannelFileSignalSource, [1180](#)
  - connect, [1183](#)
  - disconnect, [1183](#)
  - filename, [1183](#)
  - get\_left\_block, [1183](#)
  - get\_right\_block, [1183](#)
  - item\_size, [1184](#)
  - item\_type, [1184](#)
  - repeat, [1184](#)
  - samples, [1184](#)
  - sampling\_frequency, [1184](#)
- multiply\_acc
  - GPU\_Complex, [1081](#)
- mw
  - ssat\_t, [1459](#)
- n
  - ambc\_t, [357](#)
  - cl\_fft\_plan, [443](#)
  - erp\_t, [524](#)
  - lex\_t, [1147](#)
  - nav\_t, [1193](#)
  - obs\_t, [1217](#)

- pcvs\_t, 1286
- sbs\_t, 1395
- solbuf\_t, 1441
- solstatbuf\_t, 1448
- tle\_t, 1486
- tled\_t, 1488
- n\_bytes
  - ad936x\_iio\_samples, 339
- n\_channels
  - ad936x\_iio\_custom, 338
  - ad936x\_iio\_samples, 339
- n\_crc\_fail
  - cnav\_v27\_part\_t, 451
- n\_decoded
  - cnav\_v27\_part\_t, 452
- n\_interleaved\_iq\_samples
  - ad936x\_iio\_samples, 339
- n\_symbols
  - cnav\_v27\_part\_t, 452
- na
  - nav\_t, 1193
  - rtk\_t, 1358
- namax
  - nav\_t, 1193
- name
  - opt\_t, 1222
  - sta\_t, 1465
  - tled\_t, 1488
- nav
  - raw\_t, 1325
  - rtcm\_t, 1355
  - rtksvr\_t, 1377
- nav\_data
  - Tag, 1474
- nav\_data\_id
  - OSNMA\_NavData, 1236
- nav\_message
  - Galileo\_HAS\_data, 634
  - Nav\_Message\_Packet, 1189
- nav\_message\_monitor, 1184
  - ~nav\_message\_monitor, 1185
- nav\_message\_monitor.h, 2353, 2354
- nav\_message\_monitor\_sptr
  - core\_libs, 209
- Nav\_Message\_Packet, 1186
  - ~Nav\_Message\_Packet, 1187
  - nav\_message, 1189
  - Nav\_Message\_Packet, 1187
  - operator=, 1188
  - prn, 1189
  - signal, 1189
  - system, 1189
  - tow\_at\_current\_symbol\_ms, 1189
- nav\_message\_packet.h, 2354, 2355
- Nav\_Message\_Udp\_Sink, 1189
- nav\_message\_udp\_sink.h, 2356, 2357
- nav\_t, 1190
  - alm, 1192
  - cbias, 1192
  - dgps, 1192
  - eph, 1192
  - erp, 1192
  - fcb, 1192
  - geph, 1192
  - glo\_cpbias, 1192
  - glo\_fcn, 1192
  - ion\_cmp, 1192
  - ion\_gal, 1192
  - ion\_gps, 1192
  - ion\_irn, 1192
  - ion\_qzs, 1193
  - lam, 1193
  - leaps, 1193
  - lexeph, 1193
  - lexion, 1193
  - n, 1193
  - na, 1193
  - namax, 1193
  - nc, 1193
  - ncmax, 1193
  - ne, 1193
  - nemax, 1193
  - nf, 1194
  - nfmax, 1194
  - ng, 1194
  - ngmax, 1194
  - nmax, 1194
  - ns, 1194
  - nsmax, 1194
  - nt, 1194
  - ntmax, 1194
  - pclk, 1194
  - pcvs, 1194
  - peph, 1194
  - pppcorr, 1195
  - rbias, 1195
  - sbsion, 1195
  - sbssat, 1195
  - seph, 1195
  - ssr, 1195
  - tec, 1195
  - utc\_cmp, 1195
  - utc\_gal, 1195
  - utc\_glo, 1195
  - utc\_gps, 1195
  - utc\_irn, 1195
  - utc\_qzs, 1196
  - utc\_sbs, 1196
  - wlbias, 1196
- nav\_type
  - Beidou\_Dnav\_Ephemeris, 383
- navsel
  - rtksvr\_t, 1377
- navsys
  - prcopt\_t, 1300
- nb

- ntrip\_t, [1213](#)
  - rtksvr\_t, [1377](#)
  - solbuf\_t, [1441](#)
- nb\_dk
  - DSM\_KROOT\_message, [514](#)
- nb\_dp
  - DSM\_PKR\_message, [518](#)
- NB\_RTK
  - algorithms\_libs\_rtklib, [133](#)
- nbit
  - rtcm\_t, [1355](#)
- nbyte
  - raw\_t, [1325](#)
  - rtcm\_t, [1355](#)
- nc
  - nav\_t, [1193](#)
- nchannels\_in
  - Obs\_Conf, [1215](#)
- nchannels\_out
  - Obs\_Conf, [1215](#)
- ncmax
  - nav\_t, [1193](#)
- ndata
  - tec\_t, [1483](#)
- nddot
  - tled\_t, [1488](#)
- ndot
  - eph\_t, [522](#)
  - tled\_t, [1489](#)
- ne
  - nav\_t, [1193](#)
- neb
  - rtk\_t, [1358](#)
- nemax
  - nav\_t, [1193](#)
- new\_metrics
  - v27\_t, [1530](#)
- NEXOBS
  - algorithms\_libs\_rtklib, [155](#)
- next
  - half\_cyc\_tag, [1087](#)
  - kernel\_info\_t, [1133](#)
- nextPowerOf2
  - pcps\_acquisition\_fine\_doppler\_cc, [1249](#)
- nf
  - nav\_t, [1194](#)
  - prcopt\_t, [1300](#)
- NF\_RTK
  - algorithms\_libs\_rtklib, [133](#)
- nfix
  - rtk\_t, [1358](#)
- nfmax
  - nav\_t, [1194](#)
- NFREQ
  - algorithms\_libs\_rtklib, [156](#)
- NFREQGLO
  - algorithms\_libs\_rtklib, [156](#)
- ng
  - nav\_t, [1194](#)
- ngmax
  - nav\_t, [1194](#)
- NI\_RTK
  - algorithms\_libs\_rtklib, [133](#)
- nigp
  - sbsion\_t, [1401](#)
- niter
  - prcopt\_t, [1300](#)
- NL\_RTK
  - algorithms\_libs\_rtklib, [133](#)
- nmas
  - DSM\_nma\_header, [516](#)
- NMAX
  - rtklib\_preceph.h, [1828](#)
- nmax
  - erp\_t, [524](#)
  - lex\_t, [1147](#)
  - nav\_t, [1194](#)
  - obs\_t, [1217](#)
  - pcvs\_t, [1286](#)
  - sbs\_t, [1395](#)
  - solbuf\_t, [1441](#)
  - solstatbuf\_t, [1448](#)
  - tle\_t, [1486](#)
- nmea\_dump\_devname
  - Pvt\_Conf, [1311](#)
- nmea\_dump\_filename
  - Pvt\_Conf, [1311](#)
- nmea\_output\_file\_enabled
  - Pvt\_Conf, [1311](#)
- nmea\_output\_file\_path
  - Pvt\_Conf, [1311](#)
- Nmea\_Printer, [1196](#)
  - ~Nmea\_Printer, [1197](#)
  - Nmea\_Printer, [1197](#)
  - Print\_Nmea\_Line, [1197](#)
- nmea\_printer.h, [1907](#), [1908](#)
- nmea\_rate\_ms
  - Pvt\_Conf, [1311](#)
- nmeacycle
  - rtksvr\_t, [1377](#)
- nmeaintv
  - solo\_t, [1443](#)
- nmeapos
  - rtksvr\_t, [1377](#)
- nmeareq
  - rtksvr\_t, [1377](#)
- nmsg
  - rtksvr\_t, [1377](#)
- nmsg2
  - rtcm\_t, [1355](#)
- nmsg3
  - rtcm\_t, [1355](#)
- nonlinear\_tracking.h, [2296](#), [2297](#)
- Notch, [1197](#)
- notch\_cc.h, [1712](#), [1713](#)
- notch\_filter.h, [1706](#), [1707](#)

- notch\_filter\_lite.h, 1708
- notch\_lite\_cc.h, 1715, 1716
- notch\_lite\_sptr
  - input\_filter\_gr\_blocks, 89
- notch\_sptr
  - input\_filter\_gr\_blocks, 89
- NotchFilter, 1199
  - connect, 1201
  - disconnect, 1201
  - get\_left\_block, 1201
  - get\_right\_block, 1201
  - implementation, 1201
  - item\_size, 1201
  - role, 1201
- NotchFilterLite, 1202
  - connect, 1204
  - disconnect, 1204
  - get\_left\_block, 1204
  - get\_right\_block, 1204
  - implementation, 1204
  - item\_size, 1204
  - role, 1204
- NotchLite, 1205
- NP\_PPP
  - algorithms\_libs\_rtklib, 133
- NP\_RTK
  - algorithms\_libs\_rtklib, 134
- npb
  - rtksvr\_t, 1377
- npk
  - DSM\_PKR\_message, 518
- npkt
  - DSM\_PKR\_message, 518
- npktid
  - DSM\_PKR\_message, 518
- NR\_PPP
  - algorithms\_libs\_rtklib, 134
- NR\_RTK
  - algorithms\_libs\_rtklib, 134
- ns
  - nav\_t, 1194
  - pppcorr\_t, 1293
  - sol\_t, 1439
- nsat
  - msm\_h\_t, 1178
  - sbssat\_t, 1405
- NSATBDS
  - algorithms\_libs\_rtklib, 156
- NSATGAL
  - algorithms\_libs\_rtklib, 156
- NSATGLO
  - algorithms\_libs\_rtklib, 156
- NSATGPS
  - algorithms\_libs\_rtklib, 156
- NSATIRN
  - algorithms\_libs\_rtklib, 157
- NSATLEO
  - algorithms\_libs\_rtklib, 157
- NSATQZS
  - algorithms\_libs\_rtklib, 157
- NSATSBS
  - algorithms\_libs\_rtklib, 157
- nsb
  - rtksvr\_t, 1377
- nsbs
  - rtksvr\_t, 1377
- nsig
  - msm\_h\_t, 1178
- nsmax
  - nav\_t, 1194
  - pppcorr\_t, 1293
- nsol
  - rtksvr\_t, 1378
- nsr\_file\_signal\_source.h, 2019, 2020
- NsrFileSignalSource, 1206
  - create\_file\_source\_hook, 1210
  - itemTypeToSize, 1210
  - packetsPerSample, 1210
  - pre\_connect\_hook, 1210
  - pre\_disconnect\_hook, 1210
  - source, 1211
- nsta
  - pppcorr\_t, 1293
- nsvfix
  - sdr\_gnss\_packet\_t, 1408
- NSYS
  - algorithms\_libs\_rtklib, 157
- Nsys
  - Galileo\_HAS\_data, 634
- Nsys\_sub
  - Galileo\_HAS\_data, 634
- NSYSBDS
  - algorithms\_libs\_rtklib, 157
- NSYSGAL
  - algorithms\_libs\_rtklib, 157
- NSYSGLO
  - algorithms\_libs\_rtklib, 158
- NSYSGPS
  - algorithms\_libs\_rtklib, 158
- NSYSIRN
  - algorithms\_libs\_rtklib, 158
- NSYSLEO
  - algorithms\_libs\_rtklib, 158
- NSYSQZS
  - algorithms\_libs\_rtklib, 158
- nt
  - Mack\_lookup, 1158
  - nav\_t, 1194
  - pppcorr\_t, 1293
- NT\_RTK
  - algorithms\_libs\_rtklib, 134
- ntmax
  - nav\_t, 1194
  - pppcorr\_t, 1293
- NTRIP\_AGENT
  - algorithms\_libs\_rtklib, 134

- rtklib\_stream.h, [1866](#)
- NTRIP\_CLI\_PORT
  - algorithms\_libs\_rtklib, [135](#)
  - rtklib\_stream.h, [1866](#)
- NTRIP\_MAXRSP
  - algorithms\_libs\_rtklib, [135](#)
  - rtklib\_stream.h, [1866](#)
- NTRIP\_MAXSTR
  - algorithms\_libs\_rtklib, [135](#)
  - rtklib\_stream.h, [1866](#)
- NTRIP\_RSP\_ERROR
  - algorithms\_libs\_rtklib, [135](#)
  - rtklib\_stream.h, [1866](#)
- NTRIP\_RSP\_HTTP
  - algorithms\_libs\_rtklib, [135](#)
  - rtklib\_stream.h, [1867](#)
- NTRIP\_RSP\_OK\_CLI
  - algorithms\_libs\_rtklib, [135](#)
  - rtklib\_stream.h, [1867](#)
- NTRIP\_RSP\_OK\_SVR
  - algorithms\_libs\_rtklib, [135](#)
  - rtklib\_stream.h, [1867](#)
- NTRIP\_RSP\_SRCTBL
  - algorithms\_libs\_rtklib, [135](#)
  - rtklib\_stream.h, [1867](#)
- NTRIP\_RSP\_TBLEND
  - algorithms\_libs\_rtklib, [136](#)
  - rtklib\_stream.h, [1867](#)
- NTRIP\_SVR\_PORT
  - algorithms\_libs\_rtklib, [136](#)
  - rtklib\_stream.h, [1867](#)
- ntrip\_t, [1212](#)
  - buff, [1213](#)
  - mntpnt, [1213](#)
  - nb, [1213](#)
  - passwd, [1213](#)
  - state, [1213](#)
  - str, [1213](#)
  - tcp, [1213](#)
  - type, [1213](#)
  - url, [1213](#)
  - user, [1213](#)
- num\_doppler\_bins\_step2
  - Acq\_Conf, [327](#)
  - Acq\_Conf\_Fpga, [331](#)
- num\_kernels
  - cl\_fft\_plan, [443](#)
- num\_local\_mem\_banks
  - cl\_fft\_plan, [443](#)
- NUM\_RX\_VARIABLES
  - tracking\_libs, [200](#)
- NUM\_TX\_VARIABLES\_GALILEO\_E1
  - tracking\_libs, [200](#)
- NUM\_TX\_VARIABLES\_GPS\_L1\_CA
  - tracking\_libs, [200](#)
- num\_workgroups
  - kernel\_info\_t, [1133](#)
- num\_workitems\_per\_workgroup
  - kernel\_info\_t, [1133](#)
- num\_xforms\_per\_workgroup
  - kernel\_info\_t, [1133](#)
- NX
  - rtklib\_pntpos.h, [1820](#)
- nx
  - rtk\_t, [1358](#)
- NX\_PPP
  - algorithms\_libs\_rtklib, [136](#)
- NX\_RTK
  - algorithms\_libs\_rtklib, [136](#)
- OB
  - ion\_gsms\_stream\_encodings.h, [2102](#)
- OBA
  - ion\_gsms\_stream\_encodings.h, [2102](#)
- obs
  - raw\_t, [1325](#)
  - rtcm\_t, [1355](#)
  - rtksvr\_t, [1378](#)
- obs\_adapters, [173](#)
- Obs\_Conf, [1214](#)
  - always\_output\_gs, [1215](#)
  - dump, [1215](#)
  - dump\_filename, [1215](#)
  - dump\_mat, [1215](#)
  - enable\_carrier\_smoothing, [1215](#)
  - enable\_E6, [1215](#)
  - nchannels\_in, [1215](#)
  - nchannels\_out, [1215](#)
  - observable\_interval\_ms, [1215](#)
  - smoothing\_factor, [1215](#)
- obs\_conf.h, [1880](#), [1881](#)
- obs\_gr\_blocks, [174](#)
  - hybrid\_observables\_gs\_sptr, [174](#)
- obs\_t, [1216](#)
  - data, [1217](#)
  - n, [1217](#)
  - nmax, [1217](#)
- obsd\_t, [1217](#)
  - code, [1218](#)
  - D, [1218](#)
  - L, [1218](#)
  - LLI, [1218](#)
  - P, [1218](#)
  - rcv, [1218](#)
  - sat, [1218](#)
  - SNR, [1218](#)
  - time, [1218](#)
- observable\_interval\_ms
  - Obs\_Conf, [1215](#)
  - Pvt\_Conf, [1312](#)
- Observables, [173](#)
- observables\_interface.h, [2319](#), [2320](#)
- observables\_libs, [175](#)
- ObservablesInterface, [1219](#)
- obsflag
  - rtcm\_t, [1355](#)
- obuf

- raw\_t, [1325](#)
- odisp
  - prcopt\_t, [1300](#)
- off
  - pcv\_t, [1284](#)
  - raw\_t, [1326](#)
- offset
  - file\_t, [538](#)
- OG
  - ion\_gsms\_stream\_encodings.h, [2102](#)
- OGA
  - ion\_gsms\_stream\_encodings.h, [2102](#)
- old\_metrics
  - v27\_t, [1530](#)
- omega
  - Gnss\_Almanac, [863](#)
  - Gnss\_Ephemeris, [876](#)
- Omega0\_10
  - Galileo\_Almanac\_Helper, [604](#)
- OMEGA0\_10\_LSB
  - core\_system\_parameters, [308](#)
- Omega0\_7
  - Galileo\_Almanac\_Helper, [604](#)
- OMEGA0\_7\_LSB
  - core\_system\_parameters, [308](#)
- Omega0\_8
  - Galileo\_Almanac\_Helper, [604](#)
- OMEGA0\_8\_LSB
  - core\_system\_parameters, [308](#)
- Omega0red
  - Galileo\_Reduced\_CED, [653](#)
- OMEGA\_0
  - Gnss\_Almanac, [863](#)
  - Gnss\_Ephemeris, [876](#)
- OMEGA\_0\_2\_LSB
  - core\_system\_parameters, [308](#)
- OMEGA\_0\_LSB
  - core\_system\_parameters, [308](#)
- OMEGA\_2\_LSB
  - core\_system\_parameters, [308](#)
- omega\_7
  - Galileo\_Almanac\_Helper, [604](#)
- OMEGA\_7\_LSB
  - core\_system\_parameters, [308](#)
- omega\_8
  - Galileo\_Almanac\_Helper, [604](#)
- OMEGA\_8\_LSB
  - core\_system\_parameters, [309](#)
- omega\_9
  - Galileo\_Almanac\_Helper, [604](#)
- OMEGA\_9\_LSB
  - core\_system\_parameters, [309](#)
- Omega\_dot\_10
  - Galileo\_Almanac\_Helper, [604](#)
- OMEGA\_DOT\_10\_LSB
  - core\_system\_parameters, [309](#)
- OMEGA\_DOT\_3\_LSB
  - core\_system\_parameters, [309](#)
- Omega\_dot\_7
  - Galileo\_Almanac\_Helper, [604](#)
- OMEGA\_DOT\_7\_LSB
  - core\_system\_parameters, [309](#)
- Omega\_dot\_8
  - Galileo\_Almanac\_Helper, [604](#)
- OMEGA\_DOT\_8\_LSB
  - core\_system\_parameters, [309](#)
- OMEGA\_DOT\_LSB
  - core\_system\_parameters, [309](#)
- OMEGA\_LSB
  - core\_system\_parameters, [309](#)
- OMEGAdot
  - Gnss\_Almanac, [864](#)
  - Gnss\_Ephemeris, [876](#)
- OMG
  - tled\_t, [1489](#)
- omg
  - alm\_t, [354](#)
  - eph\_t, [522](#)
  - tled\_t, [1489](#)
- OMG0
  - alm\_t, [354](#)
  - eph\_t, [522](#)
- OMGd
  - alm\_t, [355](#)
  - eph\_t, [523](#)
- open\_channel
  - Fpga\_Multicorrelator\_8sc, [586](#)
- open\_device
  - Fpga\_Acquisition, [578](#)
- opencl\_ready
  - GpsL1CaPcpsOpenCIAcquisition, [1012](#)
  - pcps\_opencl\_acquisition\_cc, [1269](#)
- openpath
  - file\_t, [538](#)
- operator<<
  - Gnss\_Satellite, [882](#)
  - Gnss\_Signal, [897](#)
- operator()
  - Allocator, [352](#)
  - Deleter, [482](#)
- operator+
  - GPU\_Complex, [1081](#)
  - GPU\_Complex\_Short, [1083](#)
- operator+=
  - GPU\_Complex, [1081](#)
- operator=
  - Exponential\_Smoother, [528](#)
  - Gnss\_Satellite, [881](#)
  - Gnss\_Synchro, [901](#)
  - Nav\_Message\_Packet, [1188](#)
  - Serdes\_Galileo\_Eph, [1412](#)
  - Serdes\_Gnss\_Synchro, [1414](#)
  - Serdes\_Gps\_Eph, [1416](#)
  - Serdes\_Monitor\_Pvt, [1418](#)
  - Serdes\_Nav\_Message, [1420](#)
  - Tracking\_loop\_filter, [1497](#)



- operator==
  - Gnss\_Satellite, [882](#)
  - Gnss\_Signal, [897](#)
- operator\*
  - GPU\_Complex, [1081](#)
  - GPU\_Complex\_Short, [1083](#)
- opt
  - raw\_t, [1326](#)
  - rtcm\_t, [1355](#)
  - rtk\_t, [1358](#)
- opt\_t, [1222](#)
  - comment, [1222](#)
  - format, [1222](#)
  - name, [1222](#)
  - var, [1222](#)
- orbit\_correction\_flag
  - mt1\_header, [1180](#)
- osmosdr\_signal\_source.h, [2021](#), [2022](#)
- OsmosdrSignalSource, [1223](#)
  - connect, [1225](#)
  - disconnect, [1225](#)
  - get\_left\_block, [1225](#)
  - get\_right\_block, [1225](#)
  - item\_size, [1225](#)
- OSNMA\_data, [1226](#)
  - d\_dsm\_header, [1227](#)
  - d\_dsm\_kroot\_message, [1227](#)
  - d\_dsm\_kroot\_new\_message, [1227](#)
  - d\_dsm\_pkr\_message, [1227](#)
  - d\_mack\_message, [1227](#)
  - d\_nav\_data, [1227](#)
  - d\_nma\_header, [1227](#)
- osnma\_data.h, [2714](#), [2715](#)
- OSNMA\_DSM\_Reader, [1228](#)
- osnma\_dsm\_reader.h, [2718](#), [2719](#)
- Osnma\_Helper, [1229](#)
  - GST\_START\_EPOCH, [1230](#)
- osnma\_helper.h, [2358](#), [2359](#)
- OSNMA\_msg, [1230](#)
  - hkroot, [1231](#)
  - mack, [1231](#)
  - PRN, [1231](#)
  - TOW\_sf0, [1231](#)
  - WN\_sf0, [1231](#)
- osnma\_msg\_receiver, [1231](#)
  - ~osnma\_msg\_receiver, [1233](#)
  - msg\_handler\_osnma, [1233](#)
  - read\_merkle\_xml, [1233](#)
  - set\_merkle\_root, [1233](#)
  - verify\_dsm\_pkr, [1233](#)
- osnma\_msg\_receiver.h, [2360](#), [2361](#)
  - FRIEND\_TEST, [2361](#)
- osnma\_msg\_receiver\_sptr
  - core\_libs, [209](#)
- osnma\_nav\_data\_manager.h, [2364](#), [2365](#)
- OSNMA\_NavData, [1234](#)
  - get\_IOD\_nav, [1235](#)
  - get\_last\_received\_TOW, [1235](#)
  - get\_prn\_d, [1235](#)
  - get\_tow\_sf0, [1235](#)
  - get\_verified\_bits, [1235](#)
  - get\_verified\_status, [1235](#)
  - nav\_data\_id, [1236](#)
  - OSNMA\_NavData, [1235](#)
  - set\_ephemeris\_data, [1235](#)
  - set\_IOD\_nav, [1235](#)
  - set\_last\_received\_TOW, [1235](#)
  - set\_prn\_d, [1235](#)
  - set\_tow\_sf0, [1236](#)
  - set\_update\_verified\_bits, [1236](#)
  - set\_utc\_data, [1236](#)
  - set\_verified\_status, [1236](#)
- OSNMA\_NavDataManager, [1236](#)
- osnma\_strict
  - Pvt\_Conf, [1312](#)
- OSNMA\_TABLE\_1
  - core\_system\_parameters, [309](#)
- OSNMA\_TABLE\_10
  - core\_system\_parameters, [309](#)
- OSNMA\_TABLE\_11
  - core\_system\_parameters, [310](#)
- OSNMA\_TABLE\_15
  - core\_system\_parameters, [310](#)
- OSNMA\_TABLE\_16
  - core\_system\_parameters, [310](#)
- OSNMA\_TABLE\_2
  - core\_system\_parameters, [310](#)
- OSNMA\_TABLE\_3
  - core\_system\_parameters, [311](#)
- OSNMA\_TABLE\_5
  - core\_system\_parameters, [311](#)
- OSNMA\_TABLE\_6
  - core\_system\_parameters, [311](#)
- OSNMA\_TABLE\_7
  - core\_system\_parameters, [311](#)
- OSNMA\_TABLE\_8
  - core\_system\_parameters, [312](#)
- outb
  - stream\_t, [1469](#)
- outbt
  - stream\_t, [1469](#)
- outc
  - solstat\_t, [1446](#)
  - ssat\_t, [1459](#)
- outhead
  - solopt\_t, [1443](#)
- outopt
  - solopt\_t, [1443](#)
- output\_enabled
  - Pvt\_Conf, [1312](#)
- output\_path
  - Pvt\_Conf, [1312](#)
- output\_rate\_ms
  - Pvt\_Conf, [1312](#)
- outr
  - stream\_t, [1469](#)

- outsingle
  - prcopt\_t, 1300
- outtype
  - raw\_t, 1326
  - rtcm\_t, 1356
- overflow\_reg
  - PpsSamplestamp, 1295
- P
  - obsd\_t, 1218
  - rtk\_t, 1358
- p\_dk
  - DSM\_KROOT\_message, 515
- p\_dp
  - DSM\_PKR\_message, 518
- Pa
  - rtk\_t, 1358
- packetsPerSample
  - FileSourceBase, 552
  - FourBitCpxFileSignalSource, 575
  - NsrFileSignalSource, 1210
  - TwoBitCpxFileSignalSource, 1505
  - TwoBitPackedFileSignalSource, 1510
- page
  - raw\_t, 1326
- ParseError
  - INIReader, 1108
- part1
  - cnav\_msg\_decoder\_t, 448
- part2
  - cnav\_msg\_decoder\_t, 448
- Pass\_Through, 1237
  - connect, 1240
  - disconnect, 1240
  - get\_left\_block, 1240
  - get\_right\_block, 1240
  - implementation, 1240
  - item\_size, 1240
  - item\_type, 1240
  - role, 1240
- pass\_through.h, 1781, 1782
- passwd
  - ftp\_t, 595
  - ntrip\_t, 1213
- path
  - file\_t, 539
  - stream\_t, 1469
  - url\_t, 1527
- pbias
  - ssr\_t, 1462
- pbuf
  - rtksvr\_t, 1378
- pbuff
  - raw\_t, 1326
- pbyte
  - raw\_t, 1326
- pclk
  - nav\_t, 1194
- pclk\_t, 1241
  - clk, 1241
  - index, 1241
  - std, 1241
  - time, 1241
- pcps\_acquisition, 1242
  - general\_work, 1244
  - init, 1244
  - mag, 1244
  - set\_active, 1244
  - set\_channel, 1244
  - set\_channel\_fsm, 1245
  - set\_doppler\_center, 1245
  - set\_doppler\_max, 1245
  - set\_doppler\_step, 1245
  - set\_gnss\_synchro, 1245
  - set\_local\_code, 1246
  - set\_state, 1246
  - set\_threshold, 1246
- pcps\_acquisition.h, 1627, 1629
- pcps\_acquisition\_fine\_doppler\_cc, 1246
  - ~pcps\_acquisition\_fine\_doppler\_cc, 1249
  - general\_work, 1249
  - init, 1249
  - mag, 1249
  - nextPowerOf2, 1249
  - set\_active, 1251
  - set\_channel, 1251
  - set\_channel\_fsm, 1251
  - set\_doppler\_max, 1251
  - set\_doppler\_step, 1251
  - set\_gnss\_synchro, 1251
  - set\_local\_code, 1252
  - set\_state, 1252
  - set\_threshold, 1252
- pcps\_acquisition\_fine\_doppler\_cc.h, 1632, 1634
- pcps\_acquisition\_fine\_doppler\_cc\_sptr
  - acquisition\_adapters, 77
  - acquisition\_gr\_blocks, 79
- pcps\_acquisition\_fpga, 1252
  - ~pcps\_acquisition\_fpga, 1254
  - init, 1254
  - mag, 1254
  - reset\_acquisition, 1254
  - set\_active, 1254
  - set\_channel, 1254
  - set\_channel\_fsm, 1255
  - set\_doppler\_center, 1255
  - set\_doppler\_max, 1255
  - set\_doppler\_step, 1255
  - set\_gnss\_synchro, 1255
  - set\_local\_code, 1256
  - set\_state, 1256
  - set\_threshold, 1256
  - stop\_acquisition, 1256
- pcps\_acquisition\_fpga.h, 1637, 1638
- pcps\_acquisition\_fpga\_sptr
  - acquisition\_gr\_blocks, 79
- pcps\_acquisition\_sptr

- acquisition\_gr\_blocks, 79
- pcps\_assisted\_acquisition\_cc, 1256
  - ~pcps\_assisted\_acquisition\_cc, 1259
  - general\_work, 1259
  - init, 1259
  - mag, 1259
  - set\_active, 1259
  - set\_channel, 1260
  - set\_channel\_fsm, 1260
  - set\_doppler\_max, 1260
  - set\_doppler\_step, 1260
  - set\_gnss\_synchro, 1260
  - set\_local\_code, 1261
  - set\_state, 1261
  - set\_threshold, 1261
- pcps\_assisted\_acquisition\_cc.h, 1640, 1642
- pcps\_assisted\_acquisition\_cc\_sptr
  - acquisition\_gr\_blocks, 80
- pcps\_cccwsr\_acquisition\_cc, 1261
  - ~pcps\_cccwsr\_acquisition\_cc, 1264
  - general\_work, 1264
  - init, 1264
  - mag, 1264
  - set\_active, 1264
  - set\_channel, 1265
  - set\_channel\_fsm, 1265
  - set\_doppler\_max, 1265
  - set\_doppler\_step, 1265
  - set\_gnss\_synchro, 1265
  - set\_local\_code, 1265
  - set\_state, 1266
  - set\_threshold, 1266
- pcps\_cccwsr\_acquisition\_cc.h, 1645, 1646
- pcps\_cccwsr\_acquisition\_cc\_sptr
  - acquisition\_gr\_blocks, 80
- pcps\_openc1\_acquisition\_cc, 1266
  - ~pcps\_openc1\_acquisition\_cc, 1269
  - general\_work, 1269
  - init, 1269
  - mag, 1269
  - openc1\_ready, 1269
  - set\_active, 1269
  - set\_channel, 1270
  - set\_channel\_fsm, 1270
  - set\_doppler\_max, 1270
  - set\_doppler\_step, 1270
  - set\_gnss\_synchro, 1270
  - set\_local\_code, 1272
  - set\_state, 1272
  - set\_threshold, 1272
- pcps\_openc1\_acquisition\_cc.h, 1649, 1650
  - CL\_SILENCE\_DEPRECATED, 1650
- pcps\_openc1\_acquisition\_cc\_sptr
  - acquisition\_gr\_blocks, 80
- pcps\_quicksync\_acquisition\_cc, 1272
  - ~pcps\_quicksync\_acquisition\_cc, 1275
  - general\_work, 1275
  - init, 1275
  - mag, 1275
  - set\_active, 1275
  - set\_channel, 1276
  - set\_channel\_fsm, 1276
  - set\_doppler\_max, 1276
  - set\_doppler\_step, 1276
  - set\_gnss\_synchro, 1276
  - set\_local\_code, 1278
  - set\_state, 1278
  - set\_threshold, 1278
- pcps\_quicksync\_acquisition\_cc.h, 1654, 1655
- pcps\_quicksync\_acquisition\_cc\_sptr
  - acquisition\_gr\_blocks, 80
- pcps\_tong\_acquisition\_cc, 1278
  - ~pcps\_tong\_acquisition\_cc, 1281
  - general\_work, 1281
  - init, 1281
  - mag, 1281
  - set\_active, 1281
  - set\_channel, 1282
  - set\_channel\_fsm, 1282
  - set\_doppler\_max, 1282
  - set\_doppler\_step, 1282
  - set\_gnss\_synchro, 1282
  - set\_local\_code, 1282
  - set\_state, 1283
  - set\_threshold, 1283
- pcps\_tong\_acquisition\_cc.h, 1658, 1660
- pcps\_tong\_acquisition\_cc\_sptr
  - acquisition\_gr\_blocks, 80
- pcv
  - pcvs\_t, 1287
- pcv\_t, 1284
  - code, 1284
  - off, 1284
  - sat, 1284
  - te, 1285
  - ts, 1285
  - type, 1285
  - var, 1285
- pcvr
  - prcopt\_t, 1300
- pcvs
  - nav\_t, 1194
- pcvs\_t, 1286
  - n, 1286
  - nmax, 1286
  - pcv, 1287
- pdop
  - Monitor\_Pvt, 1174
- peph
  - nav\_t, 1194
- peph\_t, 1287
  - cov, 1288
  - index, 1288
  - pos, 1288
  - std, 1288
  - time, 1288

- vco, [1288](#)
  - vel, [1288](#)
  - vst, [1288](#)
- perr
  - exterr\_t, [529](#)
- pfa
  - Acq\_Conf, [327](#)
- pfa2
  - Acq\_Conf, [327](#)
- ph
  - ssat\_t, [1459](#)
- phase\_bias
  - Galileo\_HAS\_data, [634](#)
- phase\_bias\_cycle
  - HAS\_obs\_corrections, [1089](#)
- phase\_bias\_flag
  - mt1\_header, [1180](#)
- phase\_discontinuity\_indicator
  - Galileo\_HAS\_data, [634](#)
- phase\_unwrap
  - tracking\_libs, [203](#)
- phw
  - ssat\_t, [1459](#)
- PI\_TWO\_N19
  - core\_system\_parameters, [312](#)
- PI\_TWO\_N23
  - core\_system\_parameters, [312](#)
- PI\_TWO\_N31
  - core\_system\_parameters, [312](#)
- PI\_TWO\_N38
  - core\_system\_parameters, [312](#)
- PI\_TWO\_N43
  - core\_system\_parameters, [312](#)
- pkid
  - DSM\_KROOT\_message, [515](#)
- plen
  - raw\_t, [1326](#)
- pll\_bw\_hz
  - Dll\_Pll\_Conf, [496](#)
  - Dll\_Pll\_Conf\_Fpga, [502](#)
- pll\_bw\_narrow\_hz
  - Dll\_Pll\_Conf, [496](#)
  - Dll\_Pll\_Conf\_Fpga, [502](#)
- pll\_cloop\_two\_quadrant\_atan
  - tracking\_libs, [204](#)
- pll\_filter\_order
  - Dll\_Pll\_Conf, [496](#)
  - Dll\_Pll\_Conf\_Fpga, [503](#)
- pll\_four\_quadrant\_atan
  - tracking\_libs, [204](#)
- pll\_pull\_in\_bw\_hz
  - Dll\_Pll\_Conf, [496](#)
  - Dll\_Pll\_Conf\_Fpga, [503](#)
- plutosdr\_signal\_source.h, [2023](#), [2024](#)
- PlutosdrSignalSource, [1289](#)
  - connect, [1291](#)
  - disconnect, [1291](#)
  - get\_left\_block, [1291](#)
  - get\_right\_block, [1291](#)
  - item\_size, [1291](#)
- PMODE\_DGPS
  - algorithms\_libs\_rtklib, [158](#)
- PMODE\_FIXED
  - algorithms\_libs\_rtklib, [158](#)
- PMODE\_KINEMA
  - algorithms\_libs\_rtklib, [159](#)
- PMODE\_MOVEB
  - algorithms\_libs\_rtklib, [159](#)
- PMODE\_PPP\_FIXED
  - algorithms\_libs\_rtklib, [159](#)
- PMODE\_PPP\_KINEMA
  - algorithms\_libs\_rtklib, [159](#)
- PMODE\_PPP\_STATIC
  - algorithms\_libs\_rtklib, [159](#)
- PMODE\_SINGLE
  - algorithms\_libs\_rtklib, [159](#)
- PMODE\_STATIC
  - algorithms\_libs\_rtklib, [160](#)
- pntpos
  - rtklib\_pntpos.h, [1819](#)
- poly
  - v27\_t, [1530](#)
- POLYCRC24Q
  - algorithms\_libs\_rtklib, [160](#)
- POLYCRC32
  - algorithms\_libs\_rtklib, [160](#)
- pop\_front
  - algorithms\_libs, [110](#)
- port
  - stream\_t, [1470](#)
  - tcp\_t, [1477](#)
- pos
  - geph\_t, [789](#)
  - lexeph\_t, [1149](#)
  - peph\_t, [1288](#)
  - seph\_t, [1410](#)
  - sta\_t, [1465](#)
- pos0
  - lexion\_t, [1150](#)
- pos\_x
  - Monitor\_Pvt, [1174](#)
- pos\_y
  - Monitor\_Pvt, [1174](#)
- pos\_z
  - Monitor\_Pvt, [1174](#)
- posf
  - soloft\_t, [1443](#)
- posopt
  - prcopt\_t, [1300](#)
- POSOPT\_RINEX
  - algorithms\_libs\_rtklib, [160](#)
- post\_disconnect\_hook
  - SpirFileSignalSource, [1452](#)
- pppcorr
  - nav\_t, [1195](#)
- pppcorr\_t, [1292](#)

- ns, [1293](#)
- nsmax, [1293](#)
- nsta, [1293](#)
- nt, [1293](#)
- ntmax, [1293](#)
- rr, [1293](#)
- stas, [1293](#)
- stec, [1293](#)
- trop, [1293](#)
- pppopt
  - prcopt\_t, [1300](#)
- pps\_samplestamp.h, [2106](#), [2107](#)
- pps\_tcp\_rx, [1294](#)
  - is\_connected, [1294](#)
- PpsSamplestamp, [1295](#)
  - overflow\_reg, [1295](#)
  - samplestamp, [1295](#)
- ppstcprx.h, [2107](#), [2108](#)
- prc
  - dgps\_t, [483](#)
  - sbsfcrr\_t, [1397](#)
- prCA
  - raw\_t, [1326](#)
- PRCOPT\_DEFAULT
  - rtklib\_rtksvr.h, [1854](#)
- prcopt\_t, [1296](#)
  - antdel, [1298](#)
  - anttype, [1298](#)
  - armaxiter, [1298](#)
  - bancroft\_init, [1298](#)
  - baseline, [1298](#)
  - bdsmodear, [1298](#)
  - codesmooth, [1298](#)
  - dynamics, [1298](#)
  - elmaskar, [1298](#)
  - elmaskhold, [1298](#)
  - elmin, [1298](#)
  - eratio, [1298](#)
  - err, [1298](#)
  - exsats, [1299](#)
  - exterr, [1299](#)
  - freqopt, [1299](#)
  - glomodear, [1299](#)
  - initrst, [1299](#)
  - intpref, [1299](#)
  - ionoopt, [1299](#)
  - maxaveep, [1299](#)
  - maxgdop, [1299](#)
  - maxinno, [1299](#)
  - maxout, [1299](#)
  - maxtdiff, [1299](#)
  - minfix, [1300](#)
  - minlock, [1300](#)
  - mode, [1300](#)
  - modear, [1300](#)
  - navsys, [1300](#)
  - nf, [1300](#)
  - niter, [1300](#)
  - odisp, [1300](#)
  - outsingle, [1300](#)
  - pcvr, [1300](#)
  - posopt, [1300](#)
  - pppopt, [1300](#)
  - prn, [1301](#)
  - rb, [1301](#)
  - refpos, [1301](#)
  - rnxopt, [1301](#)
  - rovpos, [1301](#)
  - ru, [1301](#)
  - sateph, [1301](#)
  - sbascorr, [1301](#)
  - sbassatsel, [1301](#)
  - sclkstab, [1301](#)
  - snrmask, [1301](#)
  - soltype, [1301](#)
  - std, [1302](#)
  - syncsol, [1302](#)
  - thresar, [1302](#)
  - thresslip, [1302](#)
  - tidecorr, [1302](#)
  - tropopt, [1302](#)
- prcout
  - rtksvr\_t, [1378](#)
- pre\_2009\_file
  - Pvt\_Conf, [1312](#)
- pre\_connect\_hook
  - FileTimestampSignalSource, [557](#)
  - FourBitCpxFileSignalSource, [575](#)
  - NsrFileSignalSource, [1210](#)
  - SpirFileSignalSource, [1452](#)
  - TwoBitCpxFileSignalSource, [1505](#)
  - TwoBitPackedFileSignalSource, [1510](#)
- pre\_disconnect\_hook
  - FileTimestampSignalSource, [557](#)
  - FourBitCpxFileSignalSource, [575](#)
  - NsrFileSignalSource, [1210](#)
  - TwoBitCpxFileSignalSource, [1505](#)
  - TwoBitPackedFileSignalSource, [1510](#)
- preamble\_seen
  - cnav\_v27\_part\_t, [452](#)
- predicted\_doppler
  - Gnss\_Almanac, [862](#)
  - Gnss\_Ephemeris, [873](#)
- print\_IGM01
  - Rtcm, [1342](#)
- print\_IGM02
  - Rtcm, [1342](#)
- print\_IGM03
  - Rtcm, [1342](#)
- print\_IGM05
  - Rtcm, [1343](#)
- print\_MSM\_1
  - Rtcm, [1343](#)
- print\_MSM\_2
  - Rtcm, [1343](#)
- print\_MSM\_3

- Rtcm, [1343](#)
- print\_MSM\_4
  - Rtcm, [1344](#)
- print\_MSM\_5
  - Rtcm, [1344](#)
- print\_MSM\_6
  - Rtcm, [1344](#)
- print\_MSM\_7
  - Rtcm, [1344](#)
- print\_MT1001
  - Rtcm, [1345](#)
- print\_MT1002
  - Rtcm, [1345](#)
- print\_MT1003
  - Rtcm, [1345](#)
- print\_MT1004
  - Rtcm, [1345](#)
- print\_MT1005
  - Rtcm, [1345](#)
- print\_MT1005\_test
  - Rtcm, [1346](#)
  - Rtcm\_Printer, [1351](#)
- print\_MT1006
  - Rtcm, [1346](#)
- print\_MT1008
  - Rtcm, [1346](#)
- print\_MT1009
  - Rtcm, [1346](#)
- print\_MT1010
  - Rtcm, [1347](#)
- print\_MT1011
  - Rtcm, [1347](#)
- print\_MT1012
  - Rtcm, [1347](#)
- print\_MT1019
  - Rtcm, [1348](#)
- print\_MT1020
  - Rtcm, [1348](#)
- print\_MT1029
  - Rtcm, [1348](#)
- print\_MT1045
  - Rtcm, [1348](#)
- Print\_Nmea\_Line
  - Nmea\_Printer, [1197](#)
- print\_packet
  - An\_Packet\_Printer, [358](#)
- print\_rinex\_annotation
  - Rinex\_Printer, [1335](#)
- Print\_Rtcm\_Messages
  - Rtcm\_Printer, [1351](#)
- priorize\_satellites
  - GNSSFlowgraph, [914](#)
- PRN
  - Galileo\_Reduced\_CED, [653](#)
  - Glonass\_Gnav\_Almanac, [794](#)
  - Glonass\_Gnav\_Ephemeris, [804](#)
  - Gnss\_Almanac, [864](#)
  - Gnss\_Ephemeris, [876](#)
  - Gnss\_Synchro, [904](#)
  - Gps\_Acq\_Assist, [918](#)
  - OSNMA\_msg, [1231](#)
- prn
  - cnav\_msg\_t, [449](#)
  - lexmsg\_t, [1152](#)
  - Nav\_Message\_Packet, [1189](#)
  - prcopt\_t, [1301](#)
  - sbsmsg\_t, [1403](#)
  - sdr\_gnss\_packet\_t, [1408](#)
- PRN\_d
  - MACK\_tag\_info, [1162](#)
  - Tag, [1474](#)
- PRN\_HWBIA
  - algorithms\_libs\_rtklib, [160](#)
- PRN\_HWBIAS
  - algorithms\_libs\_rtklib, [160](#)
- PRNa
  - MACK\_message, [1160](#)
  - Tag, [1474](#)
- proc\_pack\_carr\_error
  - Tcp\_Packet\_Data, [1476](#)
- proc\_pack\_carrier\_doppler\_hz
  - Tcp\_Packet\_Data, [1476](#)
- proc\_pack\_code\_error
  - Tcp\_Packet\_Data, [1476](#)
- process\_test\_page
  - galileo\_e6\_has\_msg\_receiver, [616](#)
- processed\_control\_messages
  - ControlThread, [466](#)
- prog
  - solo\_t, [1443](#)
- program
  - cl\_fft\_plan, [443](#)
- Prompt\_I
  - Gnss\_Synchro, [904](#)
- Prompt\_Q
  - Gnss\_Synchro, [904](#)
- property
  - FileConfiguration, [542](#), [543](#)
  - InMemoryConfiguration, [1111](#), [1112](#)
- proto
  - ftp\_t, [595](#)
- protobuf\_enabled
  - Pvt\_Conf, [1312](#)
- proxy\_status
  - fpga\_dma-proxy.h, [2081](#)
- PRUNIT\_GLO
  - rtklib\_rtc3.h, [1837](#)
- PRUNIT\_GPS
  - rtklib\_rtc3.h, [1837](#)
- Pseudorange\_m
  - Gnss\_Synchro, [904](#)
- pt
  - ssat\_t, [1459](#)
- pull\_in\_time\_s
  - Dll\_Pll\_Conf, [496](#)
  - Dll\_Pll\_Conf\_Fpga, [503](#)

- Kf\_Conf, 1138
- pulse\_blanking\_cc, 1303
- pulse\_blanking\_cc.h, 1717, 1718
- pulse\_blanking\_cc\_sptr
  - input\_filter\_gr\_blocks, 89
- pulse\_blanking\_filter.h, 1709, 1710
- PulseBlankingFilter, 1304
  - connect, 1306
  - disconnect, 1306
  - get\_left\_block, 1306
  - get\_right\_block, 1306
  - implementation, 1306
  - item\_size, 1306
  - role, 1306
- push
  - Concurrent\_Queue< Data >, 458, 459
- push\_back
  - algorithms\_libs, 110
- pv\_Geo\_to\_ECEF
  - algorithms\_libs, 110
- PVT, 118
- pvt\_adapters, 169
- Pvt\_Conf, 1307
  - an\_dump\_devname, 1309
  - an\_output\_enabled, 1309
  - an\_rate\_ms, 1309
  - display\_rate\_ms, 1309
  - dump, 1309
  - dump\_filename, 1309
  - dump\_mat, 1309
  - enable\_pvt\_kf, 1309
  - enable\_rx\_clock\_correction, 1309
  - flag\_nmea\_tty\_port, 1309
  - flag\_rtcn\_server, 1310
  - flag\_rtcn\_tty\_port, 1310
  - geojson\_output\_enabled, 1310
  - geojson\_output\_path, 1310
  - geojson\_rate\_ms, 1310
  - gpx\_output\_enabled, 1310
  - gpx\_output\_path, 1310
  - gpx\_rate\_ms, 1310
  - has\_output\_file\_path, 1310
  - kml\_output\_enabled, 1310
  - kml\_output\_path, 1310
  - kml\_rate\_ms, 1310
  - log\_source\_timetag, 1311
  - log\_source\_timetag\_file, 1311
  - max\_obs\_block\_rx\_clock\_offset\_ms, 1311
  - measures\_ecef\_pos\_sd\_m, 1311
  - measures\_ecef\_vel\_sd\_ms, 1311
  - monitor\_enabled, 1311
  - monitor\_ephemeris\_enabled, 1311
  - nmea\_dump\_devname, 1311
  - nmea\_dump\_filename, 1311
  - nmea\_output\_file\_enabled, 1311
  - nmea\_output\_file\_path, 1311
  - nmea\_rate\_ms, 1311
  - observable\_interval\_ms, 1312
  - osnma\_strict, 1312
  - output\_enabled, 1312
  - output\_path, 1312
  - output\_rate\_ms, 1312
  - pre\_2009\_file, 1312
  - protobuf\_enabled, 1312
  - rinex\_name, 1312
  - rinex\_output\_enabled, 1312
  - rinex\_output\_path, 1312
  - rinex\_version, 1312
  - rinexobs\_rate\_ms, 1312
  - rtcm\_dump\_devname, 1313
  - rtcm\_msg\_rate\_ms, 1313
  - rtcm\_output\_file\_enabled, 1313
  - rtcm\_output\_file\_path, 1313
  - rtcm\_station\_id, 1313
  - rtcm\_tcp\_port, 1313
  - rtk\_trace\_level, 1313
  - show\_local\_time\_zone, 1313
  - system\_ecef\_pos\_sd\_m, 1313
  - system\_ecef\_vel\_sd\_ms, 1313
  - type\_of\_receiver, 1313
  - udp\_addresses, 1313
  - udp\_eph\_addresses, 1314
  - udp\_eph\_port, 1314
  - udp\_ports, 1314
  - use\_e6\_for\_pvt, 1314
  - use\_has\_corrections, 1314
  - use\_unhealthy\_sats, 1314
  - xml\_output\_enabled, 1314
  - xml\_output\_path, 1314
- pvt\_conf.h, 1909, 1910
- pvt\_gr\_blocks, 170
  - rtklib\_pvt\_gs\_sptr, 170
- pvt\_interface.h, 2321, 2322
- Pvt\_Kf, 1314
- pvt\_kf.h, 1911, 1912
- pvt\_libs, 170
  - asInt, 172
  - asString, 172
  - b\_io\_context, 172
- pvt\_sol
  - Rtklib\_Solver, 1374
- Pvt\_Solution, 1315
  - get\_clock\_drift\_ppm, 1318
  - get\_course\_over\_ground, 1318
  - get\_height, 1318
  - get\_latitude, 1318
  - get\_longitude, 1318
  - get\_num\_valid\_observations, 1318
  - get\_speed\_over\_ground, 1318
  - get\_time\_offset\_s, 1319
  - set\_clock\_drift\_ppm, 1319
  - set\_course\_over\_ground, 1319
  - set\_num\_valid\_observations, 1319
  - set\_pre\_2009\_file, 1319
  - set\_rx\_pos, 1319
  - set\_rx\_vel, 1319

- set\_speed\_over\_ground, [1319](#)
  - set\_time\_offset\_s, [1319](#)
- pvt\_solution.h, [1913](#), [1914](#)
- pvt\_ssat
  - Rtklib\_Solver, [1374](#)
- PvtInterface, [1320](#)
- qr
  - sol\_t, [1439](#)
- r
  - GPU\_Complex, [1082](#)
  - GPU\_Complex\_Short, [1083](#)
- R2D
  - core\_system\_parameters, [312](#)
- radial\_m
  - HAS\_orbit\_corrections, [1090](#)
- RANGE\_MS
  - rtklib\_rtc3.h, [1837](#)
- ratio
  - sol\_t, [1439](#)
- raw
  - rtksvr\_t, [1378](#)
- raw\_array\_signal\_source.h, [2025](#), [2026](#)
- raw\_msg
  - cnav\_msg\_t, [449](#)
- raw\_t, [1323](#)
  - buff, [1324](#)
  - dpCA, [1324](#)
  - ephSAT, [1324](#)
  - flag, [1324](#)
  - freqn, [1324](#)
  - halfc, [1325](#)
  - icpc, [1325](#)
  - icpp, [1325](#)
  - iod, [1325](#)
  - len, [1325](#)
  - lexmsg, [1325](#)
  - lockt, [1325](#)
  - msgtype, [1325](#)
  - nav, [1325](#)
  - nbyte, [1325](#)
  - obs, [1325](#)
  - obuf, [1325](#)
  - off, [1326](#)
  - opt, [1326](#)
  - outtype, [1326](#)
  - page, [1326](#)
  - pbuff, [1326](#)
  - pbyte, [1326](#)
  - plen, [1326](#)
  - prCA, [1326](#)
  - receive\_time, [1326](#)
  - reply, [1326](#)
  - sbsmsg, [1326](#)
  - sta, [1326](#)
  - subfrm, [1327](#)
  - tbase, [1327](#)
  - time, [1327](#)
  - tobs, [1327](#)
  - tod, [1327](#)
  - week, [1327](#)
- RawArraySignalSource, [1327](#)
  - connect, [1330](#)
  - disconnect, [1330](#)
  - get\_left\_block, [1330](#)
  - get\_right\_block, [1330](#)
  - item\_size, [1330](#)
- rb
  - prcopt\_t, [1301](#)
  - rtk\_t, [1359](#)
  - solbuf\_t, [1441](#)
  - tec\_t, [1483](#)
- rbias
  - nav\_t, [1195](#)
- rcv
  - obsd\_t, [1218](#)
- RE\_WGS84
  - algorithms\_libs\_rtklib, [161](#)
- read
  - Concurrent\_Map< Data >, [456](#)
- read\_acquisition\_results
  - Fpga\_Acquisition, [578](#)
- read\_merkle\_xml
  - osnma\_msg\_receiver, [1233](#)
- read\_MT1005
  - Rtcm, [1349](#)
- read\_MT1019
  - Rtcm, [1349](#)
- read\_MT1020
  - Rtcm, [1349](#)
- read\_MT1045
  - Rtcm, [1349](#)
- read\_reg32
  - Fpga\_spidev, [589](#)
- read\_sample\_counter
  - Fpga\_Multicorrelator\_8sc, [586](#)
- readProtobuffer
  - Serdes\_Galileo\_Eph, [1412](#)
  - Serdes\_Gnss\_Synchro, [1414](#)
  - Serdes\_Gps\_Eph, [1416](#)
  - Serdes\_Monitor\_Pvt, [1418](#)
  - Serdes\_Nav\_Message, [1420](#)
- real
  - cIFFT\_Complex, [444](#)
  - cIFFT\_SplitComplex, [446](#)
- receive\_time
  - raw\_t, [1326](#)
- received\_tag
  - Tag, [1474](#)
- recsno
  - sta\_t, [1465](#)
- rectype
  - sta\_t, [1465](#)
- recver
  - sta\_t, [1465](#)
- reed\_solomon.h, [2720](#), [2721](#)



- ReedSolomon, [1331](#)
  - decode, [1332](#)
  - encode\_with\_generator\_matrix, [1332](#)
  - encode\_with\_generator\_poly, [1332](#)
  - ReedSolomon, [1331](#), [1332](#)
- refd
  - ssr\_t, [1462](#)
- Reference Documents, [15](#)
- refpos
  - prcopt\_t, [1301](#)
- Region1\_flag
  - Galileo\_Iono, [643](#)
- Region2\_flag
  - Galileo\_Iono, [643](#)
- Region3\_flag
  - Galileo\_Iono, [643](#)
- Region4\_flag
  - Galileo\_Iono, [643](#)
- Region5\_flag
  - Galileo\_Iono, [643](#)
- rejc
  - solstat\_t, [1446](#)
  - ssat\_t, [1460](#)
- REL\_HUMI
  - algorithms\_libs\_rtklib, [161](#)
- remove\_dat
  - Tlm\_Conf, [1491](#)
- repeat
  - FileSourceBase, [552](#)
  - MultichannelFileSignalSource, [1184](#)
  - SpirGSS6450FileSignalSource, [1457](#)
- repeat\_satellite
  - Acq\_Conf\_Fpga, [331](#)
- reply
  - raw\_t, [1326](#)
- repmode
  - file\_t, [539](#)
- request
  - Gnss\_Sdr\_Supl\_Client, [891](#)
- resampled\_fs
  - Acq\_Conf, [327](#)
  - Acq\_Conf\_Fpga, [332](#)
- Resampler, [175](#)
- resampler
  - algorithms\_libs, [111](#)
  - ArraySignalConditioner, [363](#)
  - SignalConditioner, [1428](#)
- resampler\_adapters, [176](#)
- resampler\_gr\_blocks, [176](#)
  - direct\_resampler\_conditioner\_cb\_sptr, [177](#)
  - direct\_resampler\_conditioner\_cc\_sptr, [177](#)
  - direct\_resampler\_conditioner\_cs\_sptr, [177](#)
- resampler\_latency\_samples
  - Acq\_Conf, [327](#)
- resampler\_ratio
  - Acq\_Conf, [327](#)
- resc
  - solstat\_t, [1446](#)
  - ssat\_t, [1460](#)
- reserved
  - DSM\_KROOT\_message, [515](#)
  - DSM\_nma\_header, [516](#)
  - Galileo\_HAS\_page, [637](#)
  - mt1\_header, [1180](#)
  - sdr\_gnss\_packet\_t, [1408](#)
- reserved1
  - DSM\_KROOT\_message, [515](#)
- reset
  - AcquisitionInterface, [334](#)
  - algorithms\_libs, [111](#)
  - BeidouB1iPcpsAcquisition, [401](#)
  - BeidouB1iTelemetryDecoder, [406](#)
  - BeidouB3iPcpsAcquisition, [415](#)
  - BeidouB3iTelemetryDecoder, [420](#)
  - dll\_pll\_veml\_tracking\_fpga, [508](#)
  - GalileoE1BTelemetryDecoder, [663](#)
  - GalileoE1Pcps8msAmbiguousAcquisition, [677](#)
  - GalileoE1PcpsAmbiguousAcquisition, [683](#)
  - GalileoE1PcpsAmbiguousAcquisitionFpga, [690](#)
  - GalileoE1PcpsCccwsrAmbiguousAcquisition, [695](#)
  - GalileoE1PcpsQuickSyncAmbiguousAcquisition, [701](#)
  - GalileoE1PcpsTongAmbiguousAcquisition, [707](#)
  - GalileoE5aNoncoherentIQAcquisitionCaf, [726](#)
  - GalileoE5aPcpsAcquisition, [732](#)
  - GalileoE5aPcpsAcquisitionFpga, [739](#)
  - GalileoE5aTelemetryDecoder, [744](#)
  - GalileoE5bPcpsAcquisition, [754](#)
  - GalileoE5bPcpsAcquisitionFpga, [761](#)
  - GalileoE5bTelemetryDecoder, [767](#)
  - GalileoE6PcpsAcquisition, [775](#)
  - GalileoE6TelemetryDecoder, [781](#)
  - glonass\_l1\_ca\_telemetry\_decoder\_gs, [818](#)
  - glonass\_l2\_ca\_telemetry\_decoder\_gs, [825](#)
  - GlonassL1CaPcpsAcquisition, [837](#)
  - GlonassL1CaTelemetryDecoder, [841](#)
  - GlonassL2CaPcpsAcquisition, [854](#)
  - GlonassL2CaTelemetryDecoder, [858](#)
  - GpsL1CaPcpsAcquisition, [989](#)
  - GpsL1CaPcpsAcquisitionFineDoppler, [995](#)
  - GpsL1CaPcpsAcquisitionFpga, [1001](#)
  - GpsL1CaPcpsAssistedAcquisition, [1007](#)
  - GpsL1CaPcpsOpenCIAcquisition, [1012](#)
  - GpsL1CaPcpsQuickSyncAcquisition, [1018](#)
  - GpsL1CaPcpsTongAcquisition, [1023](#)
  - GpsL1CaTelemetryDecoder, [1031](#)
  - GpsL2CTelemetryDecoder, [1035](#)
  - GpsL2MPcpsAcquisition, [1048](#)
  - GpsL2MPcpsAcquisitionFpga, [1054](#)
  - GpsL5iPcpsAcquisition, [1068](#)
  - GpsL5iPcpsAcquisitionFpga, [1075](#)
  - GpsL5TelemetryDecoder, [1079](#)
  - HybridObservables, [1096](#)
  - Rtklib\_Pvt, [1364](#)
  - sbas\_l1\_telemetry\_decoder\_gs, [1390](#)
  - SbasL1TelemetryDecoder, [1393](#)

- Viterbi\_Decoder, [1532](#)
- reset\_acquisition
  - Fpga\_Acquisition, [578](#)
  - pcps\_acquisition\_fpga, [1254](#)
- resp
  - solstat\_t, [1446](#)
  - ssat\_t, [1460](#)
- rev
  - tled\_t, [1489](#)
- rfport
  - stream\_cfg, [1468](#)
- rinex\_name
  - Pvt\_Conf, [1312](#)
- rinex\_output\_enabled
  - Pvt\_Conf, [1312](#)
- rinex\_output\_path
  - Pvt\_Conf, [1312](#)
- Rinex\_Printer, [1333](#)
  - ~Rinex\_Printer, [1334](#)
  - get\_navfilename, [1334](#)
  - get\_obsfilename, [1334](#)
  - is\_rinex\_header\_written, [1334](#)
  - log\_rinex\_nav\_bds\_dnav, [1334](#)
  - log\_rinex\_nav\_gal\_nav, [1334](#)
  - log\_rinex\_nav\_glo\_gnav, [1335](#)
  - log\_rinex\_nav\_gps\_cnav, [1335](#)
  - log\_rinex\_nav\_gps\_nav, [1335](#)
  - print\_rinex\_annotation, [1335](#)
  - Rinex\_Printer, [1334](#)
  - set\_pre\_2009\_file, [1336](#)
- rinex\_printer.h, [1916](#)
- rinex\_version
  - Pvt\_Conf, [1312](#)
- rinexobs\_rate\_ms
  - Pvt\_Conf, [1312](#)
- rms
  - tec\_t, [1483](#)
- rxnpxopt
  - prcopt\_t, [1301](#)
- role
  - ArraySignalConditioner, [363](#)
  - BeamformerFilter, [368](#)
  - BeidouB1iDIIPIITracking, [396](#)
  - BeidouB1iPcpsAcquisition, [401](#)
  - BeidouB1iTelemetryDecoder, [407](#)
  - BeidouB3iDIIPIITracking, [410](#)
  - BeidouB3iPcpsAcquisition, [415](#)
  - BeidouB3iTelemetryDecoder, [421](#)
  - ByteToShort, [425](#)
  - Channel, [430](#)
  - CshortToGrComplex, [475](#)
  - DirectResamplerConditioner, [491](#)
  - FirFilter, [561](#)
  - FreqXlatingFirFilter, [593](#)
  - GalileoE1BTelemetryDecoder, [664](#)
  - GalileoE1DIIPIIVemlTracking, [668](#)
  - GalileoE1DIIPIIVemlTrackingFpga, [672](#)
  - GalileoE1Pcps8msAmbiguousAcquisition, [677](#)
  - GalileoE1PcpsAmbiguousAcquisition, [683](#)
  - GalileoE1PcpsAmbiguousAcquisitionFpga, [690](#)
  - GalileoE1PcpsCccwsrAmbiguousAcquisition, [695](#)
  - GalileoE1PcpsQuickSyncAmbiguousAcquisition, [701](#)
  - GalileoE1PcpsTongAmbiguousAcquisition, [707](#)
  - GalileoE1TcpConnectorTracking, [713](#)
  - GalileoE5aDIIPIITracking, [717](#)
  - GalileoE5aDIIPIITrackingFpga, [721](#)
  - GalileoE5aNoncoherentIQAcquisitionCaf, [726](#)
  - GalileoE5aPcpsAcquisition, [732](#)
  - GalileoE5aPcpsAcquisitionFpga, [739](#)
  - GalileoE5aTelemetryDecoder, [745](#)
  - GalileoE5bDIIPIITracking, [749](#)
  - GalileoE5bPcpsAcquisition, [754](#)
  - GalileoE5bPcpsAcquisitionFpga, [761](#)
  - GalileoE5bTelemetryDecoder, [767](#)
  - GalileoE6DIIPIITracking, [771](#)
  - GalileoE6PcpsAcquisition, [775](#)
  - GalileoE6TelemetryDecoder, [781](#)
  - GenSignalSource, [785](#)
  - GlionassL1CaDIIPIICAidTracking, [829](#)
  - GlionassL1CaDIIPIITracking, [833](#)
  - GlionassL1CaPcpsAcquisition, [837](#)
  - GlionassL1CaTelemetryDecoder, [842](#)
  - GlionassL2CaDIIPIICAidTracking, [846](#)
  - GlionassL2CaDIIPIITracking, [850](#)
  - GlionassL2CaPcpsAcquisition, [854](#)
  - GlionassL2CaTelemetryDecoder, [859](#)
  - GNSSBlockInterface, [911](#)
  - GpsL1CaDIIPIITracking, [968](#)
  - GpsL1CaDIIPIITrackingFpga, [972](#)
  - GpsL1CaDIIPIITrackingGPU, [977](#)
  - GpsL1CaGaussianTracking, [981](#)
  - GpsL1CaKfTracking, [985](#)
  - GpsL1CaPcpsAcquisition, [989](#)
  - GpsL1CaPcpsAcquisitionFineDoppler, [995](#)
  - GpsL1CaPcpsAcquisitionFpga, [1001](#)
  - GpsL1CaPcpsAssistedAcquisition, [1007](#)
  - GpsL1CaPcpsOpenCIAcquisition, [1012](#)
  - GpsL1CaPcpsQuickSyncAcquisition, [1018](#)
  - GpsL1CaPcpsTongAcquisition, [1023](#)
  - GpsL1CaTcpConnectorTracking, [1028](#)
  - GpsL1CaTelemetryDecoder, [1032](#)
  - GpsL2CTelemetryDecoder, [1036](#)
  - GpsL2MDIIPIITracking, [1040](#)
  - GpsL2MDIIPIITrackingFpga, [1044](#)
  - GpsL2MPcpsAcquisition, [1048](#)
  - GpsL2MPcpsAcquisitionFpga, [1054](#)
  - GpsL5DIIPIITracking, [1059](#)
  - GpsL5DIIPIITrackingFpga, [1063](#)
  - GpsL5iPcpsAcquisition, [1068](#)
  - GpsL5iPcpsAcquisitionFpga, [1075](#)
  - GpsL5TelemetryDecoder, [1080](#)
  - HybridObservables, [1097](#)
  - IbyteToCbyte, [1100](#)
  - IbyteToComplex, [1103](#)
  - IbyteToCshort, [1106](#)

- lshortToComplex, [1128](#)
- lshortToCshort, [1131](#)
- MmseResamplerConditioner, [1168](#)
- NotchFilter, [1201](#)
- NotchFilterLite, [1204](#)
- Pass\_Through, [1240](#)
- PulseBlankingFilter, [1306](#)
- Rtklib\_Pvt, [1364](#)
- SbasL1TelemetryDecoder, [1394](#)
- SignalConditioner, [1428](#)
- SignalGenerator, [1431](#)
- SignalSourceBase, [1435](#)
- ROUND\_LAMBDA
  - rtklib\_lambda.h, [1816](#)
- ROUND\_PPP
  - rtklib\_ppp.h, [1823](#)
- rovpos
  - prcopt\_t, [1301](#)
- rr
  - pppcorr\_t, [1293](#)
  - sol\_t, [1439](#)
- rrc
  - dgps\_t, [483](#)
  - sbsfcorr\_t, [1397](#)
- Rtcm, [1337](#)
  - bin\_to\_binary\_data, [1341](#)
  - bin\_to\_double, [1341](#)
  - bin\_to\_hex, [1341](#)
  - bin\_to\_uint, [1341](#)
  - binary\_data\_to\_bin, [1341](#)
  - check\_CRC, [1341](#)
  - hex\_to\_bin, [1341](#)
  - hex\_to\_int, [1341](#)
  - hex\_to\_uint, [1341](#)
  - is\_server\_running, [1341](#)
  - lock\_time, [1341](#), [1342](#)
  - print\_IGM01, [1342](#)
  - print\_IGM02, [1342](#)
  - print\_IGM03, [1342](#)
  - print\_IGM05, [1343](#)
  - print\_MSM\_1, [1343](#)
  - print\_MSM\_2, [1343](#)
  - print\_MSM\_3, [1343](#)
  - print\_MSM\_4, [1344](#)
  - print\_MSM\_5, [1344](#)
  - print\_MSM\_6, [1344](#)
  - print\_MSM\_7, [1344](#)
  - print\_MT1001, [1345](#)
  - print\_MT1002, [1345](#)
  - print\_MT1003, [1345](#)
  - print\_MT1004, [1345](#)
  - print\_MT1005, [1345](#)
  - print\_MT1005\_test, [1346](#)
  - print\_MT1006, [1346](#)
  - print\_MT1008, [1346](#)
  - print\_MT1009, [1346](#)
  - print\_MT1010, [1347](#)
  - print\_MT1011, [1347](#)
  - print\_MT1012, [1347](#)
  - print\_MT1019, [1348](#)
  - print\_MT1020, [1348](#)
  - print\_MT1029, [1348](#)
  - print\_MT1045, [1348](#)
  - read\_MT1005, [1349](#)
  - read\_MT1019, [1349](#)
  - read\_MT1020, [1349](#)
  - read\_MT1045, [1349](#)
  - Rtcm, [1340](#)
  - run\_server, [1349](#)
  - send\_message, [1349](#)
  - stop\_server, [1350](#)
- rtcm
  - rtksvr\_t, [1378](#)
- rtcm.h, [1931](#), [1932](#)
- RTCM2PREAMB
  - rtklib\_rtcm.h, [1831](#)
- RTCM3PREAMB
  - rtklib\_rtcm.h, [1831](#)
- rtcm\_dump\_devname
  - Pvt\_Conf, [1313](#)
- rtcm\_msg\_rate\_ms
  - Pvt\_Conf, [1313](#)
- rtcm\_output\_file\_enabled
  - Pvt\_Conf, [1313](#)
- rtcm\_output\_file\_path
  - Pvt\_Conf, [1313](#)
- Rtcm\_Printer, [1350](#)
  - ~Rtcm\_Printer, [1351](#)
  - lock\_time, [1351](#)
  - print\_MT1005\_test, [1351](#)
  - Print\_Rtcm\_Messages, [1351](#)
  - Rtcm\_Printer, [1351](#)
- rtcm\_printer.h, [1952](#), [1953](#)
- rtcm\_station\_id
  - Pvt\_Conf, [1313](#)
- rtcm\_t, [1353](#)
  - buff, [1354](#)
  - cp, [1354](#)
  - dgps, [1354](#)
  - ephsat, [1354](#)
  - len, [1354](#)
  - lltime, [1354](#)
  - lock, [1354](#)
  - loss, [1355](#)
  - msg, [1355](#)
  - msgtype, [1355](#)
  - msmtype, [1355](#)
  - nav, [1355](#)
  - nbit, [1355](#)
  - nbyte, [1355](#)
  - nmsg2, [1355](#)
  - nmsg3, [1355](#)
  - obs, [1355](#)
  - obsflag, [1355](#)
  - opt, [1355](#)
  - outtype, [1356](#)

- seqno, [1356](#)
- ssr, [1356](#)
- sta, [1356](#)
- stah, [1356](#)
- staid, [1356](#)
- time, [1356](#)
- time\_s, [1356](#)
- word, [1356](#)
- rtcm\_tcp\_port
  - Pvt\_Conf, [1313](#)
- rtk
  - rtksvr\_t, [1378](#)
- rtk\_lock
  - algorithms\_libs\_rtklib, [136](#)
- rtk\_t, [1357](#)
  - ambc, [1358](#)
  - errbuf, [1358](#)
  - na, [1358](#)
  - neb, [1358](#)
  - nfix, [1358](#)
  - nx, [1358](#)
  - opt, [1358](#)
  - P, [1358](#)
  - Pa, [1358](#)
  - rb, [1359](#)
  - sol, [1359](#)
  - ssat, [1359](#)
  - tt, [1359](#)
  - x, [1359](#)
  - xa, [1359](#)
- rtk\_trace\_level
  - Pvt\_Conf, [1313](#)
- rtk\_unlock
  - algorithms\_libs\_rtklib, [136](#)
- rtklib.h, [1783](#), [1792](#)
- rtklib\_conversions.h, [1808](#), [1809](#)
- rtklib\_ephemeris.h, [1810](#), [1812](#)
- rtklib\_ionex.h, [1813](#), [1814](#)
  - MIN\_EL, [1814](#)
  - MIN\_HGT, [1814](#)
  - VAR\_NOTEC, [1814](#)
- rtklib\_lambda.h, [1815](#), [1817](#)
  - LOOPMAX, [1816](#)
  - ROUND\_LAMBDA, [1816](#)
  - SGN\_LAMBDA, [1816](#)
  - SWAP\_LAMBDA, [1816](#)
- rtklib\_pntpos.h, [1817](#), [1820](#)
  - ERR\_ION, [1819](#)
  - ERR\_TROP, [1819](#)
  - MAXITR, [1819](#)
  - NX, [1820](#)
  - pntpos, [1819](#)
- rtklib\_ppp.h, [1822](#), [1825](#)
  - CONST\_AMB, [1824](#)
  - LOG\_PI, [1824](#)
  - MIN\_ARC\_GAP, [1824](#)
  - MIN\_PPP, [1823](#)
  - ROUND\_PPP, [1823](#)
- SQRT2, [1824](#)
- SWAP\_D, [1824](#)
- SWAP\_I, [1824](#)
- THRES\_RES, [1824](#)
- VAR\_BIAS, [1824](#)
- VAR\_CLK, [1825](#)
- VAR\_GRA\_PPP, [1825](#)
- VAR\_IONO\_OFF, [1825](#)
- VAR\_POS\_PPP, [1825](#)
- VAR\_ZTD, [1825](#)
- rtklib\_preceph.h, [1827](#), [1829](#)
  - EXTERR\_CLK, [1828](#)
  - EXTERR\_EPH, [1828](#)
  - MAXDTE, [1828](#)
  - NMAX, [1828](#)
- Rtklib\_Pvt, [1359](#)
  - clear\_ephemeris, [1363](#)
  - connect, [1363](#)
  - disconnect, [1363](#)
  - get\_galileo\_almanac, [1363](#)
  - get\_galileo\_ephemeris, [1363](#)
  - get\_gps\_almanac, [1363](#)
  - get\_gps\_ephemeris, [1364](#)
  - get\_latest\_PVT, [1364](#)
  - get\_left\_block, [1364](#)
  - get\_right\_block, [1364](#)
  - implementation, [1364](#)
  - item\_size, [1364](#)
  - reset, [1364](#)
  - role, [1364](#)
- rtklib\_pvt.h, [1882](#)
- rtklib\_pvt\_gs, [1365](#)
  - ~rtklib\_pvt\_gs, [1367](#)
  - clear\_ephemeris, [1367](#)
  - get\_beidou\_dnav\_almanac\_map, [1367](#)
  - get\_beidou\_dnav\_ephemeris\_map, [1367](#)
  - get\_galileo\_almanac\_map, [1367](#)
  - get\_galileo\_ephemeris\_map, [1367](#)
  - get\_gps\_almanac\_map, [1367](#)
  - get\_gps\_ephemeris\_map, [1367](#)
  - get\_latest\_PVT, [1367](#)
  - work, [1368](#)
- rtklib\_pvt\_gs.h, [1885](#), [1887](#)
- rtklib\_pvt\_gs\_sptr
  - pvt\_gr\_blocks, [170](#)
- rtklib\_rtcn.h, [1830](#), [1831](#)
  - RTCM2PREAMB, [1831](#)
  - RTCM3PREAMB, [1831](#)
- rtklib\_rtcn2.h, [1832](#), [1833](#)
- rtklib\_rtcn3.h, [1834](#), [1837](#)
  - CODES\_BDS, [1836](#)
  - CODES\_GAL, [1836](#)
  - CODES\_GLO, [1836](#)
  - CODES\_GPS, [1836](#)
  - CODES\_QZS, [1837](#)
  - CODES\_SBS, [1837](#)
  - PRUNIT\_GLO, [1837](#)
  - PRUNIT\_GPS, [1837](#)

- RANGE\_MS, [1837](#)
- SSRUDINT, [1837](#)
- rtklib\_rtkcmn.h, [1840](#), [1845](#)
  - Rx, [1844](#)
  - Ry, [1844](#)
  - Rz, [1845](#)
- rtklib\_rtkpos.h, [1848](#), [1850](#)
- rtklib\_rtksvr.h, [1852](#), [1854](#)
  - PRCOPT\_DEFAULT, [1854](#)
  - SOLOPT\_DEFAULT, [1854](#)
- rtklib\_sbass.h, [1856](#), [1858](#)
  - IGPBAND1, [1857](#)
  - IGPBAND2, [1857](#)
  - WEEKOFFSET, [1857](#)
- rtklib\_solution.h, [1859](#), [1861](#)
- Rtklib\_Solver, [1368](#)
  - beidou\_dnav\_almanac\_map, [1372](#)
  - beidou\_dnav\_ephemeris\_map, [1372](#)
  - beidou\_dnav\_iono, [1372](#)
  - beidou\_dnav\_utc\_model, [1372](#)
  - galileo\_almanac\_map, [1372](#)
  - galileo\_ephemeris\_map, [1372](#)
  - galileo\_iono, [1373](#)
  - galileo\_utc\_model, [1373](#)
  - get\_gdop, [1372](#)
  - get\_hdop, [1372](#)
  - get\_pdop, [1372](#)
  - get\_vdop, [1372](#)
  - glonass\_gnav\_almanac, [1373](#)
  - glonass\_gnav\_ephemeris\_map, [1373](#)
  - glonass\_gnav\_utc\_model, [1373](#)
  - gps\_almanac\_map, [1373](#)
  - gps\_cnav\_ephemeris\_map, [1373](#)
  - gps\_cnav\_iono, [1373](#)
  - gps\_cnav\_utc\_model, [1373](#)
  - gps\_ephemeris\_map, [1373](#)
  - gps\_iono, [1373](#)
  - gps\_utc\_model, [1374](#)
  - pvt\_sol, [1374](#)
  - pvt\_ssar, [1374](#)
- rtklib\_solver.h, [1956](#), [1957](#)
- rtklib\_stream.h, [1863](#), [1868](#)
  - FTP\_CMD, [1866](#)
  - FTP\_TIMEOUT, [1866](#)
  - MAXCLI, [1866](#)
  - MAXSTATMSG, [1866](#)
  - NTRIP\_AGENT, [1866](#)
  - NTRIP\_CLI\_PORT, [1866](#)
  - NTRIP\_MAXRSP, [1866](#)
  - NTRIP\_MAXSTR, [1866](#)
  - NTRIP\_RSP\_ERROR, [1866](#)
  - NTRIP\_RSP\_HTTP, [1867](#)
  - NTRIP\_RSP\_OK\_CLI, [1867](#)
  - NTRIP\_RSP\_OK\_SVR, [1867](#)
  - NTRIP\_RSP\_SRCTBL, [1867](#)
  - NTRIP\_RSP\_TBLEN, [1867](#)
  - NTRIP\_SVR\_PORT, [1867](#)
  - SERIBUFFSIZE, [1867](#)
  - TIMETAGH\_LEN, [1867](#)
  - TINTACT, [1867](#)
  - VER\_RTKLIB, [1867](#)
- rtklib\_tides.h, [1870](#), [1871](#)
  - GME, [1871](#)
  - GMM, [1871](#)
  - GMS, [1871](#)
- rtksvr\_t, [1375](#)
  - buff, [1376](#)
  - buffsize, [1376](#)
  - cputime, [1376](#)
  - cycle, [1376](#)
  - files, [1376](#)
  - format, [1376](#)
  - ftime, [1376](#)
  - lock, [1377](#)
  - moni, [1377](#)
  - nav, [1377](#)
  - navsel, [1377](#)
  - nb, [1377](#)
  - nmeacycle, [1377](#)
  - nmeapos, [1377](#)
  - nmeareq, [1377](#)
  - nmsg, [1377](#)
  - npb, [1377](#)
  - nsb, [1377](#)
  - nsbs, [1377](#)
  - nsol, [1378](#)
  - obs, [1378](#)
  - pbuf, [1378](#)
  - prcout, [1378](#)
  - raw, [1378](#)
  - rtcm, [1378](#)
  - rtk, [1378](#)
  - sbsmsg, [1378](#)
  - sbuf, [1378](#)
  - solbuf, [1378](#)
  - solo, [1378](#)
  - state, [1378](#)
  - stream, [1379](#)
  - thread, [1379](#)
  - tick, [1379](#)
- RTL\_TCP\_COMMAND
  - signal\_source\_libs, [185](#)
- rtl\_tcp\_command
  - signal\_source\_libs, [185](#)
- rtl\_tcp\_commands.h, [2109](#), [2110](#)
- Rtl\_Tcp\_Dongle\_Info, [1379](#)
  - get\_tuner\_gain\_count, [1380](#)
  - get\_tuner\_type, [1380](#)
- rtl\_tcp\_dongle\_info.h, [2111](#), [2112](#)
- rtl\_tcp\_signal\_source.h, [2027](#)
- rtl\_tcp\_signal\_source\_c, [1380](#)
- rtl\_tcp\_signal\_source\_c.h, [2053](#), [2054](#)
- rtl\_tcp\_signal\_source\_c\_sptr
  - signal\_source\_gr\_blocks, [182](#)
- RtlTcpSignalSource, [1382](#)
  - connect, [1385](#)

- disconnect, [1385](#)
- get\_left\_block, [1385](#)
- get\_right\_block, [1385](#)
- item\_size, [1385](#)
- ru
  - prcopt\_t, [1301](#)
- run
  - ControlThread, [466](#)
- run\_acquisition
  - Fpga\_Acquisition, [578](#)
- run\_server
  - Rtcm, [1349](#)
- running
  - GNSSFlowgraph, [914](#)
- Rx
  - rtklib\_rtkcmn.h, [1844](#)
- RX\_time
  - Gnss\_Synchro, [904](#)
  - Monitor\_Pvt, [1174](#)
- rx\_time
  - GnssTime, [915](#)
- Ry
  - rtklib\_rtkcmn.h, [1844](#)
- Rz
  - rtklib\_rtkcmn.h, [1845](#)
- saddr
  - tcp\_t, [1477](#)
- sample\_counter
  - TrackingCmd, [1498](#)
- sample\_freq\_in
  - direct\_resampler\_conditioner\_cb, [484](#)
  - direct\_resampler\_conditioner\_cc, [486](#)
  - direct\_resampler\_conditioner\_cs, [488](#)
- sample\_freq\_out
  - direct\_resampler\_conditioner\_cb, [484](#)
  - direct\_resampler\_conditioner\_cc, [486](#)
  - direct\_resampler\_conditioner\_cs, [488](#)
- sampled\_ms
  - Acq\_Conf, [327](#)
- samples
  - FileSourceBase, [552](#)
  - MultichannelFileSignalSource, [1184](#)
  - SpirGSS6450FileSignalSource, [1457](#)
- samples\_per\_chip
  - Acq\_Conf, [327](#)
- samples\_per\_code
  - Acq\_Conf, [327](#)
- samples\_per\_ms
  - Acq\_Conf, [327](#)
- samplestamp
  - PpsSamplestamp, [1295](#)
- samplesToSkip
  - FileSourceBase, [553](#)
- sampling\_frequency
  - FileSourceBase, [553](#)
  - MultichannelFileSignalSource, [1184](#)
  - SpirGSS6450FileSignalSource, [1457](#)
- sat
  - alm\_t, [355](#)
  - eph\_t, [523](#)
  - geph\_t, [789](#)
  - half\_cyc\_tag, [1088](#)
  - lexeph\_t, [1149](#)
  - obsd\_t, [1218](#)
  - pcv\_t, [1284](#)
  - sbssat\_t, [1405](#)
  - sbssatp\_t, [1407](#)
  - seph\_t, [1410](#)
  - solstat\_t, [1446](#)
  - stec\_t, [1466](#)
- satclass
  - tled\_t, [1489](#)
- satClkDrift
  - Gnss\_Ephemeris, [876](#)
- satellite\_mask
  - Galileo\_HAS\_data, [634](#)
- satellite\_submask
  - Galileo\_HAS\_data, [634](#)
- satelliteBlock
  - Beidou\_Dnav\_Ephemeris, [383](#)
  - Gps\_Ephemeris, [943](#)
- satellitePosition
  - Gnss\_Ephemeris, [873](#)
- satellitePosVelComputation
  - Gnss\_Almanac, [863](#)
- sateph
  - prcopt\_t, [1301](#)
- satno
  - tled\_t, [1489](#)
- satpos\_X
  - Gnss\_Ephemeris, [876](#)
- satpos\_Y
  - Gnss\_Ephemeris, [876](#)
- satpos\_Z
  - Gnss\_Ephemeris, [877](#)
- sats
  - msm\_h\_t, [1178](#)
- satvel\_X
  - Gnss\_Ephemeris, [877](#)
- satvel\_Y
  - Gnss\_Ephemeris, [877](#)
- satvel\_Z
  - Gnss\_Ephemeris, [877](#)
- save\_cnav\_ephemeris\_map\_xml
  - Gnss\_Sdr\_Supl\_Client, [888](#)
- save\_cnav\_utc\_xml
  - Gnss\_Sdr\_Supl\_Client, [888](#)
- save\_ephemeris\_map\_xml
  - Gnss\_Sdr\_Supl\_Client, [889](#)
- save\_gal\_almanac\_xml
  - Gnss\_Sdr\_Supl\_Client, [889](#)
- save\_gal\_ephemeris\_map\_xml
  - Gnss\_Sdr\_Supl\_Client, [889](#)
- save\_gal\_iono\_xml
  - Gnss\_Sdr\_Supl\_Client, [889](#)
- save\_gal\_utc\_xml

- Gnss\_Sdr\_Supl\_Client, [889](#)
- save\_glo\_utc\_xml
  - Gnss\_Sdr\_Supl\_Client, [889](#)
- save\_gnav\_ephemeris\_map\_xml
  - Gnss\_Sdr\_Supl\_Client, [889](#)
- save\_gps\_almanac\_xml
  - Gnss\_Sdr\_Supl\_Client, [889](#)
- save\_iono\_xml
  - Gnss\_Sdr\_Supl\_Client, [890](#)
- save\_ref\_location\_xml
  - Gnss\_Sdr\_Supl\_Client, [890](#)
- save\_ref\_time\_xml
  - Gnss\_Sdr\_Supl\_Client, [890](#)
- save\_utc\_xml
  - Gnss\_Sdr\_Supl\_Client, [890](#)
- Sbas\_Ephemeris, [1386](#)
  - b\_sv\_do\_not\_use, [1387](#)
  - d\_acc, [1387](#)
  - d\_af0, [1387](#)
  - d\_af1, [1387](#)
  - d\_pos, [1387](#)
  - d\_tof, [1387](#)
  - d\_vel, [1387](#)
  - i\_prn, [1387](#)
  - i\_sv\_ura, [1388](#)
  - i\_t0, [1388](#)
- sbas\_ephemeris.h, [2723](#), [2724](#)
- sbas\_l1\_telemetry\_decoder.h, [2133](#)
- sbas\_l1\_telemetry\_decoder\_gs, [1388](#)
  - general\_work, [1389](#)
  - reset, [1390](#)
  - set\_channel, [1390](#)
  - set\_satellite, [1390](#)
- sbas\_l1\_telemetry\_decoder\_gs.h, [2157](#), [2158](#)
- sbas\_l1\_telemetry\_decoder\_gs\_sptr
  - telemetry\_decoder\_gr\_blocks, [189](#)
- sbascorr
  - prcopt\_t, [1301](#)
- SbasL1TelemetryDecoder, [1390](#)
  - connect, [1393](#)
  - disconnect, [1393](#)
  - get\_left\_block, [1393](#)
  - get\_right\_block, [1393](#)
  - implementation, [1393](#)
  - item\_size, [1393](#)
  - reset, [1393](#)
  - role, [1394](#)
  - set\_channel, [1394](#)
  - set\_satellite, [1394](#)
- sbassatsel
  - prcopt\_t, [1301](#)
- sbs\_t, [1394](#)
  - msgs, [1395](#)
  - n, [1395](#)
  - nmax, [1395](#)
- sbsfcorr\_t, [1396](#)
  - ai, [1396](#)
  - dt, [1396](#)
  - iodf, [1396](#)
  - prc, [1397](#)
  - rrc, [1397](#)
  - t0, [1397](#)
  - udre, [1397](#)
- sbsigp\_t, [1397](#)
  - delay, [1398](#)
  - give, [1398](#)
  - lat, [1398](#)
  - lon, [1398](#)
  - t0, [1398](#)
- sbsigpband\_t, [1399](#)
  - bite, [1399](#)
  - bits, [1399](#)
  - x, [1399](#)
  - y, [1399](#)
- sbsion
  - nav\_t, [1195](#)
- sbsion\_t, [1400](#)
  - igp, [1400](#)
  - iodi, [1400](#)
  - nigp, [1401](#)
- sbslcorr\_t, [1401](#)
  - daf0, [1401](#)
  - daf1, [1402](#)
  - dpos, [1402](#)
  - dvel, [1402](#)
  - iode, [1402](#)
  - t0, [1402](#)
- sbsmsg
  - raw\_t, [1326](#)
  - rtksvr\_t, [1378](#)
- sbsmsg\_t, [1402](#)
  - msg, [1403](#)
  - prn, [1403](#)
  - tow, [1403](#)
  - week, [1403](#)
- sbssat
  - nav\_t, [1195](#)
- sbssat\_t, [1404](#)
  - iodp, [1405](#)
  - nsat, [1405](#)
  - sat, [1405](#)
  - tlat, [1405](#)
- sbssatp\_t, [1406](#)
  - fcorr, [1406](#)
  - lcorr, [1406](#)
  - sat, [1407](#)
- sbuf
  - rtksvr\_t, [1378](#)
- SC2RAD
  - core\_system\_parameters, [312](#)
- sclkstab
  - prcopt\_t, [1301](#)
- sdr\_gnss\_packet\_t, [1407](#)
  - doppler, [1408](#)
  - galileo\_satellites, [1408](#)
  - gps\_satellites, [1408](#)



- height, 1408
- latitude, 1408
- longitude, 1408
- microseconds, 1408
- nsvfix, 1408
- prn, 1408
- reserved, 1408
- snr, 1409
- status, 1409
- velocity, 1409
- sec
  - gtime\_t, 1086
- seconds
  - Agnss\_Ref\_Time, 351
- send\_message
  - Rtcm, 1349
- send\_telemetry\_msg
  - GNSSFlowgraph, 914
- sep
  - solo\_t, 1443
- seph
  - nav\_t, 1195
- seph\_t, 1409
  - acc, 1410
  - af0, 1410
  - af1, 1410
  - pos, 1410
  - sat, 1410
  - sva, 1410
  - svh, 1410
  - t0, 1410
  - tof, 1410
  - vel, 1411
- seqno
  - rtcm\_t, 1356
- sequence1
  - Mack\_lookup, 1158
- sequence2
  - Mack\_lookup, 1158
- Serdes\_Galileo\_Eph, 1411
  - ~Serdes\_Galileo\_Eph, 1412
  - createProtobuffer, 1412
  - operator=, 1412
  - readProtobuffer, 1412
  - Serdes\_Galileo\_Eph, 1412
- serdes\_galileo\_eph.h, 1959, 1960
- Serdes\_Gnss\_Synchro, 1413
  - ~Serdes\_Gnss\_Synchro, 1414
  - createProtobuffer, 1414
  - operator=, 1414
  - readProtobuffer, 1414
  - Serdes\_Gnss\_Synchro, 1414
- serdes\_gnss\_synchro.h, 2375, 2376
- Serdes\_Gps\_Eph, 1415
  - ~Serdes\_Gps\_Eph, 1416
  - createProtobuffer, 1416
  - operator=, 1416
  - readProtobuffer, 1416
  - Serdes\_Gps\_Eph, 1416
  - serdes\_gps\_eph.h, 1962, 1963
  - Serdes\_Monitor\_Pvt, 1417
    - ~Serdes\_Monitor\_Pvt, 1418
    - createProtobuffer, 1418
    - operator=, 1418
    - readProtobuffer, 1418
    - Serdes\_Monitor\_Pvt, 1418
  - serdes\_monitor\_pvt.h, 1966, 1967
  - Serdes\_Nav\_Message, 1419
    - ~Serdes\_Nav\_Message, 1419
    - createProtobuffer, 1420
    - operator=, 1420
    - readProtobuffer, 1420
    - Serdes\_Nav\_Message, 1419, 1420
  - serdes\_nav\_message.h, 2365, 2366
  - serial\_t, 1421
    - dev, 1421
    - error, 1421
  - serialize
    - Agnss\_Ref\_Location, 349
    - Agnss\_Ref\_Time, 351
    - Beidou\_Dnav\_Almanac, 377
    - Beidou\_Dnav\_Ephemeris, 382
    - Beidou\_Dnav\_Utc\_Model, 392
    - Galileo\_Almanac, 599
    - Galileo\_Ephemeris, 621
    - Galileo\_Iono, 642
    - Galileo\_Utc\_Model, 658
    - Glonass\_Gnav\_Almanac, 792
    - Glonass\_Gnav\_Ephemeris, 798
    - Glonass\_Gnav\_Utc\_Model, 810
    - Gnss\_Synchro, 901
    - Gps\_Almanac, 922
    - Gps\_CNAV\_Ephemeris, 927
    - Gps\_Ephemeris, 941
    - Gps\_Iono, 946
    - Gps\_Utc\_Model, 963
    - Monitor\_Pvt, 1172
- SERIBUFFSIZE
  - algorithms\_libs\_rtklib, 161
  - rtklib\_stream.h, 1867
- server\_name
  - Gnss\_Sdr\_Supl\_Client, 891
- server\_port
  - Gnss\_Sdr\_Supl\_Client, 891
- set\_active
  - galileo\_e5a\_noncoherentIQ\_acquisition\_caf\_cc, 611
  - galileo\_pcps\_8ms\_acquisition\_cc, 648
  - pcps\_acquisition, 1244
  - pcps\_acquisition\_fine\_doppler\_cc, 1251
  - pcps\_acquisition\_fpga, 1254
  - pcps\_assisted\_acquisition\_cc, 1259
  - pcps\_cccwsr\_acquisition\_cc, 1264
  - pcps\_openc1\_acquisition\_cc, 1269
  - pcps\_quicksync\_acquisition\_cc, 1275
  - pcps\_tong\_acquisition\_cc, 1281



- set\_alpha
  - Exponential\_Smoother, 528
- set\_block\_exp
  - Fpga\_Acquisition, 578
- set\_channel
  - AcquisitionInterface, 335
  - beidou\_b1i\_telemetry\_decoder\_gs, 371
  - beidou\_b3i\_telemetry\_decoder\_gs, 373
  - BeidouB1iDIIPIITracking, 397
  - BeidouB1iPcpsAcquisition, 401
  - BeidouB1iTelemetryDecoder, 407
  - BeidouB3iDIIPIITracking, 411
  - BeidouB3iPcpsAcquisition, 415
  - BeidouB3iTelemetryDecoder, 421
  - dll\_pll\_veml\_tracking\_fpga, 508
  - galileo\_e5a\_noncoherentIQ\_acquisition\_caf\_cc, 612
  - galileo\_pcps\_8ms\_acquisition\_cc, 649
  - galileo\_telemetry\_decoder\_gs, 655
  - GalileoE1BTelemetryDecoder, 664
  - GalileoE1DIIPIIVemlTracking, 668
  - GalileoE1DIIPIIVemlTrackingFpga, 672
  - GalileoE1Pcps8msAmbiguousAcquisition, 677
  - GalileoE1PcpsAmbiguousAcquisition, 683
  - GalileoE1PcpsAmbiguousAcquisitionFpga, 690
  - GalileoE1PcpsCccwsrAmbiguousAcquisition, 695
  - GalileoE1PcpsQuickSyncAmbiguousAcquisition, 701
  - GalileoE1PcpsTongAmbiguousAcquisition, 707
  - GalileoE1TcpConnectorTracking, 713
  - GalileoE5aDIIPIITracking, 717
  - GalileoE5aDIIPIITrackingFpga, 721
  - GalileoE5aNoncoherentIQAcquisitionCaf, 726
  - GalileoE5aPcpsAcquisition, 732
  - GalileoE5aPcpsAcquisitionFpga, 739
  - GalileoE5aTelemetryDecoder, 745
  - GalileoE5bDIIPIITracking, 749
  - GalileoE5bPcpsAcquisition, 754
  - GalileoE5bPcpsAcquisitionFpga, 761
  - GalileoE5bTelemetryDecoder, 767
  - GalileoE6DIIPIITracking, 771
  - GalileoE6PcpsAcquisition, 775
  - GalileoE6TelemetryDecoder, 781
  - glonass\_l1\_ca\_telemetry\_decoder\_gs, 818
  - glonass\_l2\_ca\_telemetry\_decoder\_gs, 825
  - GlonassL1CaDIIPIICAidTracking, 829
  - GlonassL1CaDIIPIITracking, 833
  - GlonassL1CaPcpsAcquisition, 837
  - GlonassL1CaTelemetryDecoder, 842
  - GlonassL2CaDIIPIICAidTracking, 846
  - GlonassL2CaDIIPIITracking, 850
  - GlonassL2CaPcpsAcquisition, 854
  - GlonassL2CaTelemetryDecoder, 859
  - gps\_l1\_ca\_telemetry\_decoder\_gs, 954
  - gps\_l2c\_telemetry\_decoder\_gs, 955
  - gps\_l5\_telemetry\_decoder\_gs, 957
  - Gps\_Navigation\_Message, 960
  - GpsL1CaDIIPIITracking, 968
  - GpsL1CaDIIPIITrackingFpga, 972
  - GpsL1CaDIIPIITrackingGPU, 977
  - GpsL1CaGaussianTracking, 981
  - GpsL1CaKfTracking, 985
  - GpsL1CaPcpsAcquisition, 989
  - GpsL1CaPcpsAcquisitionFineDoppler, 995
  - GpsL1CaPcpsAcquisitionFpga, 1001
  - GpsL1CaPcpsAssistedAcquisition, 1007
  - GpsL1CaPcpsOpenCIAcquisition, 1012
  - GpsL1CaPcpsQuickSyncAcquisition, 1018
  - GpsL1CaPcpsTongAcquisition, 1023
  - GpsL1CaTcpConnectorTracking, 1028
  - GpsL1CaTelemetryDecoder, 1032
  - GpsL2CTelemetryDecoder, 1036
  - GpsL2MDIIPIITracking, 1040
  - GpsL2MDIIPIITrackingFpga, 1044
  - GpsL2MPcpsAcquisition, 1048
  - GpsL2MPcpsAcquisitionFpga, 1054
  - GpsL5DIIPIITracking, 1059
  - GpsL5DIIPIITrackingFpga, 1063
  - GpsL5iPcpsAcquisition, 1068
  - GpsL5iPcpsAcquisitionFpga, 1075
  - GpsL5TelemetryDecoder, 1080
  - pcps\_acquisition, 1244
  - pcps\_acquisition\_fine\_doppler\_cc, 1251
  - pcps\_acquisition\_fpga, 1254
  - pcps\_assisted\_acquisition\_cc, 1260
  - pcps\_cccwsr\_acquisition\_cc, 1265
  - pcps\_opencl\_acquisition\_cc, 1270
  - pcps\_quicksync\_acquisition\_cc, 1276
  - pcps\_tong\_acquisition\_cc, 1282
  - sbas\_l1\_telemetry\_decoder\_gs, 1390
  - SbasL1TelemetryDecoder, 1394
  - TIm\_CRC\_Stats, 1492
  - TrackingInterface, 1500
- set\_channel\_fsm
  - AcquisitionInterface, 335
  - BeidouB1iPcpsAcquisition, 402
  - BeidouB3iPcpsAcquisition, 415
  - galileo\_e5a\_noncoherentIQ\_acquisition\_caf\_cc, 612
  - galileo\_pcps\_8ms\_acquisition\_cc, 649
  - GalileoE1Pcps8msAmbiguousAcquisition, 678
  - GalileoE1PcpsAmbiguousAcquisition, 684
  - GalileoE1PcpsAmbiguousAcquisitionFpga, 690
  - GalileoE1PcpsCccwsrAmbiguousAcquisition, 696
  - GalileoE1PcpsQuickSyncAmbiguousAcquisition, 702
  - GalileoE1PcpsTongAmbiguousAcquisition, 708
  - GalileoE5aNoncoherentIQAcquisitionCaf, 727
  - GalileoE5aPcpsAcquisition, 732
  - GalileoE5aPcpsAcquisitionFpga, 739
  - GalileoE5bPcpsAcquisition, 754
  - GalileoE5bPcpsAcquisitionFpga, 761
  - GalileoE6PcpsAcquisition, 776
  - GlonassL1CaPcpsAcquisition, 837
  - GlonassL2CaPcpsAcquisition, 854
  - GpsL1CaPcpsAcquisition, 990

- GpsL1CaPcpsAcquisitionFineDoppler, 995
- GpsL1CaPcpsAcquisitionFpga, 1001
- GpsL1CaPcpsAssistedAcquisition, 1007
- GpsL1CaPcpsOpenCIAcquisition, 1013
- GpsL1CaPcpsQuickSyncAcquisition, 1018
- GpsL1CaPcpsTongAcquisition, 1023
- GpsL2MPcpsAcquisition, 1049
- GpsL2MPcpsAcquisitionFpga, 1054
- GpsL5iPcpsAcquisition, 1069
- GpsL5iPcpsAcquisitionFpga, 1075
- pcps\_acquisition, 1245
- pcps\_acquisition\_fine\_doppler\_cc, 1251
- pcps\_acquisition\_fpga, 1255
- pcps\_assisted\_acquisition\_cc, 1260
- pcps\_cccwsr\_acquisition\_cc, 1265
- pcps\_openc1\_acquisition\_cc, 1270
- pcps\_quicksync\_acquisition\_cc, 1276
- pcps\_tong\_acquisition\_cc, 1282
- set\_clock\_drift\_ppm
  - Pvt\_Solution, 1319
- set\_configuration
  - GNSSFlowgraph, 914
- set\_control\_queue
  - ControlThread, 467
- set\_course\_over\_ground
  - Pvt\_Solution, 1319
- set\_DLL\_BW
  - Tracking\_2nd\_DLL\_filter, 1494
- set\_doppler\_center
  - AcquisitionInterface, 335
  - GalileoE1PcpsAmbiguousAcquisition, 684
  - GalileoE1PcpsAmbiguousAcquisitionFpga, 690
  - GalileoE5aPcpsAcquisition, 733
  - GalileoE5aPcpsAcquisitionFpga, 740
  - GalileoE5bPcpsAcquisition, 754
  - GalileoE5bPcpsAcquisitionFpga, 762
  - GalileoE6PcpsAcquisition, 776
  - GpsL1CaPcpsAcquisition, 990
  - GpsL1CaPcpsAcquisitionFpga, 1002
  - GpsL2MPcpsAcquisition, 1049
  - GpsL5iPcpsAcquisition, 1069
  - GpsL5iPcpsAcquisitionFpga, 1075
  - pcps\_acquisition, 1245
  - pcps\_acquisition\_fpga, 1255
- set\_doppler\_max
  - AcquisitionInterface, 335
  - BeidouB1iPcpsAcquisition, 402
  - BeidouB3iPcpsAcquisition, 416
  - Fpga\_Acquisition, 578
  - galileo\_e5a\_noncoherentIQ\_acquisition\_caf\_cc, 612
  - galileo\_pcps\_8ms\_acquisition\_cc, 649
  - GalileoE1Pcps8msAmbiguousAcquisition, 678
  - GalileoE1PcpsAmbiguousAcquisition, 684
  - GalileoE1PcpsAmbiguousAcquisitionFpga, 691
  - GalileoE1PcpsCccwsrAmbiguousAcquisition, 696
  - GalileoE1PcpsQuickSyncAmbiguousAcquisition, 702
- GalileoE1PcpsTongAmbiguousAcquisition, 708
- GalileoE5aNoncoherentIQAcquisitionCaf, 727
- GalileoE5aPcpsAcquisition, 733
- GalileoE5aPcpsAcquisitionFpga, 740
- GalileoE5bPcpsAcquisition, 755
- GalileoE5bPcpsAcquisitionFpga, 762
- GalileoE6PcpsAcquisition, 776
- GlonassL1CaPcpsAcquisition, 838
- GlonassL2CaPcpsAcquisition, 855
- GpsL1CaPcpsAcquisition, 990
- GpsL1CaPcpsAcquisitionFineDoppler, 996
- GpsL1CaPcpsAcquisitionFpga, 1002
- GpsL1CaPcpsAssistedAcquisition, 1008
- GpsL1CaPcpsOpenCIAcquisition, 1013
- GpsL1CaPcpsQuickSyncAcquisition, 1019
- GpsL1CaPcpsTongAcquisition, 1024
- GpsL2MPcpsAcquisition, 1049
- GpsL2MPcpsAcquisitionFpga, 1055
- GpsL5iPcpsAcquisition, 1069
- GpsL5iPcpsAcquisitionFpga, 1076
- pcps\_acquisition, 1245
- pcps\_acquisition\_fine\_doppler\_cc, 1251
- pcps\_acquisition\_fpga, 1255
- pcps\_assisted\_acquisition\_cc, 1260
- pcps\_cccwsr\_acquisition\_cc, 1265
- pcps\_openc1\_acquisition\_cc, 1270
- pcps\_quicksync\_acquisition\_cc, 1276
- pcps\_tong\_acquisition\_cc, 1282
- set\_doppler\_step
  - AcquisitionInterface, 335
  - BeidouB1iPcpsAcquisition, 402
  - BeidouB3iPcpsAcquisition, 416
  - Fpga\_Acquisition, 579
  - galileo\_e5a\_noncoherentIQ\_acquisition\_caf\_cc, 612
  - galileo\_pcps\_8ms\_acquisition\_cc, 649
  - GalileoE1Pcps8msAmbiguousAcquisition, 678
  - GalileoE1PcpsAmbiguousAcquisition, 684
  - GalileoE1PcpsAmbiguousAcquisitionFpga, 691
  - GalileoE1PcpsCccwsrAmbiguousAcquisition, 696
  - GalileoE1PcpsQuickSyncAmbiguousAcquisition, 702
  - GalileoE1PcpsTongAmbiguousAcquisition, 708
  - GalileoE5aNoncoherentIQAcquisitionCaf, 727
  - GalileoE5aPcpsAcquisition, 733
  - GalileoE5aPcpsAcquisitionFpga, 740
  - GalileoE5bPcpsAcquisition, 755
  - GalileoE5bPcpsAcquisitionFpga, 762
  - GalileoE6PcpsAcquisition, 776
  - GlonassL1CaPcpsAcquisition, 838
  - GlonassL2CaPcpsAcquisition, 855
  - GpsL1CaPcpsAcquisition, 990
  - GpsL1CaPcpsAcquisitionFineDoppler, 996
  - GpsL1CaPcpsAcquisitionFpga, 1002
  - GpsL1CaPcpsAssistedAcquisition, 1008
  - GpsL1CaPcpsOpenCIAcquisition, 1013
  - GpsL1CaPcpsQuickSyncAcquisition, 1019
  - GpsL1CaPcpsTongAcquisition, 1024

- GpsL2MPcpsAcquisition, 1049
- GpsL2MPcpsAcquisitionFpga, 1055
- GpsL5iPcpsAcquisition, 1069
- GpsL5iPcpsAcquisitionFpga, 1076
- pcps\_acquisition, 1245
- pcps\_acquisition\_fine\_doppler\_cc, 1251
- pcps\_acquisition\_fpga, 1255
- pcps\_assisted\_acquisition\_cc, 1260
- pcps\_cccwsr\_acquisition\_cc, 1265
- pcps\_openc1\_acquisition\_cc, 1270
- pcps\_quicksync\_acquisition\_cc, 1276
- pcps\_tong\_acquisition\_cc, 1282
- set\_doppler\_sweep
  - Fpga\_Acquisition, 579
- set\_ephemeris\_data
  - OSNMA\_NavData, 1235
- set\_flag\_ephemeris\_str\_1
  - Glonass\_Gnav\_Navigation\_Message, 807
- set\_flag\_ephemeris\_str\_2
  - Glonass\_Gnav\_Navigation\_Message, 808
- set\_flag\_ephemeris\_str\_3
  - Glonass\_Gnav\_Navigation\_Message, 808
- set\_flag\_ephemeris\_str\_4
  - Glonass\_Gnav\_Navigation\_Message, 808
- set\_flag\_new\_SOW\_available
  - Beidou\_Dnav\_Navigation\_Message, 390
- set\_flag\_TOW\_new
  - Glonass\_Gnav\_Navigation\_Message, 808
- set\_flag\_TOW\_set
  - Galileo\_Fnav\_Message, 625
  - Galileo\_Inav\_Message, 640
- set\_flag\_update\_slot\_number
  - Glonass\_Gnav\_Navigation\_Message, 808
- set\_gnss\_synchro
  - AcquisitionInterface, 336
  - BeidouB1iDIIPIITracking, 397
  - BeidouB1iPcpsAcquisition, 402
  - BeidouB3iDIIPIITracking, 411
  - BeidouB3iPcpsAcquisition, 416
  - dll\_pll\_veml\_tracking\_fpga, 508
  - galileo\_e5a\_noncoherentIQ\_acquisition\_caf\_cc, 612
  - galileo\_pcps\_8ms\_acquisition\_cc, 649
  - GalileoE1DIIPIIVemlTracking, 668
  - GalileoE1DIIPIIVemlTrackingFpga, 672
  - GalileoE1Pcps8msAmbiguousAcquisition, 678
  - GalileoE1PcpsAmbiguousAcquisition, 684
  - GalileoE1PcpsAmbiguousAcquisitionFpga, 691
  - GalileoE1PcpsCccwsrAmbiguousAcquisition, 696
  - GalileoE1PcpsQuickSyncAmbiguousAcquisition, 702
  - GalileoE1PcpsTongAmbiguousAcquisition, 708
  - GalileoE1TcpConnectorTracking, 713
  - GalileoE5aDIIPIITracking, 717
  - GalileoE5aDIIPIITrackingFpga, 721
  - GalileoE5aNoncoherentIQAcquisitionCaf, 727
  - GalileoE5aPcpsAcquisition, 733
  - GalileoE5aPcpsAcquisitionFpga, 740
  - GalileoE5bDIIPIITracking, 749
  - GalileoE5bPcpsAcquisition, 755
  - GalileoE5bPcpsAcquisitionFpga, 762
  - GalileoE6DIIPIITracking, 771
  - GalileoE6PcpsAcquisition, 776
  - GlonassL1CaDIIPIICAidTracking, 829
  - GlonassL1CaDIIPIITracking, 833
  - GlonassL1CaPcpsAcquisition, 838
  - GlonassL2CaDIIPIICAidTracking, 846
  - GlonassL2CaDIIPIITracking, 850
  - GlonassL2CaPcpsAcquisition, 855
  - GpsL1CaDIIPIITracking, 968
  - GpsL1CaDIIPIITrackingFpga, 972
  - GpsL1CaDIIPIITrackingGPU, 977
  - GpsL1CaGaussianTracking, 981
  - GpsL1CaKfTracking, 985
  - GpsL1CaPcpsAcquisition, 990
  - GpsL1CaPcpsAcquisitionFineDoppler, 996
  - GpsL1CaPcpsAcquisitionFpga, 1002
  - GpsL1CaPcpsAssistedAcquisition, 1008
  - GpsL1CaPcpsOpenCIAcquisition, 1013
  - GpsL1CaPcpsQuickSyncAcquisition, 1019
  - GpsL1CaPcpsTongAcquisition, 1024
  - GpsL1CaTcpConnectorTracking, 1028
  - GpsL2MDIIPIITracking, 1040
  - GpsL2MDIIPIITrackingFpga, 1044
  - GpsL2MPcpsAcquisition, 1049
  - GpsL2MPcpsAcquisitionFpga, 1055
  - GpsL5DIIPIITracking, 1059
  - GpsL5DIIPIITrackingFpga, 1063
  - GpsL5iPcpsAcquisition, 1069
  - GpsL5iPcpsAcquisitionFpga, 1076
  - pcps\_acquisition, 1245
  - pcps\_acquisition\_fine\_doppler\_cc, 1251
  - pcps\_acquisition\_fpga, 1255
  - pcps\_assisted\_acquisition\_cc, 1260
  - pcps\_cccwsr\_acquisition\_cc, 1265
  - pcps\_openc1\_acquisition\_cc, 1270
  - pcps\_quicksync\_acquisition\_cc, 1276
  - pcps\_tong\_acquisition\_cc, 1282
  - TrackingInterface, 1500
- set\_initial\_sample
  - Fpga\_Multicorrelator\_8sc, 586
- set\_IOD\_nav
  - OSNMA\_NavData, 1235
- set\_last\_received\_TOW
  - OSNMA\_NavData, 1235
- set\_local\_code
  - AcquisitionInterface, 336
  - BeidouB1iPcpsAcquisition, 402
  - BeidouB3iPcpsAcquisition, 416
  - Fpga\_Acquisition, 579
  - galileo\_e5a\_noncoherentIQ\_acquisition\_caf\_cc, 614
  - galileo\_pcps\_8ms\_acquisition\_cc, 649
  - GalileoE1Pcps8msAmbiguousAcquisition, 678
  - GalileoE1PcpsAmbiguousAcquisition, 684
  - GalileoE1PcpsAmbiguousAcquisitionFpga, 691

- GalileoE1PcpsCccwsrAmbiguousAcquisition, [696](#)
- GalileoE1PcpsQuickSyncAmbiguousAcquisition, [702](#)
- GalileoE1PcpsTongAmbiguousAcquisition, [708](#)
- GalileoE5aNoncoherentIQAcquisitionCaf, [727](#)
- GalileoE5aPcpsAcquisition, [733](#)
- GalileoE5aPcpsAcquisitionFpga, [740](#)
- GalileoE5bPcpsAcquisition, [755](#)
- GalileoE5bPcpsAcquisitionFpga, [762](#)
- GalileoE6PcpsAcquisition, [776](#)
- GlionassL1CaPcpsAcquisition, [838](#)
- GlionassL2CaPcpsAcquisition, [855](#)
- GpsL1CaPcpsAcquisition, [990](#)
- GpsL1CaPcpsAcquisitionFineDoppler, [996](#)
- GpsL1CaPcpsAcquisitionFpga, [1002](#)
- GpsL1CaPcpsAssistedAcquisition, [1008](#)
- GpsL1CaPcpsOpenCIAcquisition, [1013](#)
- GpsL1CaPcpsQuickSyncAcquisition, [1019](#)
- GpsL1CaPcpsTongAcquisition, [1024](#)
- GpsL2MPcpsAcquisition, [1049](#)
- GpsL2MPcpsAcquisitionFpga, [1055](#)
- GpsL5iPcpsAcquisition, [1069](#)
- GpsL5iPcpsAcquisitionFpga, [1076](#)
- pcps\_acquisition, [1246](#)
- pcps\_acquisition\_fine\_doppler\_cc, [1252](#)
- pcps\_acquisition\_fpga, [1256](#)
- pcps\_assisted\_acquisition\_cc, [1261](#)
- pcps\_cccwsr\_acquisition\_cc, [1265](#)
- pcps\_openc1\_acquisition\_cc, [1272](#)
- pcps\_quicksync\_acquisition\_cc, [1278](#)
- pcps\_tong\_acquisition\_cc, [1282](#)
- set\_local\_code\_and\_taps
  - Fpga\_Multicorrelator\_8sc, [586](#)
- set\_merkle\_root
  - Gnss\_Crypto, [869](#)
  - osnma\_msg\_receiver, [1233](#)
- set\_num\_valid\_observations
  - Pvt\_Solution, [1319](#)
- set\_output\_vectors
  - Fpga\_Multicorrelator\_8sc, [586](#)
- set\_pdi
  - Tracking\_2nd\_DLL\_filter, [1494](#)
  - Tracking\_2nd\_PLL\_filter, [1495](#)
- set\_PLL\_BW
  - Tracking\_2nd\_PLL\_filter, [1495](#)
- set\_pre\_2009\_file
  - Pvt\_Solution, [1319](#)
  - Rinex\_Printer, [1336](#)
- set\_prn\_d
  - OSNMA\_NavData, [1235](#)
- set\_property
  - FileConfiguration, [543](#)
  - InMemoryConfiguration, [1112](#)
- set\_public\_key
  - Gnss\_Crypto, [869](#)
- set\_public\_key\_type
  - Gnss\_Crypto, [869](#)
- set\_resampler\_latency
  - AcquisitionInterface, [336](#)
  - BeidouB1iPcpsAcquisition, [402](#)
  - BeidouB3iPcpsAcquisition, [416](#)
  - GalileoE1Pcps8msAmbiguousAcquisition, [678](#)
  - GalileoE1PcpsAmbiguousAcquisition, [684](#)
  - GalileoE1PcpsAmbiguousAcquisitionFpga, [691](#)
  - GalileoE1PcpsCccwsrAmbiguousAcquisition, [696](#)
  - GalileoE1PcpsQuickSyncAmbiguousAcquisition, [702](#)
  - GalileoE1PcpsTongAmbiguousAcquisition, [708](#)
  - GalileoE5aNoncoherentIQAcquisitionCaf, [727](#)
  - GalileoE5aPcpsAcquisition, [733](#)
  - GalileoE5aPcpsAcquisitionFpga, [740](#)
  - GalileoE5bPcpsAcquisition, [755](#)
  - GalileoE5bPcpsAcquisitionFpga, [762](#)
  - GalileoE6PcpsAcquisition, [776](#)
  - GlionassL1CaPcpsAcquisition, [838](#)
  - GlionassL2CaPcpsAcquisition, [855](#)
  - GpsL1CaPcpsAcquisition, [990](#)
  - GpsL1CaPcpsAcquisitionFineDoppler, [996](#)
  - GpsL1CaPcpsAcquisitionFpga, [1002](#)
  - GpsL1CaPcpsAssistedAcquisition, [1008](#)
  - GpsL1CaPcpsOpenCIAcquisition, [1013](#)
  - GpsL1CaPcpsQuickSyncAcquisition, [1019](#)
  - GpsL1CaPcpsTongAcquisition, [1024](#)
  - GpsL2MPcpsAcquisition, [1049](#)
  - GpsL2MPcpsAcquisitionFpga, [1055](#)
  - GpsL5iPcpsAcquisition, [1069](#)
  - GpsL5iPcpsAcquisitionFpga, [1076](#)
- set\_rf\_link
  - Glionass\_Gnav\_Navigation\_Message, [808](#)
- set\_rx\_pos
  - Pvt\_Solution, [1319](#)
- set\_rx\_vel
  - Pvt\_Solution, [1319](#)
- set\_samples\_for\_initialization
  - Exponential\_Smoother, [528](#)
- set\_satellite
  - beidou\_b1i\_telemetry\_decoder\_gs, [371](#)
  - beidou\_b3i\_telemetry\_decoder\_gs, [373](#)
  - BeidouB1iTelemetryDecoder, [407](#)
  - BeidouB3iTelemetryDecoder, [421](#)
  - galileo\_telemetry\_decoder\_gs, [655](#)
  - GalileoE1BTelemetryDecoder, [664](#)
  - GalileoE5aTelemetryDecoder, [745](#)
  - GalileoE5bTelemetryDecoder, [767](#)
  - GalileoE6TelemetryDecoder, [781](#)
  - glionass\_l1\_ca\_telemetry\_decoder\_gs, [818](#)
  - glionass\_l2\_ca\_telemetry\_decoder\_gs, [825](#)
  - GlionassL1CaTelemetryDecoder, [842](#)
  - GlionassL2CaTelemetryDecoder, [859](#)
  - gps\_l1\_ca\_telemetry\_decoder\_gs, [954](#)
  - gps\_l2c\_telemetry\_decoder\_gs, [956](#)
  - gps\_l5\_telemetry\_decoder\_gs, [957](#)
  - GpsL1CaTelemetryDecoder, [1032](#)
  - GpsL2CTelemetryDecoder, [1036](#)
  - GpsL5TelemetryDecoder, [1080](#)
  - sbas\_l1\_telemetry\_decoder\_gs, [1390](#)

- SbasL1TelemetryDecoder, 1394
- set\_satellite\_PRN
  - Beidou\_Dnav\_Navigation\_Message, 390
  - Gps\_Navigation\_Message, 960
- set\_secondary\_code\_lengths
  - Fpga\_Multicorrelator\_8sc, 587
- set\_signal
  - Channel, 430
  - ChannelInterface, 439
- set\_signal\_type
  - Beidou\_Dnav\_Navigation\_Message, 390
- set\_single\_doppler\_flag
  - GalileoE5aPcpsAcquisitionFpga, 740
  - GalileoE5bPcpsAcquisitionFpga, 762
- set\_speed\_over\_ground
  - Pvt\_Solution, 1319
- set\_state
  - AcquisitionInterface, 336
  - BeidouB1iPcpsAcquisition, 402
  - BeidouB3iPcpsAcquisition, 416
  - galileo\_e5a\_noncoherentIQ\_acquisition\_caf\_cc, 614
  - galileo\_pcps\_8ms\_acquisition\_cc, 650
  - GalileoE1Pcps8msAmbiguousAcquisition, 678
  - GalileoE1PcpsAmbiguousAcquisition, 684
  - GalileoE1PcpsAmbiguousAcquisitionFpga, 691
  - GalileoE1PcpsCccwsrAmbiguousAcquisition, 696
  - GalileoE1PcpsQuickSyncAmbiguousAcquisition, 702
  - GalileoE1PcpsTongAmbiguousAcquisition, 708
  - GalileoE5aNoncoherentIQAcquisitionCaf, 727
  - GalileoE5aPcpsAcquisition, 733
  - GalileoE5aPcpsAcquisitionFpga, 740
  - GalileoE5bPcpsAcquisition, 755
  - GalileoE5bPcpsAcquisitionFpga, 762
  - GalileoE6PcpsAcquisition, 776
  - GlionassL1CaPcpsAcquisition, 838
  - GlionassL2CaPcpsAcquisition, 855
  - GpsL1CaPcpsAcquisition, 990
  - GpsL1CaPcpsAcquisitionFineDoppler, 996
  - GpsL1CaPcpsAcquisitionFpga, 1002
  - GpsL1CaPcpsAssistedAcquisition, 1008
  - GpsL1CaPcpsOpenCIAcquisition, 1013
  - GpsL1CaPcpsQuickSyncAcquisition, 1019
  - GpsL1CaPcpsTongAcquisition, 1024
  - GpsL2MPcpsAcquisition, 1049
  - GpsL2MPcpsAcquisitionFpga, 1055
  - GpsL5iPcpsAcquisition, 1069
  - GpsL5iPcpsAcquisitionFpga, 1076
  - pcps\_acquisition, 1246
  - pcps\_acquisition\_fine\_doppler\_cc, 1252
  - pcps\_acquisition\_fpga, 1256
  - pcps\_assisted\_acquisition\_cc, 1261
  - pcps\_cccwsr\_acquisition\_cc, 1266
  - pcps\_opencl\_acquisition\_cc, 1272
  - pcps\_quicksync\_acquisition\_cc, 1278
  - pcps\_tong\_acquisition\_cc, 1283
- set\_switch\_position
  - Fpga\_Switch, 590
- set\_threshold
  - AcquisitionInterface, 336
  - BeidouB1iPcpsAcquisition, 402
  - BeidouB3iPcpsAcquisition, 416
  - galileo\_e5a\_noncoherentIQ\_acquisition\_caf\_cc, 614
  - galileo\_pcps\_8ms\_acquisition\_cc, 650
  - GalileoE1Pcps8msAmbiguousAcquisition, 678
  - GalileoE1PcpsAmbiguousAcquisition, 685
  - GalileoE1PcpsAmbiguousAcquisitionFpga, 691
  - GalileoE1PcpsCccwsrAmbiguousAcquisition, 696
  - GalileoE1PcpsQuickSyncAmbiguousAcquisition, 702
  - GalileoE1PcpsTongAmbiguousAcquisition, 708
  - GalileoE5aNoncoherentIQAcquisitionCaf, 728
  - GalileoE5aPcpsAcquisition, 733
  - GalileoE5aPcpsAcquisitionFpga, 741
  - GalileoE5bPcpsAcquisition, 755
  - GalileoE5bPcpsAcquisitionFpga, 763
  - GalileoE6PcpsAcquisition, 777
  - GlionassL1CaPcpsAcquisition, 838
  - GlionassL2CaPcpsAcquisition, 855
  - GpsL1CaPcpsAcquisition, 991
  - GpsL1CaPcpsAcquisitionFineDoppler, 996
  - GpsL1CaPcpsAcquisitionFpga, 1002
  - GpsL1CaPcpsAssistedAcquisition, 1008
  - GpsL1CaPcpsOpenCIAcquisition, 1013
  - GpsL1CaPcpsQuickSyncAcquisition, 1019
  - GpsL1CaPcpsTongAcquisition, 1024
  - GpsL2MPcpsAcquisition, 1050
  - GpsL2MPcpsAcquisitionFpga, 1055
  - GpsL5iPcpsAcquisition, 1070
  - GpsL5iPcpsAcquisitionFpga, 1076
  - pcps\_acquisition, 1246
  - pcps\_acquisition\_fine\_doppler\_cc, 1252
  - pcps\_acquisition\_fpga, 1256
  - pcps\_assisted\_acquisition\_cc, 1261
  - pcps\_cccwsr\_acquisition\_cc, 1266
  - pcps\_opencl\_acquisition\_cc, 1272
  - pcps\_quicksync\_acquisition\_cc, 1278
  - pcps\_tong\_acquisition\_cc, 1283
- set\_time\_offset\_s
  - Pvt\_Solution, 1319
- set\_time\_stamp
  - Galileo\_Cnav\_Message, 607
- set\_tow
  - Galileo\_Cnav\_Message, 607
- set\_TOW0\_flag
  - Galileo\_Inav\_Message, 641
- set\_TOW1\_flag
  - Galileo\_Fnav\_Message, 625
- set\_TOW2\_flag
  - Galileo\_Fnav\_Message, 625
- set\_TOW3\_flag
  - Galileo\_Fnav\_Message, 625
- set\_TOW4\_flag
  - Galileo\_Fnav\_Message, 625



- set\_TOW5\_flag
  - Galileo\_Inav\_Message, [641](#)
- set\_TOW6\_flag
  - Galileo\_Inav\_Message, [641](#)
- set\_tow\_sf0
  - OSNMA\_NavData, [1236](#)
- set\_update\_verified\_bits
  - OSNMA\_NavData, [1236](#)
- set\_utc\_data
  - OSNMA\_NavData, [1236](#)
- set\_verified\_status
  - OSNMA\_NavData, [1236](#)
- SGN\_LAMBDA
  - rtklib\_lambda.h, [1816](#)
- shift\_current\_word
  - IONGSMSChunkUnpackingCtx< WT >, [1119](#)
- shift\_padding
  - IONGSMSChunkUnpackingCtx< WT >, [1119](#)
- shift\_sample
  - IONGSMSChunkUnpackingCtx< WT >, [1119](#)
- short\_x2\_to\_cshort, [1421](#)
- short\_x2\_to\_cshort.h, [1872](#), [1873](#)
- short\_x2\_to\_cshort\_sptr
  - algorithms\_libs, [96](#)
- show\_local\_time\_zone
  - Pvt\_Conf, [1313](#)
- sig\_type
  - Beidou\_Dnav\_Ephemeris, [384](#)
- SIGN
  - ion\_gsms\_stream\_encodings.h, [2102](#)
- Signal
  - Gnss\_Synchro, [904](#)
- signal
  - Dll\_Pll\_Conf, [496](#)
  - Dll\_Pll\_Conf\_Fpga, [503](#)
  - Kf\_Conf, [1138](#)
  - Nav\_Message\_Packet, [1189](#)
- Signal Conditioner, [83](#)
- Signal model, [21](#)
- Signal Source, [178](#)
- signal\_conditioner.h, [1678](#), [1679](#)
- SIGNAL\_FREQ\_MAP
  - core\_system\_parameters, [313](#)
- signal\_generator
  - GenSignalSource, [785](#)
- signal\_generator.h, [1978](#), [1979](#)
- signal\_generator\_c, [1423](#)
  - signal\_make\_generator\_c, [1424](#)
- signal\_generator\_c.h, [1980](#), [1982](#)
  - signal\_generator\_c\_sptr, [1981](#)
  - signal\_make\_generator\_c, [1981](#)
- signal\_generator\_c\_sptr
  - signal\_generator\_c.h, [1981](#)
- signal\_health
  - Gps\_CNAV\_Ephemeris, [929](#)
- signal\_make\_generator\_c
  - signal\_generator\_c, [1424](#)
  - signal\_generator\_c.h, [1981](#)
- signal\_mask
  - Galileo\_HAS\_data, [635](#)
- signal\_source\_adapters, [178](#)
- signal\_source\_base.h, [2029](#)
- signal\_source\_gr\_blocks, [180](#)
  - ad936x\_iio\_source\_sptr, [181](#)
  - b\_io\_context, [181](#)
  - labsat23\_source\_sptr, [181](#)
  - rtl\_tcp\_signal\_source\_c\_sptr, [182](#)
  - unpack\_2bit\_samples\_sptr, [182](#)
  - unpack\_byte\_2bit\_cpx\_samples\_sptr, [182](#)
  - unpack\_byte\_2bit\_samples\_sptr, [182](#)
  - unpack\_byte\_4bit\_samples\_sptr, [182](#)
  - unpack\_intspir\_1bit\_samples\_sptr, [182](#)
  - unpack\_short\_byte\_samples\_sptr, [182](#)
  - unpack\_spir\_gss6450\_samples\_sptr, [182](#)
- signal\_source\_interface.h, [2323](#)
- signal\_source\_libs, [183](#)
  - FIR\_BUF\_SIZE, [184](#)
  - iodev, [185](#)
  - RTL\_TCP\_COMMAND, [185](#)
  - rtl\_tcp\_command, [185](#)
- SignalConditioner, [1424](#)
  - ~SignalConditioner, [1427](#)
  - connect, [1427](#)
  - data\_type\_adapter, [1427](#)
  - disconnect, [1427](#)
  - get\_left\_block, [1427](#)
  - get\_right\_block, [1428](#)
  - implementation, [1428](#)
  - input\_filter, [1428](#)
  - item\_size, [1428](#)
  - resampler, [1428](#)
  - role, [1428](#)
  - SignalConditioner, [1427](#)
- SignalGenerator, [1428](#)
  - connect, [1431](#)
  - disconnect, [1431](#)
  - get\_left\_block, [1431](#)
  - get\_right\_block, [1431](#)
  - implementation, [1431](#)
  - item\_size, [1431](#)
  - role, [1431](#)
- SignalSourceBase, [1432](#)
  - decode\_item\_type, [1434](#)
  - get\_left\_block, [1434](#)
  - getRfChannels, [1434](#)
  - implementation, [1435](#)
  - role, [1435](#)
  - SignalSourceBase, [1434](#)
- SignalSourceInterface, [1435](#)
  - GNSS block interfaces, [207](#)
- sigs
  - msm\_h\_t, [1178](#)
- SinBocCorrelationFunction
  - tracking\_libs, [204](#)
- SISA
  - Galileo\_Ephemeris, [622](#)

- size
  - algorithms\_libs, [112](#)
  - Concurrent\_Map< Data >, [457](#)
  - Concurrent\_Queue< Data >, [459](#)
- SIZE\_DSM\_BLOCKS\_BYTES
  - core\_system\_parameters, [313](#)
- Skew\_symmetric
  - algorithms\_libs, [112](#)
- skipped
  - Tag, [1474](#)
- slip
  - ssat\_t, [1460](#)
- slipc
  - solstat\_t, [1446](#)
  - ssat\_t, [1460](#)
- slope
  - Dll\_Pll\_Conf, [496](#)
  - Dll\_Pll\_Conf\_Fpga, [503](#)
  - Kf\_Conf, [1138](#)
- SM
  - ion\_gsms\_stream\_encodings.h, [2103](#)
- SMA
  - ion\_gsms\_stream\_encodings.h, [2103](#)
- smooth
  - msm\_h\_t, [1178](#)
- smoother\_length
  - Dll\_Pll\_Conf, [496](#)
  - Dll\_Pll\_Conf\_Fpga, [503](#)
  - Kf\_Conf, [1138](#)
- smoothing\_factor
  - Obs\_Conf, [1215](#)
- SNR
  - obsd\_t, [1218](#)
- snr
  - lexmsg\_t, [1152](#)
  - sdr\_gnss\_packet\_t, [1409](#)
  - solstat\_t, [1446](#)
  - ssat\_t, [1460](#)
- snrmask
  - prcopt\_t, [1301](#)
- snrmask\_t, [1437](#)
  - ena, [1437](#)
  - mask, [1437](#)
- sock
  - tcp\_t, [1477](#)
- socket\_t
  - algorithms\_libs\_rtklib, [137](#)
- sol
  - rtk\_t, [1359](#)
- sol\_t, [1438](#)
  - age, [1438](#)
  - dtr, [1438](#)
  - ns, [1439](#)
  - qr, [1439](#)
  - ratio, [1439](#)
  - rr, [1439](#)
  - stat, [1439](#)
  - thres, [1439](#)
  - time, [1439](#)
  - type, [1439](#)
- solbuf
  - rtksvr\_t, [1378](#)
- solbuf\_t, [1440](#)
  - buff, [1441](#)
  - cyclic, [1441](#)
  - data, [1441](#)
  - end, [1441](#)
  - n, [1441](#)
  - nb, [1441](#)
  - nmax, [1441](#)
  - rb, [1441](#)
  - start, [1441](#)
  - time, [1441](#)
- SOLF\_ENU
  - algorithms\_libs\_rtklib, [161](#)
- SOLF\_GSIF
  - algorithms\_libs\_rtklib, [161](#)
- SOLF\_LLH
  - algorithms\_libs\_rtklib, [161](#)
- SOLF\_NMEA
  - algorithms\_libs\_rtklib, [162](#)
- SOLF\_STAT
  - algorithms\_libs\_rtklib, [162](#)
- SOLF\_XYZ
  - algorithms\_libs\_rtklib, [162](#)
- soloapt
  - rtksvr\_t, [1378](#)
- SOLOPT\_DEFAULT
  - rtklib\_rtksvr.h, [1854](#)
- soloapt\_t, [1442](#)
  - datum, [1443](#)
  - degf, [1443](#)
  - geoid, [1443](#)
  - height, [1443](#)
  - maxsolstd, [1443](#)
  - nmeaintv, [1443](#)
  - outhead, [1443](#)
  - outopt, [1443](#)
  - posf, [1443](#)
  - prog, [1443](#)
  - sep, [1443](#)
  - solstatic, [1443](#)
  - sstat, [1444](#)
  - timef, [1444](#)
  - times, [1444](#)
  - timeu, [1444](#)
  - trace, [1444](#)
- SOLQ\_DGPS
  - algorithms\_libs\_rtklib, [162](#)
- SOLQ\_DR
  - algorithms\_libs\_rtklib, [162](#)
- SOLQ\_FIX
  - algorithms\_libs\_rtklib, [162](#)
- SOLQ\_FLOAT
  - algorithms\_libs\_rtklib, [163](#)
- SOLQ\_NONE

- algorithms\_libs\_rtklib, 163
- SOLQ\_PPP
  - algorithms\_libs\_rtklib, 163
- SOLQ\_SBAS
  - algorithms\_libs\_rtklib, 163
- SOLQ\_SINGLE
  - algorithms\_libs\_rtklib, 163
- solstat\_t, 1445
  - az, 1446
  - el, 1446
  - flag, 1446
  - freq, 1446
  - lock, 1446
  - outc, 1446
  - rejc, 1446
  - resc, 1446
  - resp, 1446
  - sat, 1446
  - slipc, 1446
  - snr, 1446
  - time, 1446
- solstatbuf\_t, 1447
  - data, 1448
  - n, 1448
  - nmax, 1448
- solstatic
  - solopt\_t, 1443
- soltype
  - prcopt\_t, 1301
- solution\_status
  - Monitor\_Pvt, 1174
- solution\_type
  - Monitor\_Pvt, 1174
- source
  - FileSourceBase, 553
  - FileTimestampSignalSource, 557
  - FourBitCpxFileSignalSource, 576
  - NsrFileSignalSource, 1211
  - SpirFileSignalSource, 1452
  - TwoBitCpxFileSignalSource, 1506
  - TwoBitPackedFileSignalSource, 1511
- source\_item\_size
  - FileSourceBase, 553
- spare1
  - Beidou\_Dnav\_Ephemeris, 384
  - Gps\_Ephemeris, 943
- spare2
  - Beidou\_Dnav\_Ephemeris, 384
  - Gps\_Ephemeris, 943
- spc
  - Dll\_Pll\_Conf, 496
  - Dll\_Pll\_Conf\_Fpga, 503
  - Kf\_Conf, 1138
- speed
  - file\_t, 539
- SPEED\_OF\_LIGHT\_M\_MS
  - core\_system\_parameters, 313
- SPEED\_OF\_LIGHT\_M\_S
  - core\_system\_parameters, 313
- SPI\_close
  - Fpga\_spidev, 589
- SPI\_open
  - Fpga\_spidev, 589
- spir\_file\_signal\_source.h, 2030, 2031
- spir\_gss6450\_file\_signal\_source.h, 2031, 2032
- SpirFileSignalSource, 1448
  - create\_file\_source\_hook, 1452
  - itemTypeToSize, 1452
  - post\_disconnect\_hook, 1452
  - pre\_connect\_hook, 1452
  - source, 1452
- SpirGSS6450FileSignalSource, 1453
  - connect, 1456
  - disconnect, 1456
  - filename, 1456
  - get\_left\_block, 1456
  - get\_right\_block, 1456, 1457
  - item\_size, 1457
  - item\_type, 1457
  - repeat, 1457
  - samples, 1457
  - sampling\_frequency, 1457
- sptr
  - FifoReader, 532
  - Gr\_Complex\_Lp\_Packet\_Source, 1085
  - IONGSMSFileSource, 1121
- SQRT2
  - rtklib\_ppp.h, 1824
- SQRT\_A\_LSB
  - core\_system\_parameters, 313
- sqrtA
  - Gnss\_Almanac, 864
  - Gnss\_Ephemeris, 877
- ssat
  - rtk\_t, 1359
- ssat\_t, 1458
  - azel, 1459
  - fix, 1459
  - gf, 1459
  - gf2, 1459
  - half, 1459
  - lock, 1459
  - mw, 1459
  - outc, 1459
  - ph, 1459
  - phw, 1459
  - pt, 1459
  - rejc, 1460
  - resc, 1460
  - resp, 1460
  - slip, 1460
  - slipc, 1460
  - snr, 1460
  - sys, 1460
  - vs, 1460
  - vsat, 1460



- ssr
  - nav\_t, [1195](#)
  - rtcm\_t, [1356](#)
- ssr\_t, [1461](#)
  - cbias, [1462](#)
  - dclk, [1462](#)
  - ddepth, [1462](#)
  - depth, [1462](#)
  - hrclk, [1462](#)
  - iod, [1462](#)
  - iodcrc, [1462](#)
  - iode, [1462](#)
  - pbias, [1462](#)
  - refd, [1462](#)
  - stdpb, [1462](#)
  - t0, [1462](#)
  - udi, [1463](#)
  - update, [1463](#)
  - ura, [1463](#)
  - yaw\_ang, [1463](#)
  - yaw\_rate, [1463](#)
- SSRUDINT
  - rtklib\_rtc3.h, [1837](#)
- sstat
  - solo\_t, [1444](#)
- sta
  - raw\_t, [1326](#)
  - rtcm\_t, [1356](#)
- sta\_t, [1464](#)
  - antdes, [1464](#)
  - antsetup, [1464](#)
  - antsno, [1465](#)
  - del, [1465](#)
  - delttype, [1465](#)
  - hgt, [1465](#)
  - itrf, [1465](#)
  - marker, [1465](#)
  - name, [1465](#)
  - pos, [1465](#)
  - recsno, [1465](#)
  - rectype, [1465](#)
  - recver, [1465](#)
- stah
  - rtcm\_t, [1356](#)
- staid
  - rtcm\_t, [1356](#)
- start
  - ad936x\_iio\_source, [341](#)
  - DMASignalSourceFPGA, [511](#)
  - FifoReader, [532](#)
  - file\_t, [539](#)
  - GNSSBlockInterface, [911](#)
  - GNSSFlowgraph, [914](#)
  - solbuf\_t, [1441](#)
- start\_acquisition
  - Channel, [430](#)
  - ChannelInterface, [440](#)
- start\_tracking
  - BeidouB1iDIIPIITracking, [397](#)
  - BeidouB3iDIIPIITracking, [411](#)
  - dll\_pll\_veml\_tracking\_fpga, [508](#)
  - GalileoE1DIIPIIVemlTracking, [668](#)
  - GalileoE1DIIPIIVemlTrackingFpga, [673](#)
  - GalileoE1TcpConnectorTracking, [713](#)
  - GalileoE5aDIIPIITracking, [717](#)
  - GalileoE5aDIIPIITrackingFpga, [722](#)
  - GalileoE5bDIIPIITracking, [749](#)
  - GalileoE6DIIPIITracking, [771](#)
  - GlionassL1CaDIIPIICAidTracking, [829](#)
  - GlionassL1CaDIIPIITracking, [833](#)
  - GlionassL2CaDIIPIICAidTracking, [846](#)
  - GlionassL2CaDIIPIITracking, [850](#)
  - GpsL1CaDIIPIITracking, [968](#)
  - GpsL1CaDIIPIITrackingFpga, [973](#)
  - GpsL1CaDIIPIITrackingGPU, [977](#)
  - GpsL1CaGaussianTracking, [981](#)
  - GpsL1CaKfTracking, [985](#)
  - GpsL1CaTcpConnectorTracking, [1028](#)
  - GpsL2MDIIPIITracking, [1040](#)
  - GpsL2MDIIPIITrackingFpga, [1044](#)
  - GpsL5DIIPIITracking, [1059](#)
  - GpsL5DIIPIITrackingFpga, [1064](#)
  - TrackingInterface, [1500](#)
- stas
  - pppcorr\_t, [1293](#)
- stat
  - lexmsg\_t, [1152](#)
  - sol\_t, [1439](#)
- state
  - ftp\_t, [595](#)
  - ntrip\_t, [1213](#)
  - rtksvr\_t, [1378](#)
  - stream\_t, [1470](#)
  - tcp\_t, [1478](#)
- status
  - sdr\_gnss\_packet\_t, [1409](#)
  - Tag, [1474](#)
- std
  - fcdb\_t, [530](#)
  - pclk\_t, [1241](#)
  - peph\_t, [1288](#)
  - prcopt\_t, [1302](#)
  - stec\_t, [1467](#)
  - trop\_t, [1501](#)
- stdpb
  - ssr\_t, [1462](#)
- stec
  - pppcorr\_t, [1293](#)
- stec\_t, [1466](#)
  - azel, [1466](#)
  - flag, [1466](#)
  - ion, [1466](#)
  - sat, [1466](#)
  - std, [1467](#)
  - time, [1467](#)
- step\_bytes

- ad936x\_iio\_samples, 339
- stop
  - ad936x\_iio\_source, 341
  - GNSSFlowgraph, 914
- stop\_acquisition
  - AcquisitionInterface, 337
  - BeidouB1iPcpsAcquisition, 403
  - BeidouB3iPcpsAcquisition, 416
  - Fpga\_Acquisition, 579
  - GalileoE1Pcps8msAmbiguousAcquisition, 679
  - GalileoE1PcpsAmbiguousAcquisition, 685
  - GalileoE1PcpsAmbiguousAcquisitionFpga, 691
  - GalileoE1PcpsCccwsrAmbiguousAcquisition, 697
  - GalileoE1PcpsQuickSyncAmbiguousAcquisition, 703
  - GalileoE1PcpsTongAmbiguousAcquisition, 709
  - GalileoE5aNoncoherentIQAcquisitionCaf, 728
  - GalileoE5aPcpsAcquisition, 734
  - GalileoE5aPcpsAcquisitionFpga, 741
  - GalileoE5bPcpsAcquisition, 755
  - GalileoE5bPcpsAcquisitionFpga, 763
  - GalileoE6PcpsAcquisition, 777
  - GlonassL1CaPcpsAcquisition, 838
  - GlonassL2CaPcpsAcquisition, 855
  - GpsL1CaPcpsAcquisition, 991
  - GpsL1CaPcpsAcquisitionFineDoppler, 996
  - GpsL1CaPcpsAcquisitionFpga, 1002
  - GpsL1CaPcpsAssistedAcquisition, 1008
  - GpsL1CaPcpsOpenCIAcquisition, 1014
  - GpsL1CaPcpsQuickSyncAcquisition, 1019
  - GpsL1CaPcpsTongAcquisition, 1024
  - GpsL2MPcpsAcquisition, 1050
  - GpsL2MPcpsAcquisitionFpga, 1055
  - GpsL5iPcpsAcquisition, 1070
  - GpsL5iPcpsAcquisitionFpga, 1076
  - pcps\_acquisition\_fpga, 1256
- stop\_channel
  - Channel, 430
  - ChannelInterface, 440
- stop\_server
  - Rtcm, 1350
- stop\_tracking
  - BeidouB1iDIIPITracking, 397
  - BeidouB3iDIIPITracking, 411
  - dll\_pll\_veml\_tracking\_fpga, 508
  - GalileoE1DIIPIVemlTracking, 668
  - GalileoE1DIIPIVemlTrackingFpga, 673
  - GalileoE1TcpConnectorTracking, 713
  - GalileoE5aDIIPITracking, 717
  - GalileoE5aDIIPITrackingFpga, 722
  - GalileoE5bDIIPITracking, 749
  - GalileoE6DIIPITracking, 771
  - GlonassL1CaDIIPICAidTracking, 829
  - GlonassL1CaDIIPITracking, 833
  - GlonassL2CaDIIPICAidTracking, 846
  - GlonassL2CaDIIPITracking, 850
  - GpsL1CaDIIPITracking, 968
  - GpsL1CaDIIPITrackingFpga, 973
  - GpsL1CaDIIPITrackingGPU, 977
  - GpsL1CaGaussianTracking, 981
  - GpsL1CaKfTracking, 985
  - GpsL1CaTcpConnectorTracking, 1028
  - GpsL2MDIIPITracking, 1040
  - GpsL2MDIIPITrackingFpga, 1044
  - GpsL5DIIPITracking, 1059
  - GpsL5DIIPITrackingFpga, 1064
  - TrackingInterface, 1500
- store\_public\_key
  - Gnss\_Crypto, 869
- str
  - ntrip\_t, 1213
- STR\_FILE
  - algorithms\_libs\_rtklib, 137
- STR\_FTP
  - algorithms\_libs\_rtklib, 137
- STR\_HTTP
  - algorithms\_libs\_rtklib, 137
- STR\_MODE\_R
  - algorithms\_libs\_rtklib, 137
- STR\_MODE\_RW
  - algorithms\_libs\_rtklib, 137
- STR\_MODE\_W
  - algorithms\_libs\_rtklib, 137
- STR\_NONE
  - algorithms\_libs\_rtklib, 138
- STR\_NTRIPCLI
  - algorithms\_libs\_rtklib, 138
- STR\_NTRIPSVR
  - algorithms\_libs\_rtklib, 138
- STR\_SERIAL
  - algorithms\_libs\_rtklib, 138
- STR\_TCPCLI
  - algorithms\_libs\_rtklib, 138
- STR\_TCPSVR
  - algorithms\_libs\_rtklib, 138
- STR\_UDP
  - algorithms\_libs\_rtklib, 138
- stream
  - rtksvr\_t, 1379
- stream\_cfg, 1467
  - bw\_hz, 1467
  - fs\_hz, 1467
  - lo\_hz, 1467
  - rfport, 1468
- stream\_t, 1468
  - inb, 1469
  - inbt, 1469
  - inr, 1469
  - lock, 1469
  - mode, 1469
  - msg, 1469
  - outb, 1469
  - outbt, 1469
  - outr, 1469
  - path, 1469
  - port, 1470

- state, [1470](#)
- tact, [1470](#)
- tick, [1470](#)
- type, [1470](#)
- StreamEncoding
  - ion\_gsms\_stream\_encodings.h, [2101](#)
- STRFMT\_NMEA
  - algorithms\_libs\_rtklib, [163](#)
- STRFMT\_RNXCLK
  - algorithms\_libs\_rtklib, [164](#)
- STRFMT\_RTCM2
  - algorithms\_libs\_rtklib, [164](#)
- STRFMT\_RTCM3
  - algorithms\_libs\_rtklib, [164](#)
- STRFMT\_SBAS
  - algorithms\_libs\_rtklib, [164](#)
- STRFMT\_SP3
  - algorithms\_libs\_rtklib, [164](#)
- string\_converter.h, [2368](#), [2369](#)
- string\_decoder
  - Glonass\_Gnav\_Navigation\_Message, [808](#)
- StringConverter, [1470](#)
- subframe\_decoder
  - Gps\_Navigation\_Message, [960](#)
- subfrm
  - raw\_t, [1327](#)
- SV\_accuracy
  - Beidou\_Dnav\_Ephemeris, [384](#)
  - Gps\_Ephemeris, [943](#)
- sv\_clock\_drift
  - Glonass\_Gnav\_Ephemeris, [799](#)
  - Gnss\_Ephemeris, [873](#)
- SV\_health
  - Beidou\_Dnav\_Almanac, [377](#)
  - Beidou\_Dnav\_Ephemeris, [384](#)
  - Gps\_Almanac, [922](#)
  - Gps\_Ephemeris, [943](#)
- sva
  - eph\_t, [523](#)
  - geph\_t, [789](#)
  - seph\_t, [1410](#)
- svconf
  - alm\_t, [355](#)
- svh
  - alm\_t, [355](#)
  - eph\_t, [523](#)
  - geph\_t, [789](#)
  - seph\_t, [1410](#)
- SVID1\_7
  - Galileo\_Almanac\_Helper, [605](#)
- SVID2\_8
  - Galileo\_Almanac\_Helper, [605](#)
- SVID3\_9
  - Galileo\_Almanac\_Helper, [605](#)
- svr
  - tcpcli\_t, [1479](#)
  - tcpsvr\_t, [1481](#)
- SWAP\_D
  - rtklib\_ppp.h, [1824](#)
- SWAP\_I
  - rtklib\_ppp.h, [1824](#)
- SWAP\_LAMBDA
  - rtklib\_lambda.h, [1816](#)
- swapintv
  - file\_t, [539](#)
- swift\_common.h, [2169](#), [2170](#)
- symbols
  - cnav\_v27\_part\_t, [452](#)
- syncsol
  - prcopt\_t, [1302](#)
- sys
  - ssat\_t, [1460](#)
- SYS\_ALL
  - algorithms\_libs\_rtklib, [164](#)
- SYS\_BDS
  - algorithms\_libs\_rtklib, [164](#)
- SYS\_GAL
  - algorithms\_libs\_rtklib, [165](#)
- SYS\_GLO
  - algorithms\_libs\_rtklib, [165](#)
- SYS\_GPS
  - algorithms\_libs\_rtklib, [165](#)
- SYS\_IRN
  - algorithms\_libs\_rtklib, [165](#)
- SYS\_LEO
  - algorithms\_libs\_rtklib, [165](#)
- SYS\_NONE
  - algorithms\_libs\_rtklib, [165](#)
- SYS\_QZS
  - algorithms\_libs\_rtklib, [166](#)
- SYS\_SBS
  - algorithms\_libs\_rtklib, [166](#)
- System
  - Gnss\_Almanac, [864](#)
  - Gnss\_Ephemeris, [877](#)
  - Gnss\_Synchro, [904](#)
- system
  - Dll\_Pll\_Conf, [496](#)
  - Dll\_Pll\_Conf\_Fpga, [503](#)
  - Kf\_Conf, [1138](#)
  - Nav\_Message\_Packet, [1189](#)
- system\_ecef\_pos\_sd\_m
  - Pvt\_Conf, [1313](#)
- system\_ecef\_vel\_sd\_ms
  - Pvt\_Conf, [1313](#)
- t0
  - dgps\_t, [483](#)
  - lexion\_t, [1150](#)
  - sbsfcrr\_t, [1397](#)
  - sbsigp\_t, [1398](#)
  - sbslcorr\_t, [1402](#)
  - seph\_t, [1410](#)
  - ssr\_t, [1462](#)
- t0a\_7
  - Galileo\_Almanac\_Helper, [605](#)
- T0A\_7\_LSB

- core\_system\_parameters, 313
- t0a\_9
  - Galileo\_Almanac\_Helper, 605
- T0A\_9\_LSB
  - core\_system\_parameters, 313
- T0C\_4\_LSB
  - core\_system\_parameters, 313
- T0E\_1\_LSB
  - core\_system\_parameters, 313
- T0T\_6\_LSB
  - core\_system\_parameters, 314
- T\_0\_G\_10\_LSB
  - core\_system\_parameters, 314
- t\_0G
  - Galileo\_Utc\_Model, 659
- T\_GD\_LSB
  - core\_system\_parameters, 314
- T\_OA
  - core\_system\_parameters, 246
- T\_OA\_LSB
  - core\_system\_parameters, 314
- T\_OC\_LSB
  - core\_system\_parameters, 314
- T\_OE\_LSB
  - core\_system\_parameters, 314
- T\_OT\_LSB
  - core\_system\_parameters, 314
- tact
  - stream\_t, 1470
  - tcp\_t, 1478
- Tag, 1472
  - ADKD, 1473
  - computed\_tag, 1473
  - cop, 1473
  - CTR, 1474
  - e\_verification\_status, 1473
  - id\_counter, 1474
  - nav\_data, 1474
  - PRN\_d, 1474
  - PRNa, 1474
  - received\_tag, 1474
  - skipped, 1474
  - status, 1474
  - Tag, 1473
  - tag\_id, 1474
  - TOW, 1474
  - WN, 1474
- tag
  - MACK\_tag\_and\_info, 1161
- tag0
  - MACK\_header, 1156
- tag\_and\_info
  - MACK\_message, 1160
- tag\_id
  - Tag, 1474
- tag\_info
  - MACK\_tag\_and\_info, 1161
- taun
  - geph\_t, 789
- tbase
  - raw\_t, 1327
- TC
  - ion\_gsms\_stream\_encodings.h, 2103
- TCA
  - ion\_gsms\_stream\_encodings.h, 2103
- tcon
  - tcp\_t, 1478
- tcp
  - ntrip\_t, 1213
- tcp\_cmd\_interface.h, 2397
- Tcp\_Communication, 1475
- tcp\_communication.h, 2299, 2300
- Tcp\_Packet\_Data, 1475
  - proc\_pack\_carr\_error, 1476
  - proc\_pack\_carrier\_doppler\_hz, 1476
  - proc\_pack\_code\_error, 1476
- tcp\_packet\_data.h, 2301, 2302
- tcp\_t, 1477
  - addr, 1477
  - port, 1477
  - saddr, 1477
  - sock, 1477
  - state, 1478
  - tact, 1478
  - tcon, 1478
  - tdis, 1478
- tcpcli\_t, 1479
  - svr, 1479
  - tirecon, 1479
  - toinact, 1479
- TcpCmdInterface, 1480
  - get\_LLH, 1480
  - get\_utc\_time, 1480
- tcpsvr\_t, 1481
  - cli, 1481
  - svr, 1481
- tdis
  - tcp\_t, 1478
- te
  - fcbd\_t, 530
  - half\_cyc\_tag, 1088
  - pcv\_t, 1285
- tec
  - nav\_t, 1195
- tec\_t, 1482
  - data, 1482
  - hgts, 1482
  - lats, 1482
  - lons, 1483
  - ndata, 1483
  - rb, 1483
  - rms, 1483
  - time, 1483
- telemetry
  - Channel, 430
- Telemetry Decoder, 185

- telemetry\_decoder\_adapters, [186](#)
- telemetry\_decoder\_gr\_blocks, [187](#)
  - beidou\_b1i\_telemetry\_decoder\_gs\_sptr, [188](#)
  - beidou\_b3i\_telemetry\_decoder\_gs\_sptr, [188](#)
  - galileo\_telemetry\_decoder\_gs\_sptr, [188](#)
  - glonass\_l1\_ca\_telemetry\_decoder\_gs\_sptr, [188](#)
  - glonass\_l2\_ca\_telemetry\_decoder\_gs\_sptr, [189](#)
  - gps\_l1\_ca\_telemetry\_decoder\_gs\_sptr, [189](#)
  - gps\_l2c\_telemetry\_decoder\_gs\_sptr, [189](#)
  - gps\_l5\_telemetry\_decoder\_gs\_sptr, [189](#)
  - sbas\_l1\_telemetry\_decoder\_gs\_sptr, [189](#)
- telemetry\_decoder\_interface.h, [2324](#)
- telemetry\_decoder\_libs, [192](#)
- telemetry\_decoder\_libswiftcnv, [189](#)
  - ABS, [190](#)
  - CLAMP\_DIFF, [190](#)
  - GPS\_L2\_V27\_HISTORY\_LENGTH\_BITS, [191](#)
  - GPS\_L2C\_V27\_DECODE\_BITS, [191](#)
  - GPS\_L2C\_V27\_DELAY\_BITS, [191](#)
  - GPS\_L2C\_V27\_INIT\_BITS, [191](#)
  - MAX, [191](#)
  - MIN, [191](#)
- TelemetryDecoderInterface, [1483](#)
- temp\_buffer\_needed
  - cl\_fft\_plan, [443](#)
- tempmemobj
  - cl\_fft\_plan, [443](#)
- tempmemobj\_imag
  - cl\_fft\_plan, [443](#)
- tempmemobj\_real
  - cl\_fft\_plan, [443](#)
- TEXT\_BLACK
  - core\_system\_parameters, [314](#)
- TEXT\_BLUE
  - core\_system\_parameters, [314](#)
- TEXT\_BOLD\_BLACK
  - core\_system\_parameters, [314](#)
- TEXT\_BOLD\_BLUE
  - core\_system\_parameters, [314](#)
- TEXT\_BOLD\_CYAN
  - core\_system\_parameters, [314](#)
- TEXT\_BOLD\_GREEN
  - core\_system\_parameters, [315](#)
- TEXT\_BOLD\_MAGENTA
  - core\_system\_parameters, [315](#)
- TEXT\_BOLD\_RED
  - core\_system\_parameters, [315](#)
- TEXT\_BOLD\_WHITE
  - core\_system\_parameters, [315](#)
- TEXT\_BOLD\_YELLOW
  - core\_system\_parameters, [315](#)
- TEXT\_CYAN
  - core\_system\_parameters, [315](#)
- TEXT\_GREEN
  - core\_system\_parameters, [315](#)
- TEXT\_MAGENTA
  - core\_system\_parameters, [315](#)
- TEXT\_RED
  - core\_system\_parameters, [315](#)
- TEXT\_RESET
  - core\_system\_parameters, [315](#)
- TEXT\_WHITE
  - core\_system\_parameters, [315](#)
- TEXT\_YELLOW
  - core\_system\_parameters, [315](#)
- TGD
  - Gps\_CNAV\_Ephemeris, [929](#)
  - Gps\_Ephemeris, [943](#)
- tgdt
  - eph\_t, [523](#)
  - lexeph\_t, [1149](#)
- TGD1
  - Beidou\_Dnav\_Ephemeris, [384](#)
- TGD2
  - Beidou\_Dnav\_Ephemeris, [384](#)
- there\_are\_e1\_channels
  - Tlm\_Conf, [1491](#)
- there\_are\_e6\_channels
  - Tlm\_Conf, [1491](#)
- thread
  - ftp\_t, [595](#)
  - rtksvr\_t, [1379](#)
- three\_bit\_look\_up
  - ion\_gsms\_stream\_encodings.h, [2103](#)
- thres
  - sol\_t, [1439](#)
- THRES\_RES
  - rtklib\_ppp.h, [1824](#)
- thresar
  - prcopt\_t, [1302](#)
- thresslip
  - prcopt\_t, [1302](#)
- tick
  - file\_t, [539](#)
  - rtksvr\_t, [1379](#)
  - stream\_t, [1470](#)
- tick\_f
  - file\_t, [539](#)
- tidecorr
  - prcopt\_t, [1302](#)
- time
  - file\_t, [539](#)
  - gtime\_t, [1086](#)
  - obsd\_t, [1218](#)
  - pclk\_t, [1241](#)
  - peph\_t, [1288](#)
  - raw\_t, [1327](#)
  - rtcm\_t, [1356](#)
  - sol\_t, [1439](#)
  - solbuf\_t, [1441](#)
  - solstat\_t, [1446](#)
  - stec\_t, [1467](#)
  - tec\_t, [1483](#)
  - trop\_t, [1501](#)
- time\_s
  - msm\_h\_t, [1178](#)

- rtcm\_t, [1356](#)
- time\_stamp
  - Galileo\_HAS\_page, [637](#)
- timed\_wait\_and\_pop
  - Concurrent\_Queue< Data >, [459](#)
- timef
  - solo\_t, [1444](#)
- times
  - solo\_t, [1444](#)
- TIMES\_GPST
  - algorithms\_libs\_rtklib, [166](#)
- TIMES\_JST
  - algorithms\_libs\_rtklib, [166](#)
- TIMES\_UTC
  - algorithms\_libs\_rtklib, [166](#)
- timetag
  - file\_t, [539](#)
- TIMETAGH\_LEN
  - algorithms\_libs\_rtklib, [166](#)
  - rtklib\_stream.h, [1867](#)
- timeu
  - solo\_t, [1444](#)
- tint
  - url\_t, [1527](#)
- tint\_s
  - msm\_h\_t, [1178](#)
- TINTACT
  - algorithms\_libs\_rtklib, [167](#)
  - rtklib\_stream.h, [1867](#)
- tirecon
  - tcpcli\_t, [1479](#)
- tlat
  - sbssat\_t, [1405](#)
- tle\_t, [1485](#)
  - data, [1486](#)
  - n, [1486](#)
  - nmax, [1486](#)
- tled\_t, [1487](#)
  - alias, [1488](#)
  - bstar, [1488](#)
  - desig, [1488](#)
  - ecc, [1488](#)
  - eleno, [1488](#)
  - epoch, [1488](#)
  - etype, [1488](#)
  - inc, [1488](#)
  - M, [1488](#)
  - n, [1488](#)
  - name, [1488](#)
  - nddot, [1488](#)
  - ndot, [1489](#)
  - OMG, [1489](#)
  - omg, [1489](#)
  - rev, [1489](#)
  - satclass, [1489](#)
  - satno, [1489](#)
- Tlm\_Conf, [1490](#)
  - dump, [1491](#)
  - dump\_crc\_stats, [1491](#)
  - dump\_crc\_stats\_filename, [1491](#)
  - dump\_filename, [1491](#)
  - dump\_mat, [1491](#)
  - enable\_navdata\_monitor, [1491](#)
  - enable\_reed\_solomon, [1491](#)
  - remove\_dat, [1491](#)
  - there\_are\_e1\_channels, [1491](#)
  - there\_are\_e6\_channels, [1491](#)
  - use\_ced, [1491](#)
- tlm\_conf.h, [2170](#), [2171](#)
- Tlm\_CRC\_Stats, [1492](#)
  - initialize, [1492](#)
  - set\_channel, [1492](#)
  - update\_CRC\_stats, [1492](#)
- tlm\_crc\_stats.h, [2172](#), [2173](#)
- tlm\_utils.h, [2174](#)
- tnext
  - ftp\_t, [595](#)
- toa
  - alm\_t, [355](#)
  - Gnss\_Almanac, [864](#)
- toas
  - alm\_t, [355](#)
- tobs
  - raw\_t, [1327](#)
- toc
  - eph\_t, [523](#)
  - Gnss\_Ephemeris, [877](#)
- tod
  - raw\_t, [1327](#)
- Todo List, [29](#)
- toe
  - eph\_t, [523](#)
  - geph\_t, [789](#)
  - Gnss\_Ephemeris, [877](#)
  - lexeph\_t, [1149](#)
- toe1
  - Gps\_CNAV\_Ephemeris, [929](#)
- toe2
  - Gps\_CNAV\_Ephemeris, [929](#)
- toes
  - eph\_t, [523](#)
- tof
  - geph\_t, [790](#)
  - lexeph\_t, [1149](#)
  - seph\_t, [1410](#)
- togeod
  - algorithms\_libs, [112](#)
- toh
  - mt1\_header, [1180](#)
- toinact
  - tcpcli\_t, [1479](#)
- top
  - Gps\_CNAV\_Ephemeris, [929](#)
- topocent
  - algorithms\_libs, [112](#)
- topts

- ftp\_t, 595
- tot
  - Galileo\_Utc\_Model, 659
  - Gps\_Utc\_Model, 963
- total\_block\_exp
  - Acq\_Conf\_Fpga, 332
- TOTRedCED
  - Galileo\_Reduced\_CED, 653
- TOW
  - MACK\_message, 1160
  - Tag, 1474
- tow
  - Agnss\_Ref\_Time, 351
  - cnav\_msg\_t, 449
  - Galileo\_HAS\_data, 635
  - Galileo\_HAS\_page, 637
  - Galileo\_Iono, 643
  - Gnss\_Ephemeris, 878
  - Gps\_Acq\_Assist, 918
  - sbsmsg\_t, 1403
- TOW\_at\_current\_symbol\_ms
  - Gnss\_Synchro, 904
  - Monitor\_Pvt, 1174
- tow\_at\_current\_symbol\_ms
  - Nav\_Message\_Packet, 1189
- tow\_ms
  - GnssTime, 915
- tow\_ms\_fraction
  - GnssTime, 915
- TOW\_sf0
  - OSNMA\_msg, 1231
- towh\_k
  - DSM\_KROOT\_message, 515
- trace
  - solopt\_t, 1444
- track\_pilot
  - Dll\_Pll\_Conf, 497
  - Dll\_Pll\_Conf\_Fpga, 503
  - Kf\_Conf, 1138
- Tracking, 193
- tracking
  - Channel, 430
- Tracking\_2nd\_DLL\_filter, 1493
  - get\_code\_nco, 1494
  - initialize, 1494
  - set\_DLL\_BW, 1494
  - set\_pdi, 1494
- tracking\_2nd\_DLL\_filter.h, 2302, 2303
- Tracking\_2nd\_PLL\_filter, 1494
  - set\_pdi, 1495
  - set\_PLL\_BW, 1495
- tracking\_2nd\_PLL\_filter.h, 2304
- tracking\_adapters, 193
- tracking\_discriminators.h, 2305, 2306
- Tracking\_FLL\_PLL\_filter, 1495
- tracking\_FLL\_PLL\_filter.h, 2309
- tracking\_gr\_blocks, 195
  - dll\_pll\_veml\_tracking\_fpga\_sptr, 196
  - dll\_pll\_veml\_tracking\_sptr, 196
  - galileo\_e1\_tcp\_connector\_tracking\_cc\_sptr, 197
  - glonass\_l1\_ca\_dll\_pll\_c\_aid\_tracking\_cc\_sptr, 197
  - glonass\_l1\_ca\_dll\_pll\_c\_aid\_tracking\_sc\_sptr, 197
  - glonass\_l1\_ca\_dll\_pll\_tracking\_cc\_sptr, 197
  - glonass\_l2\_ca\_dll\_pll\_c\_aid\_tracking\_cc\_sptr, 197
  - glonass\_l2\_ca\_dll\_pll\_c\_aid\_tracking\_sc\_sptr, 197
  - glonass\_l2\_ca\_dll\_pll\_tracking\_cc\_sptr, 197
  - gps\_l1\_ca\_dll\_pll\_tracking\_gpu\_cc\_sptr, 197
  - gps\_l1\_ca\_gaussian\_tracking\_cc\_sptr, 198
  - gps\_l1\_ca\_tcp\_connector\_tracking\_cc\_sptr, 198
- tracking\_interface.h, 2325, 2326
- tracking\_libs, 198
  - b\_io\_context, 200
  - CalculateSlope, 201
  - CalculateSlopeAbs, 201
  - carrier\_lock\_detector, 201
  - cn0\_m2m4\_estimator, 201
  - cn0\_svn\_estimator, 201
  - CosBocCorrelationFunction, 202
  - dll\_nc\_e\_minus\_l\_normalized, 202
  - dll\_nc\_vemlp\_normalized, 202
  - fil\_four\_quadrant\_atan, 203
  - GetYIntercept, 203
  - GetYInterceptAbs, 203
  - NUM\_RX\_VARIABLES, 200
  - NUM\_TX\_VARIABLES\_GALILEO\_E1, 200
  - NUM\_TX\_VARIABLES\_GPS\_L1\_CA, 200
  - phase\_unwrap, 203
  - pll\_cloop\_two\_quadrant\_atan, 204
  - pll\_four\_quadrant\_atan, 204
  - SinBocCorrelationFunction, 204
- Tracking\_loop\_filter, 1496
  - operator=, 1497
  - Tracking\_loop\_filter, 1497
- tracking\_loop\_filter.h, 2310, 2311
- Tracking\_sample\_counter
  - Gnss\_Synchro, 905
- TrackingCmd, 1497
  - carrier\_freq\_hz, 1498
  - carrier\_freq\_rate\_hz\_s, 1498
  - code\_freq\_chips, 1498
  - enable\_carrier\_nco\_cmd, 1498
  - enable\_code\_nco\_cmd, 1498
  - sample\_counter, 1498
- trackingcmd.h, 1874, 1875
- TrackingInterface, 1498
  - set\_channel, 1500
  - set\_gnss\_synchro, 1500
  - start\_tracking, 1500
  - stop\_tracking, 1500
- trop
  - pppcorr\_t, 1293
- trop\_t, 1501
  - std, 1501
  - time, 1501
  - trp, 1501
- tropopt

- prcopt\_t, [1302](#)
- TROPOPT\_COR
  - algorithms\_libs\_rtklib, [167](#)
- TROPOPT\_CORG
  - algorithms\_libs\_rtklib, [167](#)
- TROPOPT\_EST
  - algorithms\_libs\_rtklib, [167](#)
- TROPOPT\_ESTG
  - algorithms\_libs\_rtklib, [167](#)
- TROPOPT\_OFF
  - algorithms\_libs\_rtklib, [167](#)
- TROPOPT\_SAAS
  - algorithms\_libs\_rtklib, [168](#)
- TROPOPT\_SBAS
  - algorithms\_libs\_rtklib, [168](#)
- trp
  - trop\_t, [1501](#)
- try\_pop
  - Concurrent\_Queue< Data >, [459](#)
- ts
  - DSM\_KROOT\_message, [515](#)
  - fcdb\_t, [530](#)
  - half\_cyc\_tag, [1088](#)
  - pcv\_t, [1285](#)
- tspan
  - lexion\_t, [1151](#)
- tt
  - rtk\_t, [1359](#)
- TTOL\_MOVEB
  - algorithms\_libs\_rtklib, [168](#)
- ttr
  - eph\_t, [523](#)
- ttr
  - lexmsg\_t, [1152](#)
- twist\_kernel
  - cl\_fft\_plan, [443](#)
- two\_bit\_cpx\_file\_signal\_source.h, [2034](#)
- two\_bit\_look\_up
  - ion\_gsms\_stream\_encodings.h, [2103](#)
- two\_bit\_packed\_file\_signal\_source.h, [2035](#), [2036](#)
- TWO\_N10
  - core\_system\_parameters, [316](#)
- TWO\_N11
  - core\_system\_parameters, [316](#)
- TWO\_N14
  - core\_system\_parameters, [316](#)
- TWO\_N15
  - core\_system\_parameters, [316](#)
- TWO\_N16
  - core\_system\_parameters, [316](#)
- TWO\_N17
  - core\_system\_parameters, [316](#)
- TWO\_N18
  - core\_system\_parameters, [316](#)
- TWO\_N19
  - core\_system\_parameters, [316](#)
- TWO\_N2
  - core\_system\_parameters, [316](#)
- TWO\_N20
  - core\_system\_parameters, [316](#)
- TWO\_N21
  - core\_system\_parameters, [317](#)
- TWO\_N22
  - core\_system\_parameters, [317](#)
- TWO\_N23
  - core\_system\_parameters, [317](#)
- TWO\_N24
  - core\_system\_parameters, [317](#)
- TWO\_N25
  - core\_system\_parameters, [317](#)
- TWO\_N26
  - core\_system\_parameters, [317](#)
- TWO\_N27
  - core\_system\_parameters, [317](#)
- TWO\_N29
  - core\_system\_parameters, [317](#)
- TWO\_N30
  - core\_system\_parameters, [317](#)
- TWO\_N31
  - core\_system\_parameters, [317](#)
- TWO\_N32
  - core\_system\_parameters, [318](#)
- TWO\_N33
  - core\_system\_parameters, [318](#)
- TWO\_N34
  - core\_system\_parameters, [318](#)
- TWO\_N35
  - core\_system\_parameters, [318](#)
- TWO\_N38
  - core\_system\_parameters, [318](#)
- TWO\_N39
  - core\_system\_parameters, [318](#)
- TWO\_N40
  - core\_system\_parameters, [318](#)
- TWO\_N43
  - core\_system\_parameters, [318](#)
- TWO\_N44
  - core\_system\_parameters, [318](#)
- TWO\_N46
  - core\_system\_parameters, [318](#)
- TWO\_N48
  - core\_system\_parameters, [319](#)
- TWO\_N5
  - core\_system\_parameters, [319](#)
- TWO\_N50
  - core\_system\_parameters, [319](#)
- TWO\_N51
  - core\_system\_parameters, [319](#)
- TWO\_N55
  - core\_system\_parameters, [319](#)
- TWO\_N57
  - core\_system\_parameters, [319](#)
- TWO\_N59
  - core\_system\_parameters, [319](#)
- TWO\_N6
  - core\_system\_parameters, [319](#)



- TWO\_N60
  - core\_system\_parameters, 319
- TWO\_N66
  - core\_system\_parameters, 319
- TWO\_N68
  - core\_system\_parameters, 320
- TWO\_N8
  - core\_system\_parameters, 320
- TWO\_N9
  - core\_system\_parameters, 320
- TWO\_P11
  - core\_system\_parameters, 320
- TWO\_P12
  - core\_system\_parameters, 320
- TWO\_P14
  - core\_system\_parameters, 320
- TWO\_P16
  - core\_system\_parameters, 320
- TWO\_P19
  - core\_system\_parameters, 320
- TWO\_P3
  - core\_system\_parameters, 320
- TWO\_P31
  - core\_system\_parameters, 320
- TWO\_P32
  - core\_system\_parameters, 321
- TWO\_P4
  - core\_system\_parameters, 321
- TWO\_P56
  - core\_system\_parameters, 321
- TWO\_P57
  - core\_system\_parameters, 321
- TWO\_P8
  - core\_system\_parameters, 321
- TWO\_PI
  - core\_system\_parameters, 321
- TwoBitCpxFileSignalSource, 1502
  - create\_file\_source\_hook, 1505
  - itemTypeToSize, 1505
  - packetsPerSample, 1505
  - pre\_connect\_hook, 1505
  - pre\_disconnect\_hook, 1505
  - source, 1506
- TwoBitPackedFileSignalSource, 1506
  - create\_file\_source\_hook, 1510
  - itemTypeToSize, 1510
  - packetsPerSample, 1510
  - pre\_connect\_hook, 1510
  - pre\_disconnect\_hook, 1510
  - source, 1511
- type
  - lexmsg\_t, 1152
  - ntrip\_t, 1213
  - pcv\_t, 1285
  - sol\_t, 1439
  - stream\_t, 1470
  - url\_t, 1527
- type\_of\_receiver
  - Pvt\_Conf, 1313
- udi
  - ssr\_t, 1463
- udp\_addresses
  - Pvt\_Conf, 1313
- udp\_eph\_addresses
  - Pvt\_Conf, 1314
- udp\_eph\_port
  - Pvt\_Conf, 1314
- udp\_ports
  - Pvt\_Conf, 1314
- udre
  - dgps\_t, 483
  - sbsfcorr\_t, 1397
- uhd\_signal\_source.h, 2037, 2038
- UhdSignalSource, 1511
  - connect, 1514
  - disconnect, 1514
  - get\_left\_block, 1514
  - get\_right\_block, 1514
  - item\_size, 1514
- uio\_fpga.h, 2370, 2371
- uncertainty
  - Agnss\_Ref\_Location, 350
- unlock\_channel
  - Fpga\_Multicorrelator\_8sc, 587
- unpack\_2bit\_samples, 1515
- unpack\_2bit\_samples.h, 2055, 2057
- unpack\_2bit\_samples\_sptr
  - signal\_source\_gr\_blocks, 182
- unpack\_byte\_2bit\_cpx\_samples, 1516
- unpack\_byte\_2bit\_cpx\_samples.h, 2058, 2060
- unpack\_byte\_2bit\_cpx\_samples\_sptr
  - signal\_source\_gr\_blocks, 182
- unpack\_byte\_2bit\_samples, 1518
- unpack\_byte\_2bit\_samples.h, 2060, 2061
- unpack\_byte\_2bit\_samples\_sptr
  - signal\_source\_gr\_blocks, 182
- unpack\_byte\_4bit\_samples, 1519
- unpack\_byte\_4bit\_samples.h, 2062, 2063
- unpack\_byte\_4bit\_samples\_sptr
  - signal\_source\_gr\_blocks, 182
- unpack\_intspir\_1bit\_samples, 1521
- unpack\_intspir\_1bit\_samples.h, 2064, 2065
- unpack\_intspir\_1bit\_samples\_sptr
  - signal\_source\_gr\_blocks, 182
- unpack\_short\_byte\_samples, 1522
- unpack\_short\_byte\_samples.h, 2066
- unpack\_short\_byte\_samples\_sptr
  - signal\_source\_gr\_blocks, 182
- unpack\_spir\_gss6450\_samples, 1524
- unpack\_spir\_gss6450\_samples.h, 2067, 2068
- unpack\_spir\_gss6450\_samples\_sptr
  - signal\_source\_gr\_blocks, 182
- UnscentedFilter, 1526
- update
  - ssr\_t, 1463
- update\_CRC\_stats

- Tlm\_CRC\_Stats, [1492](#)
- update\_local\_code
  - Fpga\_Multicorrelator\_8sc, [587](#)
- update\_PRN
  - Gnss\_Satellite, [881](#)
- update\_prn\_code\_length
  - Fpga\_Multicorrelator\_8sc, [587](#)
- URA
  - Gps\_CNAV\_Ephemeris, [929](#)
- ura
  - lexeph\_t, [1149](#)
  - ssr\_t, [1463](#)
- URA0
  - Gps\_CNAV\_Ephemeris, [929](#)
- URA1
  - Gps\_CNAV\_Ephemeris, [930](#)
- URA2
  - Gps\_CNAV\_Ephemeris, [930](#)
- url
  - ntrip\_t, [1213](#)
- url\_t, [1527](#)
  - dir, [1527](#)
  - path, [1527](#)
  - tint, [1527](#)
  - type, [1527](#)
- use\_automatic\_resampler
  - Acq\_Conf, [328](#)
- use\_ced
  - Tlm\_Conf, [1491](#)
- use\_CFAR\_algorithm\_flag
  - Acq\_Conf, [328](#)
- use\_e6\_for\_pvt
  - Pvt\_Conf, [1314](#)
- use\_has\_corrections
  - Pvt\_Conf, [1314](#)
- use\_unhealthy\_sats
  - Pvt\_Conf, [1314](#)
- user
  - ftp\_t, [595](#)
  - ntrip\_t, [1213](#)
- user\_clk\_drift\_ppm
  - Monitor\_Pvt, [1174](#)
- user\_clk\_offset
  - Monitor\_Pvt, [1174](#)
- ut1\_utc
  - erpd\_t, [525](#)
- utc\_cmp
  - nav\_t, [1195](#)
- utc\_gal
  - nav\_t, [1195](#)
- utc\_glo
  - nav\_t, [1195](#)
- utc\_gps
  - nav\_t, [1195](#)
- utc\_irn
  - nav\_t, [1195](#)
- utc\_qzs
  - nav\_t, [1196](#)
- utc\_sbs
  - nav\_t, [1196](#)
- utc\_time
  - Beidou\_Dnav\_Navigation\_Message, [390](#)
  - Glonass\_Gnav\_Utc\_Model, [810](#)
  - Gps\_Navigation\_Message, [960](#)
  - Monitor\_Pvt, [1174](#)
- v27\_decision\_t, [1528](#)
  - w, [1528](#)
- v27\_poly\_t, [1528](#)
  - c0, [1529](#)
  - c1, [1529](#)
- v27\_t, [1529](#)
  - decisions, [1530](#)
  - decisions\_count, [1530](#)
  - decisions\_index, [1530](#)
  - metrics1, [1530](#)
  - metrics2, [1530](#)
  - new\_metrics, [1530](#)
  - old\_metrics, [1530](#)
  - poly, [1530](#)
- V27POLYA
  - fec.h, [2168](#)
- V27POLYB
  - fec.h, [2168](#)
- valid
  - Agnss\_Ref\_Location, [350](#)
  - Agnss\_Ref\_Time, [351](#)
  - Beidou\_Dnav\_Utc\_Model, [393](#)
  - Glonass\_Gnav\_Utc\_Model, [811](#)
  - Gps\_Iono, [947](#)
  - Gps\_Utc\_Model, [964](#)
  - half\_cyc\_tag, [1088](#)
- valid\_sats
  - Monitor\_Pvt, [1175](#)
- valid\_until
  - HAS\_clock\_corrections, [1089](#)
  - HAS\_orbit\_corrections, [1090](#)
- validity\_interval\_index\_clock\_fullset\_corrections
  - Galileo\_HAS\_data, [635](#)
- validity\_interval\_index\_clock\_subset\_corrections
  - Galileo\_HAS\_data, [635](#)
- validity\_interval\_index\_code\_bias\_corrections
  - Galileo\_HAS\_data, [635](#)
- validity\_interval\_index\_orbit\_corrections
  - Galileo\_HAS\_data, [635](#)
- validity\_interval\_index\_phase\_bias\_corrections
  - Galileo\_HAS\_data, [635](#)
- var
  - opt\_t, [1222](#)
  - pcv\_t, [1285](#)
- VAR\_ACC
  - algorithms\_libs\_rtklib, [168](#)
- VAR\_BIAS
  - rtklib\_ppp.h, [1824](#)
- VAR\_CLK
  - rtklib\_ppp.h, [1825](#)
- VAR\_GRA

- algorithms\_libs\_rtklib, 168
- VAR\_GRA\_PPP
  - rtklib\_ppp.h, 1825
- VAR\_HOLDAMB
  - algorithms\_libs\_rtklib, 168
- VAR\_HWBIAS
  - algorithms\_libs\_rtklib, 168
- VAR\_IONO\_OFF
  - rtklib\_ppp.h, 1825
- VAR\_NOTEC
  - rtklib\_ionex.h, 1814
- VAR\_POS
  - algorithms\_libs\_rtklib, 169
- VAR\_POS\_PPP
  - rtklib\_ppp.h, 1825
- VAR\_VEL
  - algorithms\_libs\_rtklib, 169
- VAR\_ZTD
  - rtklib\_ppp.h, 1825
- vco
  - peph\_t, 1288
- vdop
  - Monitor\_Pvt, 1175
- vector\_length
  - Dll\_Pll\_Conf, 497
  - Dll\_Pll\_Conf\_Fpga, 503
  - Kf\_Conf, 1138
- vel
  - geph\_t, 790
  - lexeph\_t, 1149
  - peph\_t, 1288
  - seph\_t, 1411
- vel\_e
  - Monitor\_Pvt, 1175
- vel\_n
  - Monitor\_Pvt, 1175
- vel\_u
  - Monitor\_Pvt, 1175
- vel\_x
  - Monitor\_Pvt, 1175
- vel\_y
  - Monitor\_Pvt, 1175
- vel\_z
  - Monitor\_Pvt, 1175
- velocity
  - sdr\_gnss\_packet\_t, 1409
- VER\_RTKLIB
  - algorithms\_libs\_rtklib, 138
  - rtklib\_stream.h, 1867
- verified
  - DSM\_KROOT\_message, 515
- verify\_dsm\_pkr
  - osnma\_msg\_receiver, 1233
- verify\_signature\_ecdsa\_p256
  - Gnss\_Crypto, 869
- verify\_signature\_ecdsa\_p521
  - Gnss\_Crypto, 870
- very\_early\_late\_space\_chips
  - Dll\_Pll\_Conf, 497
  - Dll\_Pll\_Conf\_Fpga, 503
  - Kf\_Conf, 1138
- very\_early\_late\_space\_narrow\_chips
  - Dll\_Pll\_Conf, 497
  - Dll\_Pll\_Conf\_Fpga, 503
  - Kf\_Conf, 1139
- Viterbi\_Decoder, 1530
  - decode, 1531
  - reset, 1532
  - Viterbi\_Decoder, 1531
- viterbi\_decoder.h, 2175
- Viterbi\_Decoder\_Sbas, 1532
  - decode\_block, 1532
- viterbi\_decoder\_sbas.h, 2177, 2178
- vs
  - ssat\_t, 1460
- vsat
  - ssat\_t, 1460
- vst
  - peph\_t, 1288
- w
  - v27\_decision\_t, 1528
- wait
  - GNSSFlowgraph, 914
- wait\_and\_pop
  - Concurrent\_Queue< Data >, 459
- week
  - Agnss\_Ref\_Time, 351
  - alm\_t, 355
  - eph\_t, 523
  - GnssTime, 915
  - Monitor\_Pvt, 1175
  - raw\_t, 1327
  - sbsmsg\_t, 1403
- WEEKOFFSET
  - rtklib\_sbas.h, 1857
- what\_block
  - Gnss\_Satellite, 882
- with\_word\_type
  - ion\_gsms\_chunk\_data.h, 2094
- wlbias
  - nav\_t, 1196
- WN
  - Galileo\_Iono, 644
  - Gnss\_Ephemeris, 878
  - MACK\_message, 1160
  - Tag, 1474
- WN\_0G
  - Galileo\_Utc\_Model, 659
- WN\_a\_7
  - Galileo\_Almanac\_Helper, 605
- WN\_a\_9
  - Galileo\_Almanac\_Helper, 605
- wn\_k
  - DSM\_KROOT\_message, 515
- WN\_LSF
  - Beidou\_Dnav\_Utc\_Model, 393

- Galileo\_Utc\_Model, [659](#)
- Gps\_Utc\_Model, [964](#)
- WN\_LSF\_LSB
  - core\_system\_parameters, [321](#)
- WN\_sf0
  - OSNMA\_msg, [1231](#)
- WN\_T
  - Gps\_Utc\_Model, [964](#)
- WN\_T\_LSB
  - core\_system\_parameters, [321](#)
- WNa
  - Gnss\_Almanac, [864](#)
- WNot
  - Galileo\_Utc\_Model, [659](#)
- word
  - rtcm\_t, [1356](#)
- word\_bitsize\_
  - IONGSMSChunkUnpackingCtx< WT >, [1120](#)
- word\_shift\_direction\_
  - IONGSMSChunkUnpackingCtx< WT >, [1120](#)
- work
  - rtklib\_pvt\_gs, [1368](#)
- write
  - Concurrent\_Map< Data >, [457](#)
- write\_local\_code
  - Fpga\_Acquisition, [579](#)
- write\_reg32
  - Fpga\_spidev, [589](#)
- wtime
  - file\_t, [539](#)
- x
  - clFFT\_Dim3, [445](#)
  - rtk\_t, [1359](#)
  - sbsigpband\_t, [1399](#)
- xa
  - rtk\_t, [1359](#)
- xml\_output\_enabled
  - Pvt\_Conf, [1314](#)
- xml\_output\_path
  - Pvt\_Conf, [1314](#)
- xp
  - erpd\_t, [525](#)
- xpr
  - erpd\_t, [526](#)
- y
  - clFFT\_Dim3, [445](#)
  - sbsigpband\_t, [1399](#)
- y\_intercept
  - Dll\_Pll\_Conf, [497](#)
  - Dll\_Pll\_Conf\_Fpga, [504](#)
  - Kf\_Conf, [1139](#)
- yaw\_ang
  - ssr\_t, [1463](#)
- yaw\_rate
  - ssr\_t, [1463](#)
- yp
  - erpd\_t, [526](#)
- ypr
  - erpd\_t, [526](#)
- z
  - clFFT\_Dim3, [445](#)
  - zmq\_signal\_source.h, [2039](#), [2040](#)
  - ZmqSignalSource, [1533](#)
  - connect, [1536](#)
  - disconnect, [1536](#)
  - get\_right\_block, [1536](#)
  - item\_size, [1536](#)